Mining Japanese Compound Words and Their Pronunciations from Web Pages and Tweets

Xianchao Wu
Baidu Inc.
wuxianchao@{gmail, baidu}.com

Abstract

Mining compound words and their pronunciations is essential for Japanese input method editors (IMEs). We propose to use a chunk-based dependency parser to mine new words, collocations and predicate-argument phrases from large-scale Japanese Web pages and tweets. The pronunciations of the compound words are automatically rewritten by a statistical machine translation (SMT) model. Experiments on applying the mined lexicon to a state-of-the-art Japanese IME system[1] show that the precision of Kana-Kanji conversion is significantly improved.

1 Introduction

New compound words are appearing everyday. Person names, technical terms and organization names are newly created and used in Web pages such as news, blogs, question-answering systems. Abbreviations, food names and event names are formed and shared in Twitter and Facebook. Mining of these new compound words, together with their pronunciations, is an important step for numerous natural language processing (NLP) applications. Taking Japanese as an example, the lexicons containing compound words (in a mixture of Kanjis and Kanas) and their pronunciations (in a sequence of Kanas) significantly influence the accuracies of speech generation (Schroeter et al., 2002) and IME systems (Kudo et al., 2011). In addition, monolingual compound words are shown to be helpful for bilingual SMTs (Liu et al., 2010).

In this paper, we mine three types (Figure 1) of new (i.e., not included in given lexicons) Japanese compound words and their pronunciations: (1) *words*, which are combinations of single characters and/or shorter words; (2) *collocations*, which are combinations of words; and (3) *predicate-argument phrases*, which are combinations of chunks constrained by semantic dependency relations. The sentences were parsed by a state-of-the-art chunk-based Japanese dependency parser, Cabocha[2] (Kudo and Matsumoto, 2002a) which makes use of Mecab[3] with IPA dictionary[4] for word segmenting, POS tagging, and pronunciation annotating.

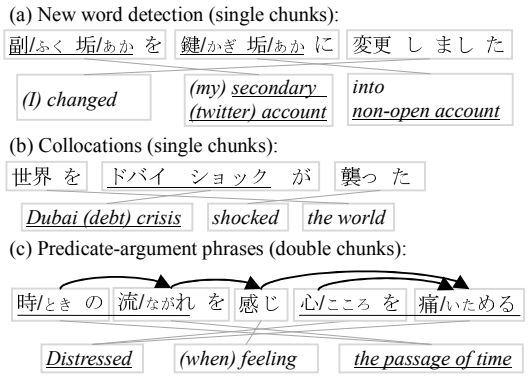

Figure 1: Examples of new (compound) words.

The first sentence in Figure 1 contains two new words which were not correctly recognized by Mecab. We call them "new words", since *new* semantic meanings are generated by the combination of single characters. There is one Kana collocation in the second sentence. Different from many former researches (Manning and Schütze, 1999; Liu et al., 2009) which only mine collocations of two words, we do not limit the number of words in our collocation lexicon. The third sentence contains two predicate-argument phrases of noun-noun modifiers and object-verb relations.

The main contribution of this paper is that the

[1] freely downloadable from www.simeji.me for Android and http://ime.baidu.jp/type/ for Windows

[2] http://code.google.com/p/cabocha/

[3] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

[4] http://code.google.com/p/mecab/downloads/detail?name=mecab-ipadic-2.7.0-20070801.tar.gz

International Joint Conference on Natural Language Processing, pages 849–853,
Nagoya, Japan, 14-18 October 2013.

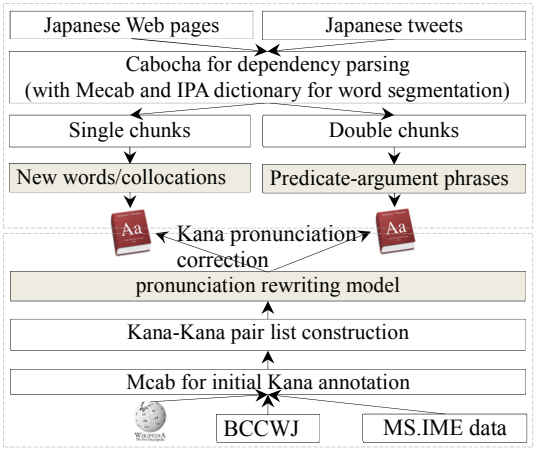

Figure 2: The lexicon mining processes.

well studied *chunk-level dependency technique* is firstly (as far as our knowledge) adapted to compound word mining. The proposed mining method has the following three parts. First, it explicitly utilize the chunk identification features and frequency information for detecting new words and collocations. Second, chunk-level semantic dependency relations are employed for determining predicate-argument phrases. Third, a Kana-to-Kana pronunciation rewriting model based on phrasal SMT framework is proposed for correcting Kana pronunciations of the compound words.

2 Compound Word Mining

Figure 2 shows our major lexicon mining process: lexicon mining in a top-down flow and pronunciation rewriting in a bottom-up flow.

2.1 Mining single chunks

Definition 1 (**Japanese chunk**) Suppose **w** being the Japanese vocabulary set, a Japanese chunk is defined as a sequence of contiguous words, $C = w_n^+ w_p^*$, where $w_n^+ \in \mathbf{w}$ is a sequence of notional words with no less than one w_n, and $w_p^* \in \mathbf{w}$ contains zero or more particles w_p. New words and collocations come from w_n^+ without w_p^*.

This mining idea is based on the fact that an Japanese morphological analyser (e.g., Mecab) tends to split one out-of-vocabulary (OOV) word into a sequence of known Kanji characters. The point is that, most of the known Kanji characters are annotated to be notional words such as nouns. Consequently, Cabocha, which takes words/characters and their POS tags as features for discriminative training using a SVM model (Kudo and Matsumoto, 2002b), can still *correctly* tend to include these single-Kanji-character words into one chunk. Thus, we can re-combine the wrongly separated pieces into one (compound) word.

	Frequency $\geq$ 20	**Frequency $\geq$ 500**
single chunk (web)	9,823,176	685,363
double chunks (web)	20,698,683	794,605
single chunk (twitter)	156,506	6,131
not in web	21,370 (13.7%)	492 (8.0%)
double chunks (twitter)	160,968	2,446
not in web	35,474 (22.0%)	443 (18.1%)

Table 1: The number of compound words mined.

2.2 Mining predicate-argument phrases

Definition 2 (**Predicate-argument phrase**) A predicate-argument phrase is defined as a labelled graph structure, $A = \langle w_h, w_n, \tau, \rho \rangle$, where $w_h, w_n \in \mathbf{w}$ are a predicate and an argument word (or chunk) of the dependency, τ is a predicate type (e.g., transitive verb), and ρ is a label of the dependency of w_h and w_n. We append one constraint during mining: w_h and w_n are adjacent. That is, the phrases mined are all contiguous without gaps. The predicate-argument phrases mined in this way is helpful for context-based Kana-Kanji conversion of Japanese IME.

Japanese is a typical Subject-Object-Verb language. The direct object phrase normally appears before the verb. For example, for two input Kana sequences "やさいをいためる" (野菜/vegetables を/particle 炒める/cooking: stir-fried vegetables) and "こころをいためる" (心/heart を痛める/hurt: hurt ones heart), even "いためる" takes the similar keyboard typing, the first candidate Kanji words are totally different. The users will be angry to see the candidate of "心を炒める" (stir-fried heart) for "こころをいためる". It is the pre-verb objects that determines the dynamic choosing of the correct Kanji verbs.

2.3 Experiments on compound word mining

We use two data sets for compound word mining. The first set contains 200G Japanese Web pages (1.9 billion sentences) which were downloaded by an in-house Web crawler. The second set contains 44.7 million Japanese tweets (28.8 words/tweet) which were downloaded by using an open source Java library twitter4j[5] which implemented the Twitter Streaming API[6].

[5]http://twitter4j.org/ja/index.html
[6]https://dev.twitter.com/docs/streaming-apis

Lexicons	Frequency ≥ 20	Precision
alignment method	2,562	76.5%
single chunk	16,673	93.0%
double chunks	9,099	91.5%

Table 2: The number of entries and precisions of the alignment method (Liu et al., 2009) and our approach, using 2M sentences.

Table 1 shows the statistics of the single/double chunk lexicons (of frequencies $\geq$ 20 or 500). We compared the novel entries included in the twitter lexicons but not the web. The ratio ranges from 8.0% to 22.0%, reflecting a special bag of compound words used in tweets instead of the traditional web pages.

We compare our lexicons with two baselines, one is the C-value approach (Frantzi and Ananiadou, 1999) with given POS sequences and the other is the monolingual word alignment approach (Liu et al., 2009). We ask Japanese linguists to give a POS sequence set with 128 rules for compound word mining. Applying C-value approach with these rules to the 200G web data yields a lexicon of 884,766 entries (frequency $\geq$ 500). Our single (double) chunk lexicon shares around 30% (7%) with this lexicon. This lexicon is used in our baseline Japanese IME system (Table 5).

During our re-implementation of the alignment approach, we found that the EM algorithm (Dempster et al., 1977) for word aligning the 1.9 billion sentences is too time-consuming. Instead, we only used the first 2M sentences (28.4 words/sentence) of the web data for intuitive comparison. The statistics are shown in Table 2. The precisions are computed by manually evaluating the top-200 entries (with higher frequencies) in each lexicon. The lexicons mined by our approach outperforms the baseline in a big distance, both precision and the number of entries successfully mined.

3 Pronunciation Rewriting Model

Our pronunciation rewriting model mapping from the compound words' original pronunciations to their correct pronunciations. It is a generative model based on the phrasal SMT framework. We limit the model monotonically rewrite initial Kana sequences to their correct forms without reordering. We use Moses[7] (Koehn et al., 2007) to implement this model by setting the source and target sides to be Kana sequences.

[7]http://www.statmt.org/moses/

The Kana-Kana rewriting model improves the traditional Kanji-Kana predication models (Hatori and Suzuki, 2011) in the following aspects. First, data sparseness problem of Kanji-Kana approach can be mitigated in a sense, since the number of Kanas in Japanese is no more than 50 yet the number of Kanjis is tens of thousands. Second, Kana-Kana pairs are easier to be aligned with each other, since most Kanjis are pronounced by no less than two Kanas and consequently the number of Kanas almost doubles the number of Kanjis in the experiment sets. Finally, the entries in the final lexicons contain two Kana pronunciations, before and after correcting. We argue this is helpful to improve the user experiences of IME systems where we need to cover the users' typing mistakes.

3.1 Mining Kanji-Kana entries from Wiki

For training the rewriting model, we mine a Kana-Kanji lexicon from parenthetical expressions in Japanese Wikipedia pages[8], a high quality collection of new words. The only problem is to determine the pre-brackets Kanji sequence that exactly corresponds to the in-bracket Kana sequence.

Our method is inspired by (Okazaki and Ananiadou, 2006; Wu et al., 2009). They used a term recognition approach to build monolingual abbreviation dictionaries from English articles (Okazaki and Ananiadou, 2006) and to build Chinese-English abbreviation dictionaries from Chinese Web pages (Wu et al., 2009). For locating a textual fragment with a Kanji sequence and its Kana pronunciation in a pattern of "Kanji sequence (Kana sequence)", we use the heuristic formula:

$$\mathrm{LH}(c) = \mathrm{freq}(c) - \sum_{t \in T_c} \mathrm{freq}(t) \times \frac{\mathrm{freq}(t)}{\sum_{t \in T_c} \mathrm{freq}(t)}.$$

Here, c is a Kanji candidate (sub-)sequence; freq(c) denotes the frequency of co-occurrence of c with the in-brackets Kana sequence; and T_c is a set of nested Kanji sequence candidates, each of which consists of a preceding Kanji or Kana character followed by the candidate c.

Table 3 shows the number of entries mined by setting the LH score to be $\geq$ 3, 4, or 5. From the table, we observe that as LH threshold is added by one, the number of entries is cut nearly a half. For each entry set, we further randomly selected 200 entries and checked their correctnesses by

[8]All the Japanese pages until 2012.06.03 were used. Examples can be found in http://ja.wikipedia.org/wiki/三日月

LH $\geq$	# of Entries	Precision
3	42,423	95.0%
4	18,348	95.5%
5	10,234	96.0%

Table 3: Kanji-Kana entries mined from Wiki.

System	Prec.	BLEU-4	src/trg	Data	Train/Dev/Test
baseline	70.2%	0.8663	4.9/7.0	bcc-	25.3k/0.5k/0.5k
Ours	90.4%	0.9687	7.0/7.0	wj	
baseline	49.8%	0.6734	2.8/4.9	wiki	17.3k/0.5k/0.5k
Ours	62.2%	0.7380	4.9/4.9		
baseline	43.5%	0.9504	58.0/78.1	ms	5.6k/0.2k/0.2k
Ours	62.0%	0.9737	80.7/78.1		

Table 4: Pronunciation predication accuracies.

hand. The precisions ranges from 95% to 96%. Moreover, this mining approach can make use of parenthetical expressions appearing in not only Wikipedia but also the total Japanese Web pages.

3.2 Experiments on pronunciation rewriting

As shown in Figure 2, we use three data sets for training our pronunciation rewriting model. The first set is a Kanji-Kana compound lexicon collected from the 2009 Core Data of the Balanced Corpus of Contemporary Written Japanese (BCCWJ) corpus (Maekawa, 2008). The second is the Microsoft Research IME data[9] (Suzuki and Gao, 2005). The third set is the Wikipedia Kana-Kanji lexicon with LH ≥ 4 (Table 3).

The precisions and BLEU-4 scores (Papineni et al., 2002) of the baseline system (Hatori and Suzuki, 2011) and our approach are shown in Table 4. The baseline system takes character-level translation units. From Table 4, we observe that the number of Kanas is larger than the number of Kanjis while the number of initial Kanas and corrected Kanas are almost the same. Our approach yield significant improvements ($p < 0.01$) in both precisions and BLEU-4 scores.

4 Japanese IME Evaluation

As an application-oriented evaluation, we finally integrate the mined lexicons (as a cloud service) into a state-of-the-art Japanese IME system. The system is constructed based on the n-pos model (Mori et al., 1999; Komachi et al., 2008; Kudo et al., 2011). For training the n-pos model, we used 2.5TB Japanese Web pages as the training data. We run Mecab on Hadoop[10], an open source software that implemented the Map-Reduce framework (Dean and Ghemawat, 2004), for parallel word segmenting and POS tagging the data.

IME	Top-1	Top-3	Top-5	Top-9	Test Set
baseline	38.93%	63.76%	70.47%	74.50%	twitter.net
+twitter	48.99%	70.47%	73.15%	75.17%	
baseline	50.16%	75.10%	82.99%	87.46%	JDMWE
+web	52.01%	78.61%	85.34%	89.07%	
baseline	56.23%	84.29%	91.18%	93.80%	Nagoya
+web	58.16%	84.65%	92.01%	94.35%	

Table 5: The top-n precision improvements of appending the mined twitter/web lexicons to a baseline IME system.

For testifying the lexicons mined from the 200G Web data and from the tweets, we respectively use three test sets: (1) "twitter.net" with 149 entries which is a manually collected Twitter new word lexicon[11]; (2) partial "JDMWE" (Shudo et al., 2011) lexicon with 2,169 entries; and (3) "Nagoya" compound word lexicon[12] with 3,628 entries such as idioms.

The top-n (=1, 3, 5, 9) precisions are listed in Table 5. In the baseline system, we used the compound lexicon that was mined by the C-value approach using 128 POS sequences. For direct comparison, we replace this compound lexicon respectively by the web and twitter lexicons (frequency ≥ 500). In the twitter.net test set, the precision of the top-1 candidate significantly ($p < 0.01$) improves from 38.93% to 48.99% (+10.06%). In the JDMWE and Nagoya test sets, the web lexicon can also significantly improve the top-1 precisions of around 2% ($p < 0.05$). Through these numbers, we can say that the proposed approach is helpful for improving the accuracies of real-world Japanese IME application.

5 Conclusion

We have proposed an approach for mining new Japanese compound words from single/double chunks generated by a chunk-based dependency parser. Experiments show that the approach works well on mining new words, collocations and predicate-argument phrases from large-scale Web pages and tweets. We achieved significant improvements on top-n precisions when integrating the mined compound words together with their Kana pronunciations into a state-of-the-art Japanese IME system with million level users.

[9] http://research.microsoft.com/en-us/downloads/AF99E662-B77B-4622-ADAA-7AB9F2842B20/default.aspx

[10] http://hadoop.apache.org/

[11] can be downloaded from http://netyougo.com/

[12] http://kotoba.nuee.nagoya-u.ac.jp/jc2/base/list

Acknowledgments

The author thanks the anonymous reviewers for improving the earlier version of this paper.

References

Jeffrey Dean and Sanjay Ghemawat. 2004. Mapreduce: simplified data processing on large clusters. In *Proceedings of OSDI*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

Katerina T. Frantzi and Sophia Ananiadou. 1999. The c-value/nc-value domain independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6:145–179.

Jun Hatori and Hisami Suzuki. 2011. Japanese pronunciation prediction as phrasal statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 120–128, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180.

Mamoru Komachi, Shinsuke Mori, and Hiroyuki Tokunaga. 2008. Japanese, the ambiguous, and input methods (in japanese). In *Proceedings of the Summer Programming Symposium of Information Processing Society of Japan*.

Taku Kudo and Yuji Matsumoto. 2002a. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69.

Taku Kudo and Yuji Matsumoto. 2002b. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL-2002*, pages 63–69. Taipei, Taiwan.

Taku Kudo, Taiyaki Komatsu, Toshiyuki Hanaoka, Jun Mukai, and Yusuke Tabata. 2011. Mozc: A statistical kana-kanji conversion system (in japanese). In *Proceedings of Japan Natural Language Processing*, pages 948–951.

Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2009. Collocation extraction using monolingual word alignment method. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 487–495, Singapore, August. Association for Computational Linguistics.

Zhanyi Liu, Haifeng Wang, Hua Wu, and Sheng Li. 2010. Improving statistical machine translation with monolingual collocation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 825–833, Uppsala, Sweden, July. Association for Computational Linguistics.

Kikuo Maekawa. 2008. Compilation of the kotonoha-bccwj corpus (in japanese). *Nihongo no kenkyu (Studies in Japanese)*, 4:82–95.

Chris Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, May.

Shinsuke Mori, Masatoshi Tsuchiya, Osamu Yamaji, and Makoto Nagao. 1999. Kana-kanji conversion by a stochastic model (in japanese). *Journal of Information Processing Society of Japan*, 40(7).

Naoaki Okazaki and Sophia Ananiadou. 2006. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(22):3089–3095.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Juergen Schroeter, Alistair Conkie, Ann Syrdal, Mark Beutnagel, Matthias Jilka, Volker Strom, Jun Kim, Hong goo Kang, and David Kapilow. 2002. A perspective on the next challenges for tts research. In *Proceedings of 2002 IEEE Workshop on Speech Synthesis*.

Kosho Shudo, Akira Kurahone, and Toshifumi Tanabe. 2011. A comprehensive dictionary of multiword expressions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 161–170, Portland, Oregon, USA, June. Association for Computational Linguistics.

Hisami Suzuki and Jianfeng Gao. 2005. Microsoft research ime corpus. Technical Report MSR-TR-2005-168, Microsoft Research.

Xianchao Wu, Naoaki Okazaki, and Jun'ichi Tsujii. 2009. Semi-supervised lexicon mining from parenthetical expressions in monolingual web pages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 424–432, Boulder, Colorado, June. Association for Computational Linguistics.