Phonetically Based Extraction of Japanese Synonyms from Rakuten Ichiba's Item Titles

Ohnmar Htun[a], Koji Murakami[b], Yu Hirate[a]
[a]Rakuten Institute of Technology Tokyo, Rakuten Inc., Tokyo, Japan
ohnmar.htun@rakuten.com
yu.hirate@rakuten.com
[b]Rakuten Institute of Technology New York, Rakuten USA, Inc., New York, USA
koji.murakami@rakuten.com

Abstract

This paper presents a method for the phonetically based extraction of Japanese synonyms from item titles of Rakuten Ichiba. In general, synonyms are words with the same or similar meaning in a semantic sense; however, we focus here on those synonyms which appear as transliterations between English and Japanese, using Katakana, Hiragana, Kanji and a mixture of these scripts. The method consists of three parts: generation of the candidate word pairs using phrase detection (collocation) at the preprocessing stage; mapping similar sounds using Soundex and a cross-language sound group; measuring the similarity based on the Levenshtein and stochastic distances; and ranking the synonym pairs using fuzzy matching in the post-processing stage. We carry out two experiments based on two different sound mapping datasets, each of which measures the similarity scores from two different algorithms. The results from the baseline and cross-language models achieve precision values of 0.9208 and 0.9983, respectively. Our method is applicable to various fields of linguistic research, for example building a thesaurus/new name entity lookup for a search engine, machine translation and natural language generation, and improving output of voice recognition systems.

Keywords: Japanese synonym, transliteration mining, phonetic similarity, information retrieval

1. Introduction

Due to linguistic borrowing between languages, phonetic similarities can be found within a language (i.e., transcription) or between two or more languages (i.e., transliteration). In Japanese, Katakana is used to express sound effects and transliterated foreign words using Japanese pronunciation rules and syllables. The ending of words is therefore quite different from the original pronunciation. Fashion-related words are mostly constructed using foreign language words, for examples, "Lounge Style|ラウンジスタイル [RAUNJISUTAIRU]", "Glenfield|グレンフィールド [GURENFIRUDO]", and "Insignia Dress|インシグニアドレス [INSHIGUNIADORESU]".

Typically, synonyms are words with the same or similar meaning in a semantic sense, and can be easily found in a thesaurus. However, synonyms in Japanese can be found not only as semantically relevant words, but also as words that are phonetically equivalent across languages. For example, "basket" in English can be translated into Japanese as 籠 [KAGO], 篭 [KAGO], or transliterated as バスケット [BASUKETTO] by adopting sounds directly from the source language; this is also known as a "Loanword" or "Transliterated word". Newly created consumer products and services are being introduced to offline marketplaces and online digital market spaces on a daily basis, and many loanwords have been created as synonyms for consumer products in Japanese. In fact, query expansions in E-commerce search engines require the construction of sets of these synonymous names for concepts. The motivation for this work is to extract new synonym pairs from item-title phrases in the ladies' fashion database of Rakuten Ichiba (楽天市場) [1] to enhance the vocabularies of synonym dictionary in the search platform development.

In this work, we focus on extracting synonyms appearing as transliterations between English and Japanese, using Katakana, Hiragana and Kanji or a mixture of these scripts. The method presented here is an extension of prior research (Htun et al., 2011; Htun et al., 2012; Finch et al., 2012). The current approach is slightly different from previous studies; rather than bilingual pairs, the format of the test datasets contains long phrases with mixed encoding such as Latin alphabets, Japanese scripts, symbols and other annotated formats (e.g., date & measurement). The Gensim phrases (collocation) detection module (Mikolov T et al., 2013) is used to generate the candidate pairs in the preprocessing stage. The process of mapping sound uses Soundex (SDX) and cross-language sound grouping (CLSG). When measuring similarity, the Levenshtein distance (LD) algorithm (Levenshtein, 1966) is used to measure the CLSG directly, and each edit operation has a weight of one. The stochastic distance (SD) model (Ristad et al., 1998; Sajjad et al., 2012; Htun et al., 2012) is used to adjust the training parameters and iterations. The addition of a post-processing step with fuzzy matching[2] helps in extracting the synonyms accurately. The experiments generated two results since we constructed two models using baseline Soundex training (SDX-SD) and cross-language phonetic training (CLSG-SD). Our testbed contains 139,493 synonym pairs in the training data and 4,178,660 candidate pairs in the testing data. The results from baseline and cross-language models achieved a precision of 0.9208 and 0.9983 respectively.

The main contribution of this paper is the demonstration of a novel practical method by applying it to a real business support system; it is also applicable to various linguistic research studies, for example building a thesaurus/new name entity lookup for a search engine, machine translation and natural language generation, or improving the output of a voice recognition system. The remainder of the paper is organized as follows: in Section 2, we review prior

[1] https://www.rakuten.co.jp/

[2] https://pypi.python.org/pypi/fuzzywuzzy

research; Section 3 presents our methodology; Section 4 describes the experiments; Section 5 provides experimental data; Section 6 presents the results; Section 7 gives a short evaluation and discussion of the results obtained in the previous section; and Section 8 concludes this work.

2. Related Work

Earlier studies of phonetically based Japanese synonym extraction are reported by Tsuji et al. (2002). These authors manually construct transliteration rules between French and Japanese, and between English-Japanese. Katakana words convert into mora units[3], then match the character level between Japanese and French, and rank the pairs based on their frequency. They apply a string matching algorithm to find the longest common subsequence and use Dice to extract the word from the French part of corpora. However, the result achieves a precision of only 80% and a recall of 20%, the amount of the test data is very small.

A technique similar to phonetic matching has been applied to Japanese search engines using the PostgreSQL open source database by Yusukawa et al. (2012). They develop a sound grouping based on the similarity of Japanese speech sounds, and matching based on morphological analysis (MeCab [4]); they then extract terms from the document using Indri[5] and apply the Fuzzy string-matching function of PostgreSQL[6]. Using this method, they extract 84 million terms from the 67 million Japanese documents in the ClueWeb09-JA [7]collection. This work integrates an internal module of jpfuzzystrmatch into PostgreSQL. However, it suffers from an excessive generation of matches (i.e., both correct and incorrect).

Another approach to generating a large list of technical transliterated terms between Japanese and English employs a function of phrase-based statistical machine translation (PBSMT) function from Moses (Koehn et al., 2007). This is used to train a bilingual dictionary (Katakana-English) and aligned bilingual pairs (Japanese-English) using Wikipedia article titles (69,000 pair in total), and is tested with a large amount of data (24 million parallel title pairs). This method generates 7 million phrase pairs (Katakana-English) with high precision and recall, they consider to generate transliteration pairs from non-parallel data.

Prior research by Htun et al. (2012) and Finch et al. (2012) has been extended by adding a new approach (word reordering) to the joint process of transliteration and translation pairs (Finch et al., 2017) for mining bilingual lexicons from pairs of parallel short word sequences. They use four methods: the GIZA++ alignment tool (Och and Nay, 2003); the joint length base measure; stochastic edit distance based Dirichlet process model; and the stochastic edit distance base Dirichlet process model with word reordering. These are tested and evaluated using bilingual Wikipedia article titles in English-Japanese (137,780) and English-Chinese (192,407). However, this new approach achieves an F-score of only 0.898 for English-Japanese and 0.82 in English-Chinese, and the computational cost is excessively high. Our model uses only SD with a noise model (Htun et al., 2012) based on a single-word, and our current approach allows model learning of one or more words.

A variety of approaches have been proposed to extract Japanese-English transliterated pairs, most of which attempt to extract pairs from the bilingual corpora using different measures or learning algorithms. In recent years, the most popular word embedding model, Word2Vec (Mikolov et al., 2013a, 2013b), has enabled researchers to estimate the representations of words, as in the famous example: “King – Man + Woman = Queen”. However, this representation cannot identify whether the words are similar to or different from each other in terms of pronunciation. Our approach uses phrase detection (collocation) to generate the candidate pairs. This approach gives a reduction in the computational cost of pairing and adds phonological knowledge support to the LD and SD model similarity scores. In post-processing, fuzzy partial matching eliminates duplicated extended pairs with the same sound. The experimental results show that our CLSG-SD model achieves a precision of more than 0.99, a significant improvement over previously proposed models (Htun et al., 2012).

3. Methodology

Our methodology consists of three steps:

- preprocessing;
- measurement of phonetic similarity; and
- post processing.

Figure 1 gives an overview of this methodology.

3.1 Preprocessing

3.1.1 Removing Abbreviations

We first clean abbreviations and formatted segments in the title strings using regular expression processing.

3.1.2 Parsing with MeCab

The cleaned strings are parsed using MeCab[8] for word segmenting, POS tagging and elimination of some unnecessary segments. (e.g., a segment “の で” in feature of “助詞,接続助詞/particle, connecting particle”).

3.1.2 Pairing Using Phrase (collocation) Detection

The Phrases module in genism (Mikolov et al., 2013a, 2013b) has two basic steps.

- Collection of the word and word bigram frequencies, using a corpus of documents. This is referred to as training the model.
- Use of the trained model to detect phrases in the corpus. The detected phrase will merge with neighboring words if it is evaluated as being part of a collocation.

Trigrams use phrases transformed into bigrams as input, and we iterate the two steps above. We generate phrases based on bigram with the minimum count (i.e., min_count)

[3] https://en.wikipedia.org/wiki/Mora_(linguistics)
[4] https://github.com/taku910/mecab
[5] https://www.lemurproject.org/lemur/indexing.php
[6] https://www.postgresql.org/
[7] http://lemurproject.org/clueweb09.php/
[8] https://github.com/neologd/mecab-ipadic-neologd/blob/master/README.ja.md

set to one and the threshold set to nine. The trigram counts use the default parameters.

3.2 Measurement of Phonetic Similarity

3.2.1 Romanization and Simplification of Sounds

Our method involves only the measurement of phonetic strings. Non-Latin language scripts are therefore first converted into Romanized versions. We utilize various Japanese Romanization converters from the Python library, such as jaconv[9], romkan 0.2.1[10], and jProcessing 0.1[11]. The next step, simplifying sounds, has two stages. The first simplification corresponds to the native phoneme of each language. For example, gya[ギャ] is simplified as 'g', tsu[つ] is simplified as 'S' in Japanese, and 'sh' is simplified as 'S' in English. In the second step, we simplify this again using SDX and CLSG (Kodama, 2010; Htun et al., 2011; Htun et al., 2012).

3.2.2 Measuring Similarity

Levenshtein Distance

The LD (Levenshtein, 1966) is a dissimilarity measure between two strings. It is the minimum number of character edits required to transform one string into the other, using the edit operations of insertion, deletion, or substitution of a single character. The editing cost for each operation set is one, and the LD is calculated as follows:

$$LD = I + D + S \qquad (1)$$

where I = the number of insertions
D = the number of deletions
S = the number of substitutions

The LD is normalized, denoted here by LDn, and defined as follows:

$$LDn = 1 - \left(\frac{LD}{(L1 + L2)}\right) \qquad (2)$$

where L1 and L2 are the lengths of converted strings from the sound simplification process. LD_n lies in the range $0 \leq LD_n \leq 1$. We refer to this score as the LD similarity result.

Stochastic Distance

The SD is an unsupervised generative model (Ristad et al., 1998; Sajjad et al., 2012) that can assign a joint probability to a pair of strings using the probabilities of edit operations. An edit cost (P_j) is calculated by applying the negative logarithm to the joint probability of an edit (e) as given below:

$$P_j = -Log(P(e)) \qquad (3)$$

Exponentially many edit sequences may be generated, and this increases the probability of the entire string pair P(X,Y). The edit distance is defined as $d_s(X,Y)$ and is calculated by summing the derivation probabilities over all paths as follows:

$$d_s(X,Y) = \sum_{s \epsilon Z} \sum_{j \epsilon s} P_j \qquad (4)$$

Z = {s_1, s_2, s_3, ..., s_i} is the set of all edit operation sequences that are generated between strings X and Y.

An edit is represented by j, and s = (j_1, j_2, j_3, ..., j_n) denotes a sequence of edits (an edit path).

The edit costs are learned using the expectation maximization (EM) algorithm, which involves a forward-backward dynamic programing technique. The SD learns using data with both transliteration and non-transliteration, and has two sub-models: transliteration (clean model), which assigns a high probability, and non-transliteration (noise model), which assigns a low probability. The full SD model is an interpolation of both models.

$$P(X,Y) = (1 - \lambda)P_t(X,Y) + \lambda P_n(X,Y) \qquad (5)$$

where λ is the prior probability of the data being noise (a non-transliteration pair), P_t is the probability of the clean model, and P_n is the probability of the noise model.

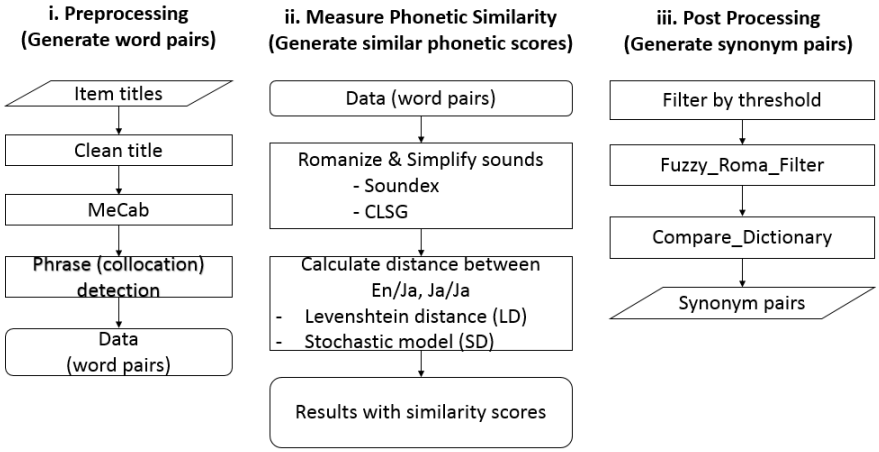

Figure 1: Overview of the methodology

[9] https://pypi.python.org/pypi/jaconv/
[10] https://pypi.python.org/pypi/romkan
[11] https://pypi.python.org/pypi/jProcessing/0.1

3.3 Post-processing

3.3.1 Filter by Thresholds

Thresholding is commonly used in information retrieval (IR) analysis. It is a procedure similar to clustering to assign a similarity score to a class indicating whether or not the score is greater than a predefined threshold. The performance of IR algorithms depends on the output quality of the thresholding process. For example, we assign a threshold value (T) to SD scores as: non-synonym > T ≥ synonym. We use joint thresholds (both LD and SD) in each experimental result.

3.3.2 Fuzzy Roma Filter

To eliminate pairs with similar sounds and meaning with one or more additional characters (known as pairing error), the fuzzy ratio function is used to rank these kinds of similar strings and to extract the top-ranked string.

3.3.3 Dictionary Comparison

The main objective of this stage is to extract new synonym pairs which are not included in existing dictionaries. This function involves only straightforward matching with synonym pairs from existing dictionaries. Finally, the new synonym pairs are extracted.

4. Experiments

The experiments were carried out to measure phonetic similarity using two methods on two different phonetic coding datasets, giving a total of four experimental conditions as shown below:

Experiment - I		**Experiment -II**	
SDX Grouping Data		CLSG Grouping Data	
Levenshtein	Stochastic	Levenshtein	Stochastic

Table 1: Set of experiments

Experiment I involves two algorithms using SDX, and the baseline measurements are compared to the results from Experiment II.

Soundex:
The Romanized candidate pairs are converted to a four-character code that is based on the six-articulation group. For example, the candidate pair "bamboo grass|バンブーグラス" is converted into SDX coding as "B512|B512".

Cross-Language Phonetic Grouping:
The CLSG approach is an extension of Soundex, and focuses on finding similar-sounding text between English and a group of Asian languages: Indonesian, Japanese, Korean, Malay, Myanmar, Thai, and Vietnamese. This experiment used CLSG version 1. For example, the candidate pair "bamboo grass|バンブーグラス" is converted into CLSG coding as "191574|191574".

Levenshtein Distance:
In (Htun et al., 2011), a variable weight in substitution operation sets 0.5 if the relation of phonetic coding characters belongs to the same place of articulation and manner; however, it sets 1 if it is not in the same place of articulation and manner. In this experiment, we apply 1 for each operation (i.e., insertion, deletion, and substitution) and measure directly to the phonetic coding converted strings.

Stochastic Distance:
The model was trained in a completely unsupervised way. The average training time was about two hours for 242,207 pairs, using 400 training iterations. Testing time was mostly less than one minute in all cases, from the minimum 55,892 training pairs to the maximum of 1,188,291. Training and testing data should use the phonetic coding; otherwise, the model cannot learn from the testing data. The SD function returns a probability score between 0 and 1.

Threshold and Filtering:
We used a joint threshold to filter out non-phonetic synonym pairs. In the baseline experiment, we allocated joint thresholds of a SDX-LD similarity score and a SDX-SD probability score of 0.875 and 0.9999 respectively. In the same way, the experiment using CLSG data applied a joint threshold of a CLSG-LD similarity score and CLSG-SD probability score of 0.85 and 0.9999 respectively.

5. Data

5.1 Training Data

The training data contained 242,207 synonym pairs of Japanese-English transliterations and Japanese-Japanese transcriptions, taken from the existing thesaurus dictionary and the Egi (RIT) transliteration dataset (2017). Training data was also required to clean unnecessary numerical characters, symbols, and so on. Some examples of source training data pairs (before cleaning and converting to phonetic transcriptions) are given in Table 2.

Synonym-1	**Synonym-2**
黒糖クルミ	黒糖くるみ
カツウラ化粧品	かつ化粧品
黒胡椒黒胡麻ペースト	黒ごまペースト
TIMETIMER	タイムタイマー
TIME VOYAGER	タイムボイジャー
ロストボール	ろすとボール
mickeycandybowl!	ミッキーキャンディーボール
ベッキー♪#	ベッキー
任天堂 wifi	ニンテンドーwi-fi

Table 2: Examples of source training data

Figure 2 shows the statistics for the types of synonym pairs. The greatest number of synonym types was English-Katakana transliteration pairs, with 171,867 in total. The lowest number of synonym types were English-Hiragana and English-Kanji with 313 and 328 receptively.

5.2 Test Data

The test dataset was extracted from titles of Rakuten Ichiba women's fashion items, and contained a total of 5,821,560 titles in 12 sub-categories. Following the process of pairing, 4,178,660 candidate pairs were generated. Table 3 presents statistics for the number of titles and generated candidate pairs in each sub-category.

	Sub-category women's Fashion	# of titles	# of candidate pairs
1	Tops	1,831,078	1,188,291
2	Dresses	172,441	162,482
3	Outerwear	531,059	446,724
4	Bottoms	989,564	687,201
5	Other Fashion	187,290	148,188
6	Others	431,931	217,840
7	Suits	54,302	57,268
8	Kimonos	609689	440,741
9	One Piece Dresses	714,313	528,371
10	Costumes	149,469	154,802
11	Swimwear	111,543	90,860
12	All-in-One	38,881	55,892

Table 3: Number of titles and candidate pairs generated in each sub-category by the CLSG test

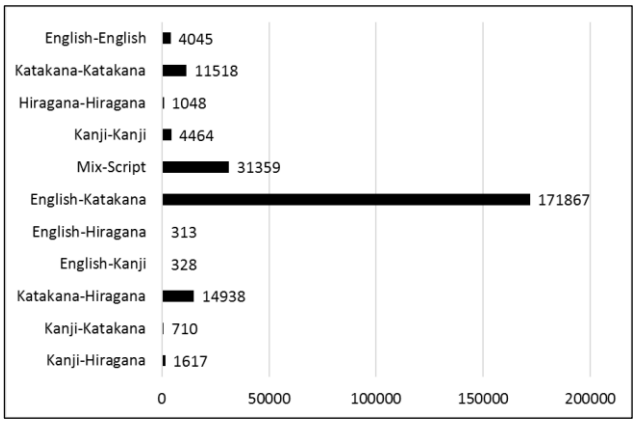

Figure 2: Type of synonyms in training data

	Experiment I		Experiment II	
Synonym pair	SDX_LD	SDX_SD	CLSG_LD	CLSG_SD
mawaru penguindrum|輪るピングドラム	1	0.999997	0.947368	1
senbonzakura|千本桜衣装	1	0.999996	0.875	0.999969
rage burst|レイジバースト	0.875	0.999905	0.909091	0.999962
parasite chest|パラサイトチェスト	1	0.999996	0.866667	0.999976
bone princess|ボーンプリンセス	1	0.999998	1	1
ensemble star fine|あんさんぶるスターズ	0.875	0.999874	0.85	0.999992
touka gettan|桃華月憚	1	0.999994	0.909091	0.999949
durarara|デュラララ	1	0.999997	1	0.999941

Table 4: Examples of phonetic similarity scores from the results of Experiments I and II

		Experiment I (SDX)			Experiment II (CLSG)		
	Subcategory	Extracted pairs	Recall	Precision	Extracted pairs	Recall	Precision
1	Tops	7,104	0.6861	0.90	5,649	0.7045	0.99
2	Dresses	466	0.7036	0.97	400	0.7000	1.00
3	Outerwear	4,027	0.6907	0.97	3,274	0.7045	0.99
4	Bottoms	4,720	0.8875	0.90	3,949	0.7000	1.00
5	Other Fashion	498	0.7017	0.88	385	0.7000	1.00
6	Others	1,230	0.7098	0.87	1,011	0.7000	1.00
7	Suits	533	0.7000	1.00	423	0.7000	1.00
8	Kimonos	314	0.6944	0.90	211	0.7000	1.00
9	One Piece Dresses	3,648	0.6925	0.87	3,019	0.7000	1.00
10	Costumes	443	0.6995	0.94	357	0.7000	1.00
11	Swimwear	308	0.6896	0.91	261	0.7000	1.00
12	All-in-One	553	0.7074	0.94	474	0.7000	1.00

Table 5: Number of extracted synonym pairs and precision of random 100 samples in each sub-category

6. Results

Several examples of phonetic similarity scores from the results of Experiment I and II are shown in Table 4. The scores returned by each method are scaled to the range [0,1]. We used the metrics of precision and recall, and Table 5 shows the performance of both LD and SD for each experiment. We used a phonetic similarity measure technique to extract synonym candidates, and extracted 23,844 pairs of synonyms for the baseline, with an average precision of 0.9208, and about 19,413 pairs of synonyms in Experiment II with a high precision of 0.9983 on average. In each experiment, we applied a joint threshold of 0.875 for SDX-LD and 0.9999 for SDX-SD for the baseline Experiment I, and a joint threshold of 0.85 for CLSG-LD and 0.9999 for CLSG-SD in Experiment II.

7. Discussion

The proposed methodology aims to produce synonym word pairs that are not found in the existing dictionaries of Rakuten Ichiba. We therefore focused on extracting as many synonyms as possible, whereas the results should exclude the synonyms from existing dictionaries.

Paring Words/Phrases

In our test data, item titles were mix-encoding strings which form pairs of English and Japanese words or phrases. We developed an approach utilizing the phrase detection function of the Genism library to pair words or phrases (Mikolov et al., 2013a, 2013b). This technique greatly reduced the computational cost of generating all possible pairs in each test category dataset.

Phonetic Coding

Although the various language scripts are written in Latin/Romanized scripts, the spelling does not always correspond directly to the pronunciation. Because loanwords are generally written in Katakana/Romaji and are pronounced using Japanese pronunciation rules and Japanese syllables, there may be many variations in spelling for the same transliteration. In this experiment, we focused on extracting not only transliteration between English and Katakana, but also between English and Romaji, Hiragana and Kanji. A novel approach based on CLSG helped to increase the precision and reduce the parameter of the learning process.

Measuring/Learning

Normalizing the LD value makes it easy to determine a threshold of best-N extraction from the results. LD can be applied rapidly to diverse information retrieval (IR) tasks. In our previous work, SD learned a one-to-one form of bilingual word pairs (e.g., platinum|プラチナ), whereas now it can learn phrases/segments, for example “v-neck pullover deck shirts|vネックプルオーバーデッキシャツ”.

Thresholding

The allocation of a threshold is a key to differentiate synonym and non-synonym pairs. In this experiment, we manually set a reasonable value for the threshold for each method, and then evaluated the precision of a randomly selected 100 synonym pairs in the final step (i.e., after excluding synonym pairs from the existing dictionaries). Although the use of joint thresholds in each experiment optimized the synonym extraction task, the allocation of thresholds had to be done manually. Automatic allocation should therefore be considered in the future.

Fuzzy Ranking

Due to frequent co-occurrences (words/phrases) in the paring process, some incorrect pairs appeared as one or more unnecessary characters in addition to the words. For example, if we applied an individual threshold of 0.9999 to CLSG-SD, this kind of error could be avoided; otherwise, the fuzzy score can be satisfied to eliminate these incorrect pairs (See Table 6).

Synonym pairs	CLSG-SD	Fuzzy rank
dub_collection|ダブコレクションリング	0.997029	41
dub_collection|ダブコレクション	**0.999999**	**48**
dub_collection|ダブコレクションダブ	0.999389	42
dub_collection|ダブコレクションレディース	0.997871	39

Table 6: Examples of error pairs and scores in CLSG-SD and fuzzy ranking

Evaluation

For the evaluation, a sample of 100 synonym pairs from each category result was first randomly selected. There were 1,200 synonym pairs for 12 categories in Experiments I and II respectively. Then, each experiment sample set was annotated manually and used to calculate the precision and recall (See Table 5). The results of Experiment II showed improved performance for the CLSG coded dataset over Experiment I (i.e., the baseline), which used the SDX coded dataset (See Figure 3).

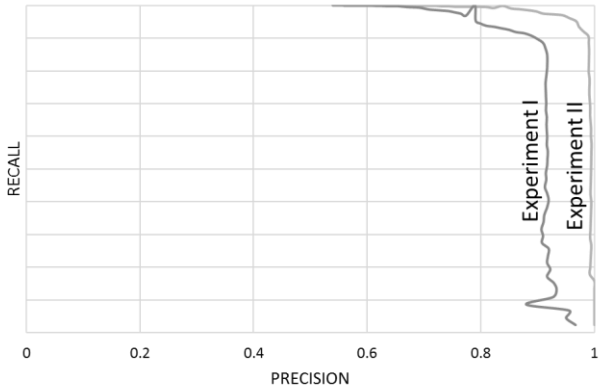

Figure 3: Performance comparison for Experiments I & II

8. Conclusion

We present here a practically oriented approach for the extraction of Japanese synonyms based on phonetic similarity, with high precision. Our test datasets are not bilingual pairs, and the generation of candidate pairs therefore posed a challenge at the early stages, since we do

not want to omit any possible pairs in the generation process. Integration of the phrase detection module of genism reduced the computational cost and maximized the coverage of bilingual candidate pairs. However, SD learning improved from one-to-one word pairs to one-or-more phrases, and the probabilistic scores of synonyms were higher than in previous studies. In future work, we aim to investigate ways of optimizing the learning parameter of the SD model. Allocation of the thresholding process also requires improvement. Experiment II achieved high values for precision. In the future, we intend to develop a deep learning neural network model integrated with a phonetic concept to enhance the performance. We also aim to extend our system to extract the synonym pairs in other languages.

9. References

Finch, A., Harada, T., Tanaka-Ishii, K., and Eiichiro Sumita (2017). Inducing a Bilingual Lexicon from Short Parallel Multiword Sequences. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 16(3), Article 15DOI: https://doi.org/10.1145/3003726

Finch, A. M., Htun, O., and Sumia, E. (2012). The NICT translation system for IWSLT 2012. In Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT"12). 121–125.

Ristad, E. S., and Yianilos, P N. (1998). Learning string edit distance. IEEE Transactions on Pattern Recognition and Machine Intelligence, 20(5):522–532.

Och, F. J., and Ney, H. (2003). A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19–51.

Sajjad, H., Fraser, A., and Schmid, H. (2012). A statistical model for unsupervised and semisupervised transliteration mining. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 1:469–477. http://www.aclweb.org/anthology/P12-1049

Richardson, R., Nakazawa, T., and Kurohashi, S. (2014). Bilingual dictionary construction with transliteration filtering. In Proceedings of LREC 2014, Ninth International Conference on Language Resources and Evaluation, 1013–1017

Tsuji, K., Daille, B., and Kageura, K. (2002). Extracting French-Japanese word pairs from bilingual corpora based on transliteration rule. In Proceedings of 3rd LREC, pp. 499–502.

Yusukawa, M., Culpepper, J. S., and Scholer, F. (2012). Phonetic matching in Japanese. In Proceedings of SIGIR 2012 Workshop on Open Source Information Retrieval (OSIR2012), Portland, Oregon, USA, 68–71. http://opensearchlab.otago.ac.nz/

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a) Efficient estimation of word representations in vector space. ArXiv13013781Cs. Available: http://arxiv.org/abs/1301.3781, accessed 11 June 2017

Mikolov, T., Yih, W.-T., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. HLT-NAACL.

Htun, O., Shigeaki, K., and Mikami, Y. (2011). Cross-Language Phonetic Similarity Measure on Terms Appeared in Asian Language. International Journal of Intelligent Information Processing, 2(2).

Htun, O., Finch, A., Sumita, E., and Mikami, Y. (2012). Improving transliteration mining by integrating expert knowledge with statistical approaches. International Journal of Computer Applications, 58.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin 4, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In ACL 2007.

Shigeaki, K (2010). String edit distance for computing phonological similarity between words. In Proceedings of the International Symposium on Global Multidisciplinary Engineering.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Journal of Soviet Physics Doklady, 10(8):707–709.

Wu, X. (2013). Mining Japanese compound words and their pronunciations from web pages and tweets. In Proceedings of the International Joint Conference on Natural Language Processing, Nagoya, Japan, 14-18 October 2013, 849–853.

Xianchao Wu, "Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets", In proceedings of International Joint Conference on Natural Language Processing, pages 849–853, Nagoya, Japan, 14-18 October 2013.