Support Vector Machine Based Orthographic Disambiguation

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 variation can be a serious problem for many natural language-processing applications. Japanese in particular contains orthographic variation, because the large quantity of transliteration from other languages causes many possible spelling variations. To manage this problem, this paper proposes a support vector machine (SVM)-based classifier that can determine whether two terms are equivalent. We automatically collected both positive examples (sets of equivalent term pairs) and negative examples (sets of inequivalent term pairs). Experimental results yielded high levels of accuracy (87.8%), demonstrating the feasibility of the proposed approach.

1 Introduction

Orthographic variation can be a serious problem for many natural language-processing (NLP) applications, such as information extraction (IE), question answering (QA), and machine translation (MT). For example, many example-based machine translation (EBMT) (Nagao, 1984) methods, such as (Somers, 1999; Richardson et al., 2001; Sumita, 2001; Carl and Way, 2003; Aramaki and Kurohashi, 2004; Nakazawa et al., 2006), utilize a translation dictionary during bilingual text alignment. Also, several statistical machine translation (SMT)(Brown et al., 1993) methods set initial translation parameters using a translation dictionary. When consulting a dictionary, a system must disambiguate orthographic variation.

The following terms are an example of Japanese orthographic variation, corresponding to the term "*Avogadro's number*":

1. アヴォガドロ数
   (A **VO** GA DO RO SU),
2. アボガドロ数
   (A **BO** GA DO RO SU).

Although both terms are frequently used (term (1) resulted in 25,700 Google hits and Term (2) resulted in 25,000 Google hits[1]), translation dictionaries contain only one of the terms, resulting in low levels of accuracy with dictionary-based bilingual text alignment.

This paper focuses on Japanese orthographic disambiguation. Japanese orthographic variance is closely related to transliteration, because transliteration relies on pronunciation, the great differences between the sounds made in Japanese and in Western languages (mainly English) results in a variety of possible spellings.

Researchers have already proposed methods to solve this problem. For example, Knight(1998) developed a back-transliteration method using a probabilistic

[1]We got the results on May 14, 2007.

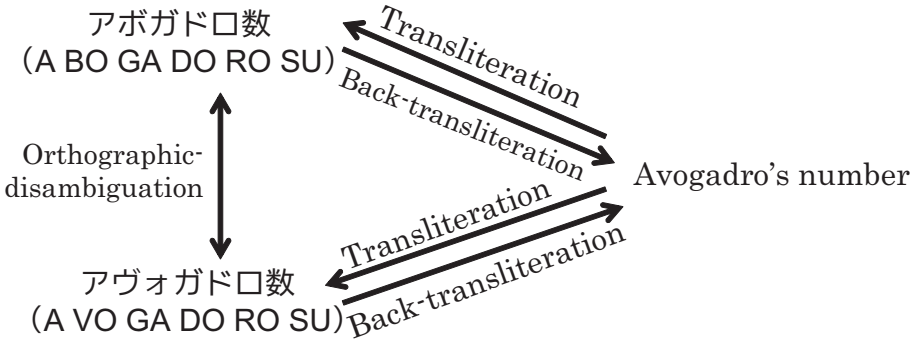

Figure 1: Transliteration and Orthographic Variation.

model. Goto et al.(2004) also developed a probabilistic model, which takes into account surrounding context. Lin and Chen(2002) developed a perceptron learning algorithm for back-transliteration. While these methods differ, they all share the same goal: being able to back-transliterate a given term into another language.

By contrast, this paper proposes a new task schema: given two Japanese terms, the system determines whether they are equivalent. Figure 1 illustrates our task schema; a foreign term can be transliterated into Japanese in several ways. While previous methods can yield suitable back-transliteration for a term, our system determines whether a pair of Japanese terms originates from the same foreign word. We expect our task-setting is more direct and practical for many applications, such as dictionary consulting in MT, IE, and so on.

For this process, our proposed method uses a machine learning technique (support vector machine, hereafter SVM (Vapnik, 1999)), which requires the two following types of data:

1. Positive examples: a term pair, which are spelled differently, but have the same meaning; and,

2. Negative examples: a term pair, which are spelled differently and have differing meanings.

While previous methods have utilized only positive examples, our proposed method also

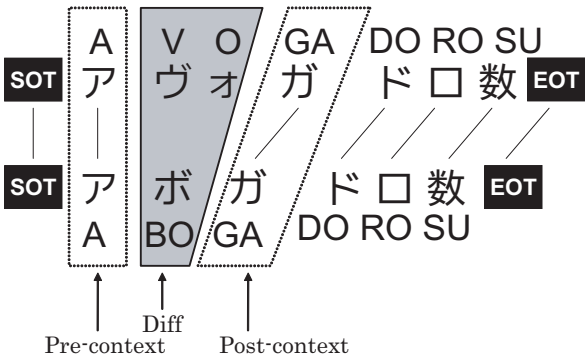

Figure 2: An Example of DIFF, PRE-CONTEXT and POST-CONTEXT.

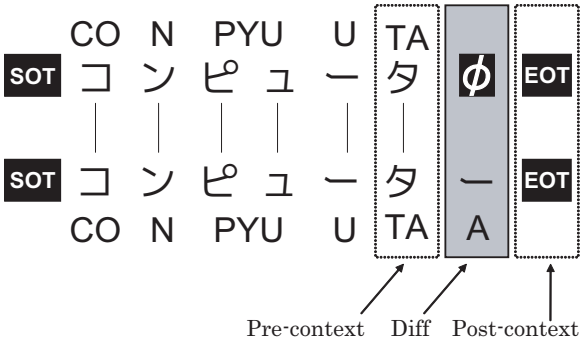

Figure 3: Another Example of DIFF, PRE-CONTEXT and POST-CONTEXT.

incorporates negative examples. Both examples can be generated automatically from translation dictionaries using spelling similarity and heuristic rules.

Experimental results yielded high accuracy (87.8%), demonstrating the feasibility of the proposed approach.

Although we investigated the performance in the medical terms, the proposed method does not depend on the target domain.

Section 2 of this paper describes how training data are built. Section 3 describes the learning method, and Section 4 presents the experimental results. Section 5 discusses related work, and Section 6 presents our conclusions.

2 Automatic Example Building

This section describes how training data are built; Section 2.1 discusses positive examples, and Section 2.2 discusses negative examples. Note that the latter is a novel task.

2.1 Positive Examples

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

The method consists of the following two steps:

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

STEP 2: Then, for each extracted set, we generate possible two term pairs ($term_1$ and $term_2$), and calculate the spelling similarity between them. Spelling similarity is measured using the following edit-distance based similarity $SIM(term_1, term_2)$:

$$SIM(term_1, term_2) = 1 - \frac{\text{EditDistance}(term_1, term_2) \times 2}{len(term_1) + len(term_2)},$$

where $len(term_1)$ is the length (the number of characters) of $term_1$, $len(term_2)$ is the length (the number of characters) of $term_2$, $EditDistance(term_1, term_2)$ is the minimum number of point mutations required to change $term_1$ into $term_2$, 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).

Any term pair with more than a threshold (TH) similarity is considered a positive example [2].

2.2 Negative Examples

As mentioned in Section 1, generating negative examples is a novel process in this field.

One simple way is to select two words from a dictionary randomly. However, such a simple method would generate a huge quantity of meaningless examples. Therefore, as in our collection of positive examples, we collected only term pairs with similar spellings.

Another problem is a balance of the example quantity. In the preliminary experiments, the number of negative examples was about three times as the number positive examples, leading to a negative bias.

Therefore, we investigated the Google hits of each term pair by using a query, such as "アヴォガドロ数 アボガドロ数".

Then, we utilize only negative examples with many Google hits, and reject low-hits examples, because of the following two reasons:

1. **Popularity**: We expect that a more popular term pair is more informative.
2. **Reliability**: We hypothesize that an orthographic pair rarely appears in one document, because one document usually has an orthographic consistency. Therefore, we can expect that if two terms cooccur in one document, they are not orthographic variants, ensuring reliability for negative examples.

The detailed steps are as follows:

[2] We set $TH = 0.8$.

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

STEP 2: Then, for each extracted set, we generate possible pairs, and calculate the spelling similarity between them. Any term pair exceeding a threshold (TH) similarity is considered a negative example candidate.

STEP 3: Finally, we investigate the Google hits for each candidate. We only use the top K-hits candidates as negative examples[3].

3 Leaning Method

Application of the method described in Section 2 yields training data, consisting of triple expressions $< term_1, term_2, +1/-1 >$, in which "+1" indicates a positive example (orthographic variants), and "-1" indicates a negative example (different terms). Table 1 provides some examples.

The next problem is how to convert training data into machine learning features. We regard the different parts and context (window size ± 1) as features:

1. DIFF: differing characters between two translations;
2. PRE-CONTEXT: previous character of DIFF; and
3. POST-CONTEXT: subsequent character of DIFF.

Figure 2 provides examples of these features. Since the different part is a gray area ("*VO(ヴォ)*" and "BO(ボ)"), we consider DIFF to be "VO:BO (ヴォ:ボ)" itself, PRE-CONTEXT to be "A (ア)" in a dotted box, and POST-CONTEXT to be "GA (ガ)" also in a dotted box.

Figure 3 provides another example; the insertion/deletion of a character can be considered the Diff using ϕ, such as "$\boxed{\phi}$:A ($\boxed{\phi}$:ー)". In addition, the start ($\boxed{SOT}$) or end ($\boxed{EOT}$) of a term can be considered a character.

Note that both PRE-CONTEXT and POST-CONTEXT consist of one character pair, while the DIFF can be a pair of $n : m$ characters ($n \geq 0$, $m \geq 0$).

In learning, we can use a back-off technique to prevent problems related to data sparseness. As a result, each different point utilizes the following four features:

- Diff + Pre-context + Post-context
- (1-back-off-a) Diff + Pre-context
- (1-back-off-b) Diff + Post-context
- (2-back-off) Diff

Figure 4 presents some examples.

4 Experiments

4.1 Test-set

To evaluate the performance of our system, we manually built a test-set as follows:

First, we extracted 5,013 similar spelling term pairs, that have more than ($SIM > 0.8$), from two dictionaries (Nanzando, 2001b),(Ito et al., 2003).

Then, for each pair, we annotated whether it is an equivalent pair (orthographic variants) or not (different terms).

Finally, we randomly extracted 883 pairs form it. We regard it as a test-set. The test-set consists of 312 positive examples and 571 negative examples. The others (4,130 examples) are used for training in comparative methods (BYHAND and COMBINATION mentioned in Section 4.3).

4.2 Training-set

By using the proposed method (in Section 2), we automatically built a training-set from two translation dictionaries (Japan Medical Terminology English-Japanese(Nanzando, 2001a) and 25-thousand-terms Medical Dictionary(MEID, 2005)). As a result, we got a training-set, consisting of 68,608 examples (21,380 positive examples and 47,228 negative examples).

[3] In the experiments in Section 4, we set $K = 21,380$, which is equal to the number of positive examples.

P/N*	$Term_1$	$Term_2$
+1	ヨードピラセット (YO O DO PI RA SE TTO; iodopyracet)	ヨードピラセト (YO O DO PI RA SE TO; iodopyracet)
+1	マイクロメーター (MA I KU RO ME E TA A; micrometer)	マイクロメータ (MA I KU RO ME E TA; micrometer)
+1	アンプリファイア (A N PU RI FA I A; amplifier)	アンプリファイヤー (A N PU RI FA I YA A; amplifier)
+1	オシロスコープ (O SI RO SU KO O PU; oscilloscope)	オッシロスコープ (O SSI RO SU KO O PU; oscilloscope)
+1	動コンプライアンス (DO U KO N PU RA I A N SU; dynamic compliance)	動的コンプライアンス (DO U TE KI KO N PU RA I A N SU; dynamic compliance)
+1	浸透圧性ショック (SI N TO O A TU SE I SYO KKU; osmotic shock)	浸透圧ショック (SI N TO O A TU SYO KKU; osmotic shock)
+1	マールブルグウイルス (MA A RU BU RU GU U I RU SU; Marburg virus)	マルブルグウイルス (MA RU BU RU GU U I RU SU; Marburg virus)
+1	ドールトンの法則 (DO O RU TO N NO HO O SO KU; Dalton law)	ドルトンの法則 (DO RU TO N NO HO O SO KU; Dalton law)
-1	B 型肝炎 (BI I GA TA KA N E N; hepatitis B)	C 型肝炎 (SI I GA TA KA N E N; hepatitis C)
-1	トランス (TO RA N SU; trance)	トランジスタ (TO RA N JI SU TA; transistor)
-1	ビタミン P (BI TA MI N PI I; vitamin P)	ビタミン C (BI TA MI N SI I; vitamin C)
-1	カドミウム (KA DO MI U MU; cadmium)	カルシウム (KA RU SI U MU; calcium)
-1	アルコール (A RU KO O RU; alcohol)	グルコース (GU RU KO O SU; glucose)
-1	メラトニン (ME RA TO NI N; melatonin)	セロトニン (SE RA TO NI N; serotonine)
-1	クローン (KU RO O N; clone)	クラーレ (KU RA A RE; curare)
-1	ケトン生成 (KE TO N SE I SE I; ketogenesis)	メタン生成 (ME TA N SE I SE I; methanation)
-1	リード指数 (RI I DO SI SU U; Reid index)	リビー指数 (RI BI I SI SU U; Livi index)
-1	トマチン (TO MA CHI N; tomatine)	ヘマチン (HE MA CHI N; haematin)
-1	バルーン法 (BA RU U N HO; balloon method)	ラグーン法 (RA GU U N HO; lagoon method)

Table 1: Some Examples of Training-set.

* "+1" indicates positive examples, and "-1" indicates negative examples.

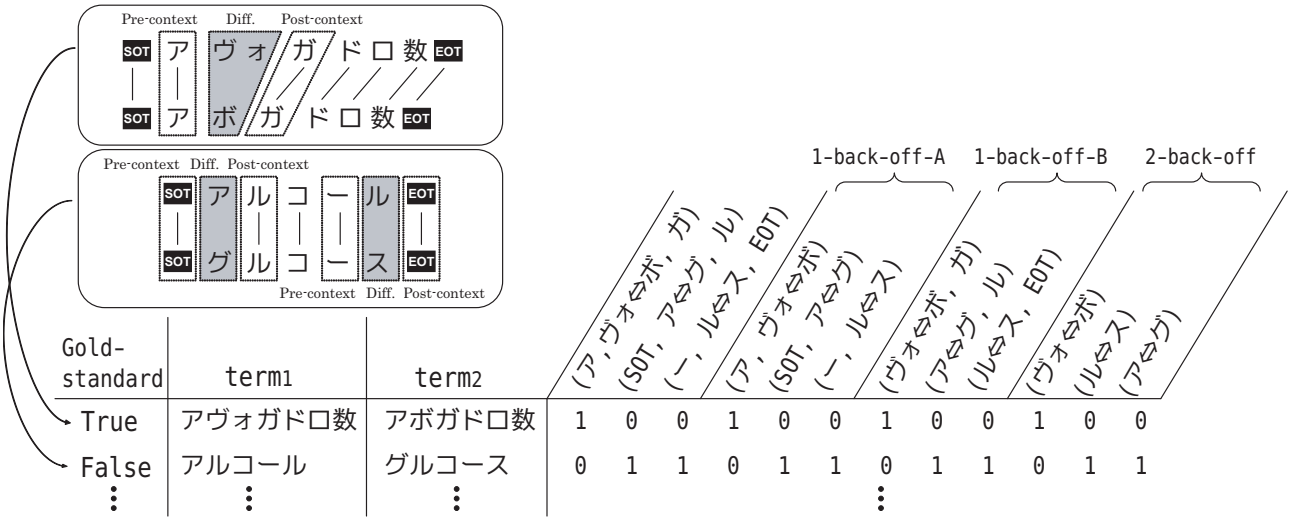

Figure 4: An Example of Features.

4.3 Comparative Methods

we compared the following methods:

1. **EDITDISTANCE(TH)**: an edit-distance-based method, which regards an example with a spelling similarity $SIM(term_1, term_2) > TH$ as an orthographic variants. The performance of this method changes, depending on TH.

2. **BYHAND**: a SVM-based method, trained by manually annotated corpus, consists of 4,130 examples.

3. **AUTOMATIC**: a SVM-based method, trained by an automatically build training-set.

4. **COMBINATION**: a SVM-based method, trained by both BYHAND corpus and AUTOMATIC corpus.

For SVM learning, we used TinySVM[4] with a linear kernel[5].

4.4 Evaluation

To evaluate our method, we used three measures, precision, recall and accuracy, defined as follows:

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

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

$$Accuracy = \frac{\text{\# of pairs correct}}{\text{total \# of pairs in test-set}}.$$

4.5 Results

First, we checked the performance of **EDITDISTANCE(TH)** in various TH values. Figure 5 presents the results. While the precision is basically proportional to the spelling similarity (TH), it drops down in the high TH ($TH \div 0.96$), indicating a highly similar spelling term pair not always have to be the orthographic variants.

Table 2 presents the performance of all methods. AUTOMATIC did not obtain a higher accuracy than BYHAND, the combination of them is the highest accuracy, demonstrating the basic feasibility of our approach. The precision-recall graph (Figure 6) also shows the advantage of COMBINATION .

4.6 Error Analysis

We investigated the errors from COMBINATION, and found that many errors came from

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

[5]Although we tried a polynomial kernel and an RBF kernel, their performance are almost equal to a linear kernel.

a verbal omission, which is different phenomenon from transliteration.

For example, a test-set has the following positive example:

1. カルシウム・チャネル
   (calcium channel; KA RU SI U MU CHA NE RU),
2. カルシウム**イオン**・チャネル
   (calcium **ion** channel; KA RU SI U MU **I O N** CHA NE RU).

Because a term "*ion*" is without saying inferable in this case, it can be omitted. Capturing such an operation requires a very high level of understanding of the meaning of the terms.

To focus on a transliteration problem, we manually removed such examples from our test-set, and built a sub-set of it, consisting of only transliterations. The result is shown in Table 3. The accuracy of COMBINATION is higher than 90%.

It is difficult to compare this accuracy to that of the previous studies because (1) their corpus were different from ours and (2) previous studies focused on back-transliteration. However, we can say that the present accuracy is, at least, not behind from the previous researchers (64% by (Knight and Graehl, 1998) and 87.7% by (Goto et al., 2004)). We expect that the present accuracy is practical in many applications.

Finally, we investigate the differences between AUTOMATIC and BYHAND results (the AUTOMATIC accuracy is much lower than the BYHAND by 8.5 points in Table 2). One of the reasons is dictionary specific styles, such as numerous expression variants ("8,8, ⑧,Ⅷ,viii,ⅷ,viii,八 (Japanese number expression)"), hyphenation variants ("‐,—,=,=,･") and so on. Because the BYHAND training-set and the test-set came from the same dictionaries, BYHAND already knows such variants are meaningless differences. However, AUTOMATIC, using different dictionaries, sometimes suffered from unseen number expression/hyphenation variants.

Note that in transliteration accuracy (in Table 3), their accuracies (BYHAND and AUTOMATIC) are not so different.

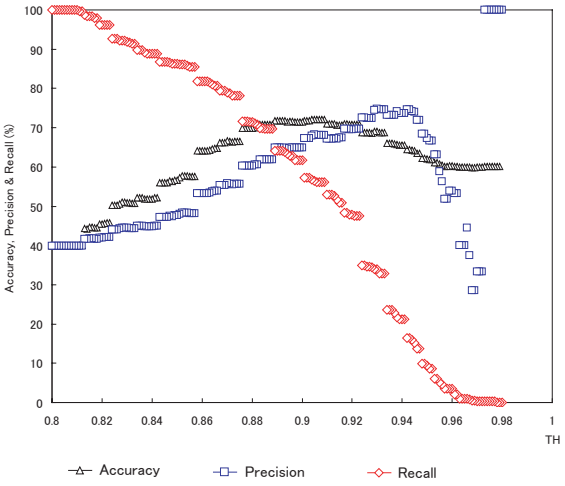

Figure 5: *TH* and EDITDISTANCE Performance.

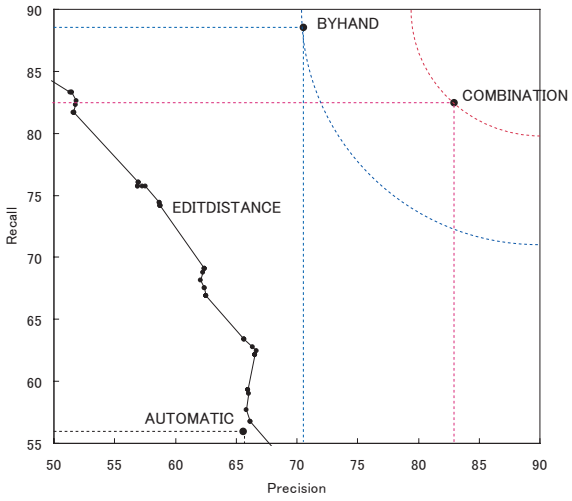

Figure 6: Precision and Recall.

Table 2: Results

methods	Precision	Recall	Accuracy
EDIT-DISTANCE(0.91)	67.2%(164/244)	52.6% (164/312)	70.9% (626/883)
BYHAND	70.4%(276/392)	**88.4% (276/312)**	82.7% (731/883)
AUTOMATIC	65.7%(177/269)	56.7% (177/312)	74.2% (656/883)
COMBINATION	**82.9%(258/311)**	82.6% (258/312)	**87.8% (776/883)**

* The performance in EDIT-DISTANCE(0.91) showed the highest accuracy in various TH values.

Table 3: Results of a sub-set (Transliteration Only)

methods	Precision	Recall	Accuracy
BYHAND	67.7%(122/180)	**91.0%(122/134)**	80.3% (286/356)
AUTOMATIC	77.3%(109/141)	81.3% (109/134)	83.9% (299/356)
COMBINATION	**90.6%(117/129)**	90.7% (117/134)	**91.9% (327/356)**

5 Related Works

As noted in Section 1, transliteration is the field most relevant to our work, because many orthographic variations come from borrowed words. Our proposed method differs from previous studies in the following three ways: (1) task setting, (2) negative examples, and (3) target scope.

5.1 Task Setting

Most previous studies have involved finding the most suitable back-transliteration of a term.

For example, given an observed Japanese string o by optical character recognition (OCR) software, Knight and Graehl (1998) finds a suitable English word w. For this process, they developed a probabilistic model that decomposed a transliteration into sub-operations as follows:

$$P(w)P(e|w)P(j|e)P(k|j)P(o|k),$$

where $P(w)$ generates written English word sequences, $P(e|w)$ pronounces English word sequences, $P(j|e)$ converts English sounds into Japanese sounds, $P(k|j)$ converts Japanese sounds to KATAKANA writing, and $P(o|k)$ introduces misspellings caused by OCR.

While this method is phoneme-based, Bilac and Tanaka(2004) combined phoneme-based and graphme-based transliteration. Goto et al.(2004) proposed a similar method, utilizing the surrounding context.

Such methods are not only applicable to Japanese; it can also be used for Arabic(Stalls and Knight, 1998; Sherif and Kondrak, 2007), Chinese(Li et al., 2007), Persian(Karimi et al., 2007).

The task-setting involved in our method differs from previous methods. Our methodology involves determining whether two terms in the same language are equivalent, making our task-setting more direct and suitable than previous methods for many applications, such as dictionary consulting in MT and information retrieval.

Note that Yoon et al.(2007) also proposed a discriminative transliteration method, but their system determines whether a target term is transliterated from a source term or not.

5.2 Negative Examples

Our task setting requires negative examples, consisting of term pairs with similar spellings, but different meanings.

By contrast, previous research involved only positive examples. For example, Masuyama et al.(2004) collected 178,569 Japanese transliteration variants (positive examples) from large corpora. However, they paid little attention to negative examples.

5.3 Target Scope

As mentioned above, orthographic variation in Japanese results mainly from transliteration. However, our target includes several different phenomena, such as verbal omissions mentioned in Section 4.6. Although the accuracy for omissions is not enough, our method addresses it easily, while previous methods are unable to handle this kind of phenomenon.

6 Conclusion

In this paper, we proposed a SVM-based orthographic disambiguation method. We also proposed a method for collecting both positive and negative examples. Experimental results yielded high levels of accuracy (87.8%), demonstrating the feasibility of the proposed approach.

Acknowledgments

Part of this research is supported by Grant-in-Aid for Scientific Research of Japan Society 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 and Sadao Kurohashi. 2004. Example-based machine translation using structual translation examples. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT2004)*, pages 91–94.

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.

Peter F. Brown, Stephen A. Della Pietra, Vi cent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Michael Carl and Andy Way. 2003. *Recent Advances in Example-based Machine Translation*. Kluwer Academic Publishers.

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. *IGAKU-SHOIN'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, 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.

Wei-Hao Lin and Hsin-Hsi Chen. 2002. Backward machine transliteration by learning phonetic similarity. In *Proceeding of the 6th conference on Natural language learning*, pages 1–7.

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.

Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *In Artificial and Human Intelligence*, pages 173–180.

Toshiaki Nakazawa, Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2006. Example-based machine translation based on deeper NLP. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT2006)*, pages 64–70.

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

Nanzando. 2001b. *Promedica ver.3*. NANZANDO Co.,Ltd.

Stephen D. Richardson, William B. Dolan, Arul Menezes, and Monica Corston-Oliver. 2001. Overcoming the customization bottleneck using example-based MT. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL2001) Workshop on Data-Driven Methods in Machine Translation*, pages 9–16.

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.

Harold Somers. 1999. Example-based machine translation. In *Machine Translation*, pages 113–157.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of The 17th International Conference on Computational Linguistics (COLING1998) Workshop on Computational Approaches to Semitic Languages*.

Eiichiro Sumita. 2001. Example-based machine translation using dp-matching between word sequences. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL2001) Workshop on Data-Driven Methods in Machine Translation*, pages 1–8.

Vladimir Vapnik. 1999. *The Nature of Statistical Learning Theory*. Springer-Verlag.

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.