Bilingual Dictionary Construction with Transliteration Filtering

John Richardson[†], **Toshiaki Nakazawa**[‡], **Sadao Kurohashi**[†]

[†]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[‡]Japan Science and Technology Agency, Chiyoda-ku, Tokyo 102-8666, Japan
john@nlp.ist.i.kyoto-u.ac.jp, nakazawa@pa.jst.jp, kuro@i.kyoto-u.ac.jp

Abstract

In this paper we present a bilingual transliteration lexicon of 170K Japanese-English technical terms in the scientific domain. Translation pairs are extracted by filtering a large list of transliteration candidates generated automatically from a phrase table trained on parallel corpora. Filtering uses a novel transliteration similarity measure based on a discriminative phrase-based machine translation approach. We demonstrate that the extracted dictionary is accurate and of high recall (F_1-score 0.8). Our lexicon contains not only single words but also multi-word expressions, and is freely available. Our experiments focus on Katakana-English lexicon construction, however it would be possible to apply the proposed methods to transliteration extraction for a variety of language pairs.

Keywords: transliteration, lexicon, Katakana

1. Introduction

A large lexicon of technical terms is important for many natural language processing applications, particularly machine translation. The aim of this research is to generate a large Japanese-English dictionary for the scientific domain.

The great majority of technical terms in Japanese are transliterations of English words. It is therefore an interesting option to consider designing a system specifically for transliteration extraction, as this allows us to improve the accuracy greatly by making use of a transliteration model, while sacrificing only minimal coverage. Although we concentrate on Japanese-English in this paper, our model could easily be extended to generate dictionaries for other languages that contain many transliterations, such as Arabic, Korean and Russian.

Transliterations in the Japanese language are written with a special writing system ('Katakana') used primarily for transliterations, and this makes it relatively simple to find likely transliterations. There can be a number of spelling variations for the same transliteration. For example both *konpyuuta* コンピュータ and *konpyuutaa* コンピューター are valid transliterations of 'computer'. There is however often a preferred spelling.

Our method is to use a discriminative extension to phrase-based statistical machine translation (PBSMT) to filter a list of transliteration candidates. This model is able to score transliterations with high accuracy. The candidates to be filtered are generated automatically by using a standard PBSMT system to extract phrase pairs from a parallel corpus.

It can be observed that alignment accuracy is relatively high for transliterations compared to other words, as a result of their rarity and relatively low ambiguity. We believe therefore that the use of a phrase table is a good choice for a source of high quality transliteration candidates. Another motivation for the use of phrase-based SMT is that it is possible to consider multi-word expressions.

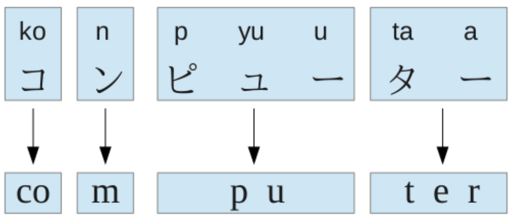

Figure 1: Example of Katakana–English transliteration phrase alignment.

2. Transliteration Model

We begin by constructing a transliteration model that can be used to score candidate transliteration pairs. A standard PBSMT decoder is designed to find the best translation of an input sentence, however we require a discriminative model to measure the likelihood that a candidate translation pair is correct. We therefore design a discriminative filter that can make use of phrase translation probabilities.

The model learns probabilities $P(f \mid e)$ of obtaining the character sequence $f = f_1, ..., f_m$ by transliterating the sequence $e = e_1, ..., e_n$. We wish to learn probabilities such as those shown in Table 1.

The probabilities are learned from a set of transliteration pairs using a grapheme-based machine translation model. We model transliteration as a phrase-based machine translation task on a character rather than word level, treating character groups as phrases. The model is trained by learning phrase alignments such as that shown in Figure 1 from a training corpus (for details see Section 4.1.).

The field of phrase-based SMT has been well studied

f		e	$P(f \mid e)$
チェ	*che*	che	0.463
シュ	*shu*	che	0.122
シェ	*she*	che	0.082
ケ	*ke*	che	0.075
ヒェ	*he*	che	0.038
チ	*chi*	che	0.028
ッシュ	*sshu*	che	0.027
ケー	*kee*	che	0.024
ヒャ	*hya*	che	0.014
チェー	*chee*	che	0.008

Table 1: Examples of transliteration probabilities from PBSMT model.

and there exists a standard toolset enabling the construction of such a model. Character alignment was performed by GIZA++ (Och and Ney, 2003) with the 'grow-diag-final' heuristic for training. We used the default configuration of the Moses (Koehn et al., 2007) tuning pipeline with the distortion limit set to 1 (as transliteration requires monotonic alignment) to generate a phrase table for transliteration. The phrase translation probabilities could then be extracted directly from the phrase table.

2.1. Scoring Metric

In order to measure the likelihood that an input word pair is a correct transliteration, we define a transliteration likelihood score. This score makes use of the probabilities (such as in Table 1) of a Japanese character sequence f given some English character sequence e, which we denote as $P(f \mid e)$.

The score of a candidate pair is calculated by considering each possible segmentation of the English and Japanese words s and t into n character groups $s_1, ..., s_n$ and $t_1, ..., t_n$, then calculating the product of the translation probabilities of each corresponding character sequence, i.e. $\prod_{i=1}^{n} P(t_i \mid s_i)$. It is necessary to find the optimal segmentation a (containing $|a|$ character groups), giving the final score as:

$$\max_a \prod_{i=1}^{|a|} P(t_{a,i} \mid s_{a,i}) \tag{1}$$

In contrast to HMM-based models, we do not consider the transition probability $P(s_i \mid s_{i-1})$ as it is given that the candidates are actual words (they were extracted from a bilingual corpus). That is to say we make the assumption that $P(t_i \mid s_i)$ does not depend on s_{i-1} or t_{i-1}, which in practice is true if the segments are sufficiently long.

Naturally the set of all possible segment pairs is very large, and therefore some of the longer and rarer character sequences do not appear in the training data. These segment pairs will yield zero transliteration probabilities. This issue is unavoidable, however it is not such a serious problem as the model considers the combination of shorter character groups. This achieves an effect similar to backoff and interpolation smoothing methods used in language modeling.

2.2. Finding an Optimal Segmentation

In Equation 1 we require the evaluation of every possible segmentation. It is however impractical to use brute-force search for long candidates, as the number of possible segmentations is of the order $O(4^n)$ for a candidate pair of length (m, n) $(m > n)$.

To improve the efficiency of score computation, we designed a k-best greedy segmentation algorithm. Beginning with the entire words e and f, we find the k-best prefix segment pairs $(e_{1,1}, f_{1,1}), ..., (e_{1,k}, f_{1,k})$. The remainder of the words are then split in a similar way by finding the k-best prefix segments. This is repeated until the candidate pair has been fully segmented. See Figure 2 for an illustration of this algorithm.

The complexity of the segmentation problem can now be reduced to $O(k^n)$ by considering the k-best prefixes at each step or $O(kn)$ by keeping a global k-best list. While this is an approximate algorithm, it will always give a lower bound, which is suitable for filtering out unlikely candidates. This bound can be improved by using larger k.

In a practical implementation we can further increase the speed by using the cutoff C that will later be used for filtering the candidates. Since it is always the case that $score(t_1, ..., t_i \mid s_1, ..., s_i) \geq score(t_1, ..., t_i, t_{i+1} | s_1, ..., s_i, s_{i+1})$ (since $P(t_{i+1} | s_{i+1}) \leq 1$), we can skip any partial segmentations with scores $\leq C$. This works comfortably for words of practical length (under ~40 characters).

3. Filtering

We now introduce our filtering process. We first generate a large list of candidate translation pairs, then filter these candidates with the transliteration likelihood score defined in Section 2.1..

The candidates are generated automatically by using the standard PBSMT pipeline to align and extract translation phrase pairs from a large parallel corpus. We consider only the phrase pairs containing transliterations in the Japanese side by removing entries that contain non-Katakana characters. Examples of extracted candidates are shown in Table 2. A benefit of using phrase-based SMT is that we are able to consider multi-word expressions and not only single words.

Finally, the dictionary is created by filtering the candidates by score. We take $C = 0.1$ as the cutoff, including all candidates scoring higher than this threshold. For any given word we keep all possible translations that score higher than the cutoff in order to include all valid spelling variations. This is important for Japanese, where there can be a number of such variations appearing in unedited texts, such as on the web. The value 0.1 was chosen to ensure a very high precision, however this could be reduced if such a clean dictionary is not required. The ease of selecting the required precision/recall is a benefit of this approach.

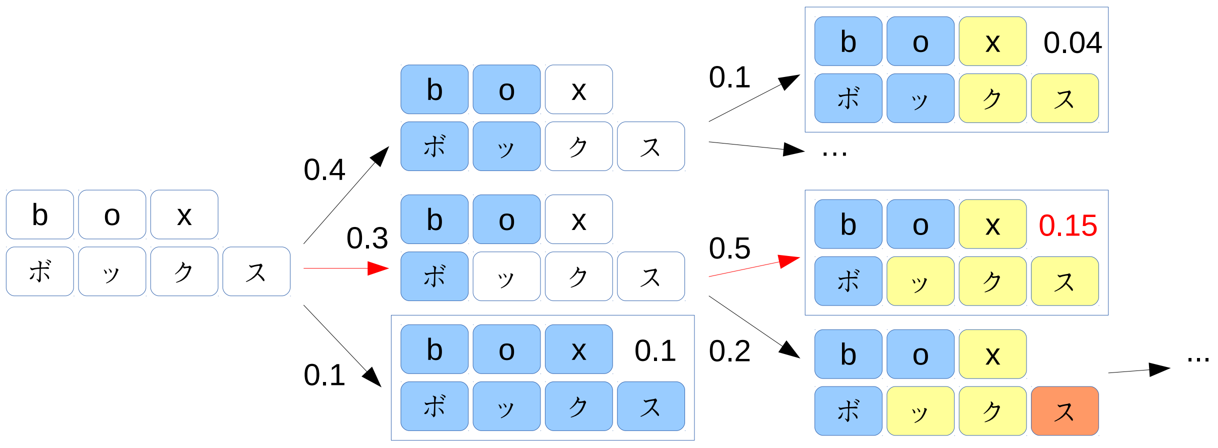

Figure 2: *k*-best greedy segmentation algorithm.

Japanese		English	Correct?
アセチレン マトリックス	*asechiren matorikkusu*	acetylene matrices	Yes
アセチレン メタセシス	*asechiren metaseshisu*	acetylene metathesis	Yes
アセチレン モノテルペン	*asechiren monoterupen*	acetylenic monoterpene	Yes
アセチレン モノマー	*asechiren monomaa*	acetylene derivative having a	No
アセチレン モノマー	*asechiren monomaa*	acetylene derivative having	No
アセチレン モノマー	*asechiren monomaa*	acetylene monomers	Yes
アセチレン リンケージ	*asechiren rinkeeji*	acetylene linkage	Yes
アセチレンカチオン	*asechirenkachion*	-x2.pi.u vibronic-coupling	No

Table 2: Example candidate transliterations from phrase table.

4. Dictionary Construction

The aim of this research is to provide a method of generating a large bilingual lexicon that is simple to implement. We constructed such a lexicon, a dictionary of Katakana-English scientific terminology, in order to evaluate the proposed method. The dictionary was evaluated based on its coverage and accuracy of entries.

4.1. Data Sets

The data used to train the transliteration model in Section 2. consisted of Katakana-English dictionary entries and aligned pairs of Japanese-English Wikipedia article titles (69K entries in total).

The phrase candidates for filtering were extracted from a large parallel corpus of research paper title pairs provided by the Japan Science and Technology Agency. The corpus contained 24M parallel title pairs, generating 7M Katakana-English phrase candidates after removing entries that contained non-Katakana characters.

4.2. Results and Evaluation

Using the parallel corpus above to generate candidates, we were able to build a dictionary containing 170K entries. The extracted dictionary can be obtained for free[1]. Some examples of extracted terms can be seen in Table 3.

In order to evaluate the quality and coverage of our dictionary, we created a test set of candidate word pairs. We selected 1000 candidates generated from the paper abstracts corpus at random and tagged them as 'translation' or 'not-translation' by hand. The extraction algorithm used to generate the full dictionary was run on this test set and the extracted lexicon tested against the reference labels.

The system achieved a precision of 1.000 and recall of 0.667 (F_1-score 0.8) using the cutoff $C = 0.1$. By changing the value of C it would be possible if required to increase the recall by sacrificing some precision. The highest F_1-score was around 0.85 with $C \approx 0.005$. See Figure 3 for the precision-recall curve.

5. Related Work

The automatic construction of bilingual lexicons has been one of the most studied areas in the field of Natural Language Processing in recent years, especially with the hope of harnessing the vast sum of data available on the web. A variety of approaches have been proposed, most of which focus on the extraction of generic lexicons, however in this paper we focus on

[1]http://orchid.kuee.kyoto-u.ac.jp/~john/files/lrec2014.tar.bz2

Japanese		**English**	**Score**
フルオロカーボンガス	*furuorokaabongasu*	fluorocarbon gas	0.68
フルオロカーボンシロキサン	*furuorokaabonshirokisan*	fluorocarbon siloxane	0.20
フルオロカーボンポリマー	*furuorokaabonporimaa*	fluorocarbon polymer	0.97
ヘキサフルオロイソプロパノール	*hekisafuruoroisopuropanooru*	hexafluoroisopropanol	0.83
ヘキサフルオロプロピレン	*hexisafuruoropuropiren*	hexafluoropropylene	0.22
ヘキサフルオロベンゼン	*hekisafuruorobenzen*	hexafluorobenzene	0.43

Table 3: Example entries from extracted lexicon.

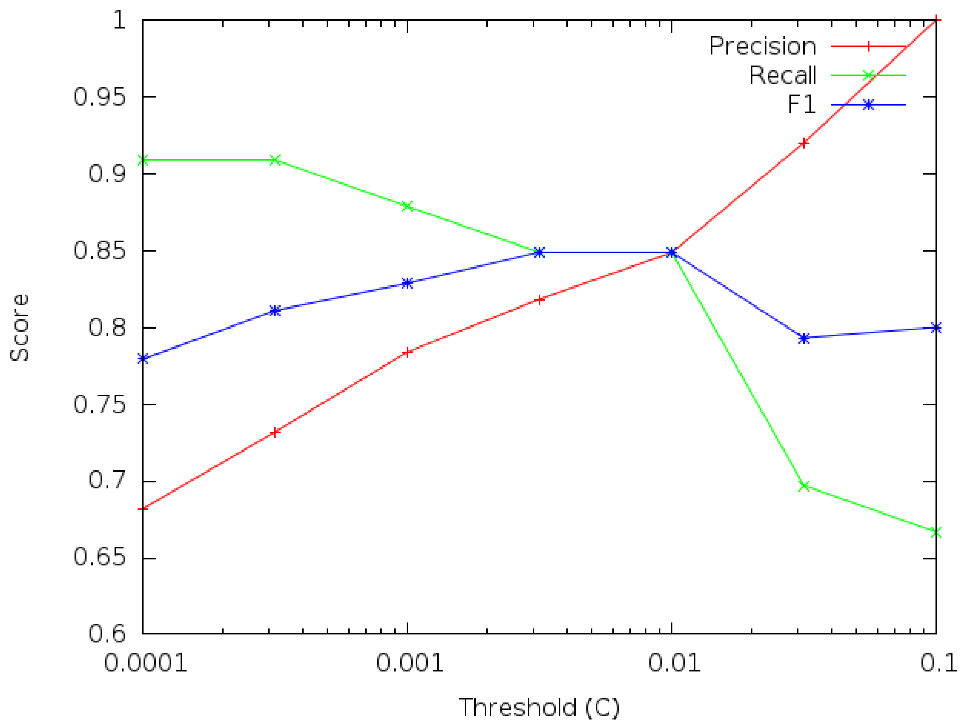

Figure 3: Precision-recall curve for various thresholds (C).

the subtask of transliteration extraction.

There has been much work on the task of transliteration generation, i.e. predicting the correct transliteration of some given input word. This research began with the introduction of Machine Transliteration (Knight and Graehl, 1998), which has progressed to consider a variety of methods ranging from simple edit distance and noisy-channel models (Brill et al., 2001) to conditional random fields (Ganesh et al., 2008) and finite state automata (Noeman and Madkour, 2010).

The PBSMT-based transliteration system described in Section 2. was implemented as specified in previous work on transliteration extraction such as Richardson et al. (2013) (Japanese and Korean), Matthews (2007) (Chinese and Arabic), and Antony et al. (2010) (Kannada).

The task of identifying and extracting correct transliteration pairs however is less explored. This classification problem is the key to using filtering for bilingual lexicon extraction, and we have proposed a novel approach to score transliteration candidates.

6. Conclusion and Future Work

In this paper we have introduced an approach to extracting a large dictionary of Japanese-English technical terms using a transliteration-based approach. The core of our method is a discriminative transliteration model based on the framework of statistical machine translation.

Analysis of our extraction lexicon show that transliteration-based filtering is able to generate resources with high precision and recall. It is possible to extract not only single words, as is the case in many previous studies, but also multi-word expressions. An additional benefit of our approach is that is it simple to adjust the required precision/recall ratio by setting the cutoff.

In the future we would like to extend our system to extract dictionaries for other language pairs, such as Chinese-English, using a similar transliteration model. The identification of transliterations is non-trivial for some languages, so this is a further consideration to be made in future work. It is also important to consider how to generate as large a candidate list as possible, ideally from non-parallel data, as this is the main factor in determining the size of the extracted dictionary.

Acknowledgments

We would like to thank the Japan Science and Technology Agency for the use of their bilingual research paper title pairs corpus. The first author is supported by a Japanese Government (MEXT) research scholarship. Thank you also to the anonymous reviewers for their useful feedback.

7. References

P.J. Antony, V.P. Ajith, and K.P. Soman 2010. Statistical Method for English to Kannada Transliteration. *BAIP 2010, CCIS 70, pp. 356–362.*

Eric Brill, Gary Kacmarcik and Chris Brockett. 2001. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*, 2001.

Surya Ganesh, Sree Harsha, Prasad Pingali, Vasudeva Varma. 2008. Statistical Transliteration for Cross Language Information Retrieval using HMM alignment model and CRF. In *2nd International Workshop on Cross Language Information Access, IJCNLP 2008.*

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007.*

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics*, 24.

David Matthews. 2007. Machine Transliteration of Proper Names. *Masters Thesis, School of Informatics, University of Edinburgh.*

Sara Noeman and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework.
In *Proceedings of the 2010 Named Entities Workshop.*

Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. In *Computational Linguistics 2003.*

John Richardson, Toshiaki Nakazawa and Sadao Kurohashi. 2013. Robust Transliteration Mining from Comparable Corpora with Bilingual Topic Models. In *IJCNLP 2013.*