Orthographic Disambiguation Incorporating Transliterated Probability

Eiji ARAMAKI **Takeshi IMAI** **Kengo Miyo** **Kazuhiko Ohe**
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8655, Japan
aramaki@hcc.h.u-tokyo.ac.jp

Abstract

Orthographic variance is a fundamental problem for many natural language processing applications. The Japanese language, in particular, contains many orthographic variants for two main reasons: (1) transliterated words allow many possible spelling variations, and (2) many characters in Japanese nouns can be omitted or substituted. Previous studies have mainly focused on the former problem; in contrast, this study has addressed both problems using the same framework. First, we automatically collected both positive examples (sets of equivalent term pairs) and negative examples (sets of inequivalent term pairs). Then, by using both sets of examples, a support vector machine based classifier determined whether two terms (t_1 and t_2) were equivalent. To boost accuracy, we added a transliterated probability $P(t_1|s)P(t_2|s)$, which is the probability that both terms (t_1 and t_2) were transliterated from the same source term (s), to the machine learning features. Experimental results yielded high levels of accuracy, demonstrating the feasibility of the proposed approach.

1 Introduction

Spelling variations, such as "*center*" and "*centre*", which have different spellings but identical meanings, are problematic for many NLP applications including information extraction (IE), question answering (QA), and machine transliteration (MT). In this paper, these variations can be termed **orthographic variants**.

Table 1: Examples of Orthographic Variants.

spaghetti	Thompson operation
スパゲッティ ⟨ supagettji ⟩	トンプソンの手術法 (Thompson's operation method)
スパゲッティー ⟨ supagettjii ⟩	トンプソンの手術 (Thompson's operation method)
スパゲティ ⟨ supagetji ⟩	トンプソン術 (Thompson operation)
スパゲティー ⟨ supagetjii ⟩	
スパゲテイ ⟨ supagetji ⟩	

* ⟨⟩ indicates a pronunciation. () indicates a translation.

The Japanese language, in particular, contains many orthographic variants, for two main reasons:

1. It imports many words from other languages using transliteration, resulting in many possible spelling variations. For example, Masuyama et al. (2004) found at least six different spellings for " spaghetti " in newspaper articles (Table 1 Left).

2. Many characters in Japanese nouns can be omitted or substituted, leading to tons of insertion variations (Daille et al., 1996) (Table 1 Right).

To address these problems, this study developed a support vector machine (SVM) based classifier that

can determine whether two terms are equivalent. Because a SVM-based approach requires positive and negative examples, we also developed a method to automatically generate both examples.

Our proposed method differs from previously developed methods in two ways.

1. Previous studies have focused solely on the former problem (transliteration); our target scope is wider. We addressed both transliteration and character omissions/substitutions using the same framework.

2. Most previous studies have focused on back-transliteration (Knight and Graehl, 1998; Goto et al., 2004), which has the goal of generating a source word (s) for a Japanese term (t). In contrast, we employed a discriminative approach, which has the goal of determining whether two terms (t_1 and t_2) are equivalent. These two goals are related. For example, if two terms (t_1 and t_2) were transliterated from the same word (s), they should be orthographic variants. To incorporate this information, we incorporated a transliterated-probability ($P(s|t_1) \times P(s|t_2)$) into the SVM features.

Although we investigated performance using medical terms, our proposed method does not depend on a target domain[1].

2 Orthographic Variance in Dictionary Entries

Before developing our methodology, we examined problems related to orthographic variance.

First, we investigated the amount of orthographic variance between two dictionaries' entries (DIC1 (Ito et al., 2003), totaling 69,604 entries, and DIC2 (Nanzando, 2001), totaling 27,971 entries).

Exact matches between entries only occurred for 10,577 terms (15.1% of DIC1, and 37.8% of DIC2). From other entries, we extracted orthographic variance as follows.

STEP 1: Extracting Term Pairs with Similar Spelling

We extracted term pairs with similar spelling (t_1 and t_2) using edit distance-based similarity (defined by Table 2). We extracted term pairs with $SIM_{ed} > 0.8$, and found 5,064 term pairs with similar spelling.

STEP 2: Judging Orthographic Variance

We then manually judged whether each term pair was composed of orthographic variants (whether or not they had the same meaning).

Our results indicated that 1,889 (37.3%) of the terms were orthographic variants.

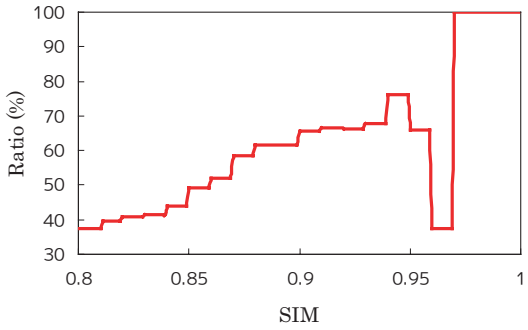

Figure 1: Similarity Threshold and Orthographic Variants Ratio.

Figure 1 presents the relation between the orthographic variation ratio and similarity threshold (0.8-1.0). As shown in the figure, a higher similarity threshold (SIM=0.96-97) does not always indicate that terms are orthographic variants.

The following term pair is a typical example:

1. 変異 **B** 型肝炎ウィルス
   (*mutated hepatitis type* **B** *virus*),

2. 変異 **C** 型肝炎ウィルス
   (*mutated hepatitis type* **C** *virus*).

They have only one character difference ("**B**" and "**C**"), resulting in high levels of spelling similarity, but the meanings are not equivalent. This type of limitation, intrinsic to measurements of spelling similarity, motivated us to develop an SVM-based classifier.

3 Method

We developed an SVM-based classifier that determines whether two terms are equivalent. Section 3.1

[1]The domain could affect the performance, because most of medical terms are imported from other languages, leading to many orthographic variants.

Table 2: Edit Distance-based Similarity (SIM_{ed}).

The edit distance-based similarity (SIM_{ed}) between two terms $(t_1, t2)$ is defined as follows:

$$SIM_{ed}(t_1, t_2) = 1 - \frac{\text{EditDistance}(t_1, t_2) \times 2}{len(t_1) + len(t_2)},$$

where $len(t_1)$ is the number of characters of t_1, $len(t_2)$ is the number of characters of t_2, Edit Distance(t_1, t_2) is the minimum number of point mutations required to change t_1 into t_2, where a point mutation is one of: (1) a change in a character, (2) the insertion of a character, and (3) the deletion of a character. For details, see (Levenshtein, 1965).

will describe the method we used to build training data, and Section 3.2 will introduce the classifier.

3.1 Automatic Building of Examples

Positive Examples

Our method uses a straight forward approach to extract positive examples. The basic idea is that orthographic variants should have (1) similar spelling, and (2) the same English translation.

The method consists of the following two steps:

STEP 1: First, using two or more translation dictionaries, extract a set of Japanese terms with the same English translation.

STEP 2: Then, for each extracted set, generate two possible term pairs (t_1 and t_2) and calculate the spelling similarity between them. Spelling similarity is measured by edit distance-based similarity (see Section 2). Any term pair with more than a threshold ($SIMed(t_1, t_2) > 0.8$) similarity is considered a positive example.

Negative Examples

We based our method of extracting negative examples using the dictionary-based method. As with positive examples, we collected term pairs with similar spellings ($SIMed(t_1, t_2) > 0.8$), but differing English translations.

However, the above heuristic is not sufficient to extract negative examples; different English terms might have the same meaning, which could cause unsuitable negative examples.

For example, t_1 "胃癌 (*stomach cancer*)" and t_2 "胃がん (*stomach carcinoma*)": although these words have differing English translations, unfortunately they are not a negative example ("*cancer*" and "*carcinoma*" are synonymous).

To address this problem, we employed a corpus-based approach, hypothesizing that if two terms are orthographic variants, they should rarely both appear in the same document. Conversely, if both terms appear together in many documents, they are unlikely to be orthographic variants (negative examples).

Based on this assumption, we defined the following scoring method:

$$Score(t_1, t_2) = \frac{\log(HIT(t_1, t_2))}{max(\log(HIT(t_1)), \log(HIT(t_2)))},$$

where $HIT(t)$ is the number of Google hits for a query t. We only used negative examples with the highest K score, and discarded the others[2].

3.2 SVM-Based Classifier

The next problem was how to convert training-data into machine learning features. We used two types of features.

Character-Based Features

We expressed different characters between two terms and their context (window size ± 1) as features, shown in Table 3. Thus, to represent an omission, "ϕ (*null*)" is considered a character. Two examples are provided in Figures 2.

Note that if terms contain two or more differing parts, all the differing parts are converted into features.

Similarity-based Features

Another type of feature is the similarity between two terms (t_1 and t_2). We employed two similarities:

1. Edit distance-based similarity $SIM_{ed}(t_1, t_2)$ (see Section 2).

2. Transliterated similarity, which is the probability that two terms (t_1 and t_2) were transliterated

[2] In the experiments in Section 4, we set K is 41,120, which is equal to the number of positive examples.

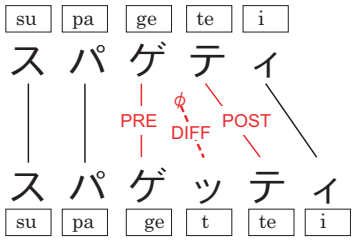

Table 3: Character-based Features.

LEX-DIFF	Differing characters between two terms, consisting of a pair of $n : m$ characters ($n > 0$ and $m > 0$). For example, we regard "ッ(t)→ ϕ" as LEX-DIFF in Figure 2 TOP.
LEX-PRE	Previous character of DIFF. We regard "ゲ (ge)" as LEX-PRE in Figure 2 TOP.
LEX-POST	Subsequent character of DIFF. We regard "テ (te)" as LEX-POST in Figure 2 TOP.
TYPE-DIFF	A script type of differing characters between two terms, classified into four categories: (1) HIRAGANA-script, (2) KATAKANA-script, (3) Chinese-character script or (4) others (symbols, numerous expressions etc.)) We regard "KATAKANA→ ϕ" as TYPE-DIFF in Figure 2 TOP.
TYPE-PRE	A type previous character of DIFF. We regard "KATAKANA" as TYPE-PRE in Figure 2 TOP.
TYPE-POST	A type subsequent character of DIFF. We regard "KATAKANA" as TYPE-POST in Figure **??**.
LEN-DIFF	A length (the number of characters) of differing parts.

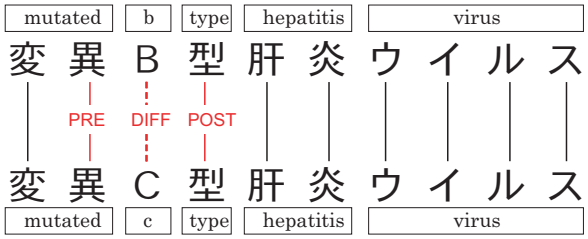

Figure 2: A Positive Example (TOP) and A Negative Example (BOTTOM).

from the same source word (t) (defined in Table 4).

Note that the latter, transliterated similarity, is applicable to a situation in which the input pair is transliterated.

4 Experiments

4.1 Test-Set

To evaluate the performance of our system, we used judged term pairs, as discussed in Section 2 (ALL-SET). We also extracted a sub-set of these pairs in order to focus on a transliteration problem (TRANS-SET).

1. **ALL-SET**: This set consisted of all examples (1,889 orthographic variants of 5,064 pairs)

2. **TRANS-SET**: This set contained only examples of transliteration (543 orthographic variants or 1,111 pairs).

4.2 Training-Set

Using the proposed method set out in Section 3, we automatically constructed a training-set from two translation dictionaries (Japan Medical Terminology English-Japanese(Nanzando, 2001) and 25-Thousand-Term Medical Dictionary(MEID, 2005)).

The resulting training-set consisted of 82,240 examples (41,120 positive examples and 41,120 negative examples).

4.3 Comparative Methods

We compared the following methods:

1. **SIM-ED**: An edit distance-based method, which regards an input with a similarity $SIM_{ed}(t_1, t_2) > TH$ as an orthographic variant.

2. **SIM-TR**: A transliterated based method, which regards an input with a spelling similarity $SIM_{tr}(t_1, t_2) > TH$ as an orthographic variant (TRANS-SET only).

3. **PROPOSED**: Our proposed method without SIM_{tr} features.

4. **PROPOSED+TR**: Our proposed method with SIM_{tr} features. (TRANS-SET only).

For SVM learning, we used TinySVM[3] with polynomial kernel (d=2).

4.4 Evaluation

We used the three following measures to evaluate our method:

$$Precision = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs found}},$$

$$Recall = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs correct}},$$

$$F_{\beta=1} = 2 \times \frac{Recall \times Precision}{Recall + Precision}.$$

4.5 Results

Table 5 presents the performance of all methods. The accuracy of similarity-based methods (SIM-ED and SIM-TR) varied depending on the threshold (TH). Figure 3 is a precision-recall graph of all methods in TRANS-SET.

In ALL-SET, PROPOSED outperformed a similarity-based method (SIM-ED) in $F_{\beta=1}$, demonstrating the feasibility of the proposed discriminative approach.

[3]http://chasen.org/ taku/software/TinySVM/

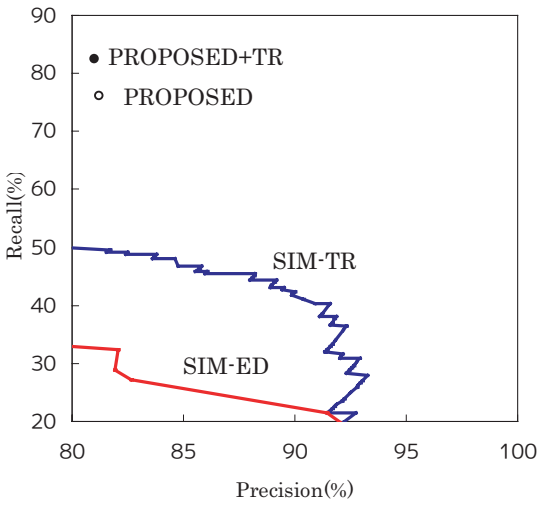

Figure 3: SIM and orthographic variants ratio.

In TRANS-SET, PROPOSED also outperformed two similarity-based methods (SIM-ED and SIM-TR). In addition, PROPOSED+TR yielded higher levels of accuracy than PROPOSED. Based on this result, we can conclude that adding transliterated-probability improved accuracy.

It was difficult to compare accuracy between the results of our study and previous studies. Previous studies used different corpora, and also focused on (back-) transliteration. However, our accuracy levels were at least as good as those in previous studies (64% by (Knight and Graehl, 1998) and 87.7% by (Goto et al., 2004)).

4.6 Error Analysis

We investigated errors from PROPOSED and PROPOSED+TR, and found two main types.

1. **Different Script Types**
   The Japanese language can be expressed using three types of script: KANJI (Chinese characters), KATAKANA, and HIRAGANA. Although each of these scripts can be converted to another, (such as "癲癇" ("*epilepsia*" in KANJI script) and "てんかん" ("*epilepsia*" in HIRAGANA script), our method cannot deal with this phenomenon. Future research will need to add steps to solve this problem.

2. **Transliteration from Non-English Lan-**

Table 5: Results

	ALL-SET			TRANS-SET		
	Precision	Recall	$F_{\beta=1}$	Precision	Recall	$F_{\beta=1}$
SIM-ED	65.2%	64.6%	0.65	91.2%	36.3%	0.51
SIM-TR	-	-	-	**92.6%**	43.9%	0.59
PROPOSED	78.2%	70.2%	0.73	81.9%	75.6%	0.78
PROPOSED+TR	-	-	-	81.7%	**82.7%**	**0.82**

* The performance in SIM-ED and SIM-TR showed the highest $F_{\beta=1}$ values.

guages

While our experimental set consisted of medical terms, including a few transliterations from Latin or German, transliteration-probability was trained using transliterations from the English language (using a general dictionary). Therefore, PROPOSED+TR results are inferior when inputs are from non-English languages. In a general domain, SIM-TR and PROPOSED+TR would probably yield higher accuracy.

5 Related Works

As noted in Section 1, transliteration is the most relevant field to our work, because it results in many orthographic variations.

Most previous transliteration studies have focused on finding the most suitable back-transliteration of a term. For example, Knight (1998) proposed a probabilistic model for transliteration. Goto et al.(2004) proposed a similar method, utilizing surrounding characters.

Their method is not only applicable to Japanese; it has already been used for Korean(Oh and Choi, 2002; Oh and Choi, 2005; Oh and Isahara, 2007), Arabic(Stalls and Knight, 1998; Sherif and Kondrak, 2007), Chinese(Li et al., 2007), and Persian(Karimi et al., 2007).

Our method uses a different kind of task-setting, compared to previous methods. It is based on determining whether two terms within the same language are equivalent. It provides high levels of accuracy, which should be practical for many applications.

Another issue is that of how to represent transliteration phenomena. Methods can be classified into three main types: grapheme-based (Li et al., 2004); phoneme-based (Knight and Graehl, 1998); and combinations of both these methods(hybrid-model(Bilac and Tanaka, 2004) and correspondence-based model(Oh and Choi, 2002; Oh and Choi, 2005)). Our proposed method employed a grapheme-based approach. We selected this kind of approach because it allows us to handle not only transliteration but also character omissions/substitutions, which we would not be able to address using a phoneme-based approach (and a combination approach).

Yoon et al. (2007) also proposed a discriminative transliteration method, but their system was based on determining whether a target term was transliterated from a source term.

Bergsma and Kondrak (2007) and Aramaki et al. (2007) proposed on a discriminative method for similar spelling terms. However, they did not deal with a transliterated probability.

Masuyama et al. (2004) collected 178,569 Japanese transliteration variants (positive examples) from a large corpus. In contrast, we collected both positive and negative examples in order to train the classifier.

6 Conclusion

We developed an SVM-based orthographic disambiguation classifier, incorporating transliteration probability. We also developed a method for collecting both positive and negative examples. Experimental results yielded high levels of accuracy, demonstrating the feasibility of the proposed approach. Our proposed classifier could become a fundamental technology for many NLP applications.

Acknowledgments

Part of this research is supported by Grant-in-Aid for Scientific Research of Japan So-

Table 4: Transliterated Similarity (SIM_{tr}).

The transliterated similarity (SIM_{tr}) between two terms $(t_1, t2)$ is defined as follows[a]:

$$SIM_{tr}(t_1, t_2) = \sum_{s \in S} P(t_1|s)P(t_2|s),$$

where S is a set of back-transliterations that are generated from both t_1 and t_2, $P(e|t)$ is a probability of Japanese term (t) comes from a source term s.

$$P(t|s) = \prod_{k=1}^{|K|} P(t_k|s_k),$$

$$P(t_k|s_k) = \frac{\text{frequency of } s_k \rightarrow t_k}{\text{frequency of } s_k},$$

where $|K|$ is the number of characters in a term t, t_k is the k-th character of a term t, s_k is the k-th character sequence of a term s, "frequency of $s_k \rightarrow t_k$" is the occurrences of the alignments, "frequency of s_k" is the occurrences of a character s_k.

To get alignment, we extracted 100,128 transliterated term pairs from a transliteration dictionary (EDP, 2005), and estimate its alignment by using GIZA++[b]. We aligned in Japanese-to-English direction, and got $1 : m$ alignments (one Japanese character : m alphabetical characters) to calculate $P(t_k|s_k)$. These formulas are equal to (Karimi et al., 2007).

[a] $SIM_{tr}(t_1, t_2)$ is a similarity (not a probability)
[b] http://www.fjoch.com/GIZA++.html

ciety for the Promotion of Science (Project Number:16200039, F.Y.2004-2007 and 18700133, F.Y.2006-2007) and the Research Collaboration Project (#047100001247) with Japan Anatomy Laboratory Co.Ltd.

References

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2007. Support vector machine based orthographic disambiguation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI2007)*, pages 21–30.

Shane Bergsma and Grzegorz Kondrak. 2007. Alignment-based discriminative string similarity. In *Proceedings of the Association for Computational Linguistics (ACL2007)*, pages 656–663.

Slaven Bilac and Hozumi Tanaka. 2004. A hybrid back-transliteration system for Japanese. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 597–603.

B. Daille, B. Habert, C. Jacquemin, and J. Royaut. 1996. Empirical observation of term variations and principles for their description. *Terminology*, 3(2):197–258.

EDP. 2005. Eijiro Japanese-English dictionary, electronic dictionary project.

Isao Goto, Naoto Kato, Terumasa Ehara, and Hideki Tanaka. 2004. Back transliteration from Japanese to English using target English context. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 827–833.

M. Ito, H. Imura, and H. Takahisa. 2003. *IGAKUSHOIN'S MEDICAL DICTIONARY*. Igakusyoin.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2007. Collapsed consonant and vowel models: New approaches for English-Persian transliteration and back-transliteration. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 648–655.

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

V. I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL2004)*, pages 159–166.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 120–127.

Takeshi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. 2004. Automatic construction of Japanese KATAKANA variant list from large corpus. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 1214–1219.

MEID. 2005. *25-Mango Medical Dictionary*. Nichigai Associates, Inc.

Nanzando. 2001. *Japan Medical Terminology English-Japanese 2nd Edition*. Committee of Medical Terminology, NANZANDO Co.,Ltd.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In *Proceedings of The 19th International Conference on Computational Linguistics (COLING2002)*, pages 758–764.

Jong-Hoon Oh and Key-Sun Choi. 2005. An ensemble of grapheme and phoneme for machine transliteration. In *Proceedings of Second International Joint Conference on Natural Language Processing (IJCNLP2005)*, pages 450–461.

Jong-Hoon Oh and Hitoshi Isahara. 2007. Machine transliteration using multiple transliteration engines and hypothesis re-ranking. In *Proceedings of MT Summit XI*, pages 353–360.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 944–951.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of The International Conference on Computational Linguistics and the 36th Annual Meeting of the Association of Computational Linguistics (COLING-ACL1998) Workshop on Computational Approaches to Semitic Languages*.

Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. 2007. Multilingual transliteration using feature based phonetic method. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 112–119.