Using a Chunk-based Dependency Parser to Mine Compound Words from Tweets

Xianchao Wu
Baidu (Japan) Inc.
Roppongi-Hills Mori-Tower 34F, 6-10-1 Roppongi Minato-ku, Tokyo 106-0032
{wuxianchao}@baidu.com

1 Introduction

New words are appearing everyday in online communication applications, such as Twitter[1]. Twitter is the world's most famous online social networking and microblogging service that enables its users to send/read text-based messages of up to 140 characters, known as "tweets". Due to the facts that tweets are online typed (as fast as possible) within a limited number of characters, tweets are full of hand-made abbreviations and informal words. These facts make a difference between tweets and frequently used texts in regular web pages, such as news, blogs. Consequently, traditional hand-made corpora (in domains such as news) for natural language processing, such as word segmentation, part-of-speech (POS) tagging, parsing, need to be "domain adapted" to be well suitable to tweets. That is, if one Japanese new (compound) word is not successfully recognized by a word segmentation toolkit, we can hardly ensure the word been well covered by a Japanese Input Method Editor (IME) or well translated by a statistical machine translation system.

In this paper, we focus on novel compound word detection from Japanese tweets. we propose a method for mining contiguous compound words from single/double Bensetsus generated by a state-of-the-art chunk-based dependency parser, Cabocha[2] (Kudo and Matsumoto, 2002) which makes use of Mecab[3] with IPA dictionary[4] for Japanese word segmentation, POS tagging, and pronunciation annotating. In this paper, we use Bensetsu to represent one Japanese "chunk", i.e., one central word such as verb or noun, followed by zero or many assistant words such as particles. Bensetsu is specialized for Japanese, and corresponds to words such as "chunk, phrase, clause" in English. The mined compound words with their kana pronunciations and POS tags can be easily applied to n-pos model based Japanese IME systems, such as the freely available Baidu Japanese IME[5] (Chen et al., 2012).

This paper is organized as follows: we describe the detailed mining algorithm in Section 2; experiments and conclusion are given respectively in Section 3 and Section 4.

2 Compound Word Mining

2.1 Mining single Bensetsu

In case of single Bensetsu, compound words are mined by simply remove the particles in the left-hand-side and right-hand-side of the central word(s). Specially, the particle that connects two central words (such as "wo/を" in "yasai/やさい/野菜/vegetables を itameru/いためる/炒める/cooking") will not be trimmed.

This mining idea is based on the fact that Mecab tends to split one out-of-vocabulary (OOV) word which contains several Japanese Kanji individually into several words in which each Kanji character for one word. Yet, for Cabocha, it tends to include these single-Kanji-character words into one Bensetsu. Thus, we can re-combine the wrongly separated pieces into one (compound) word. This

[1] http://twitter.com/
[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
[5] http://ime.baidu.jp/type/?source=pstop

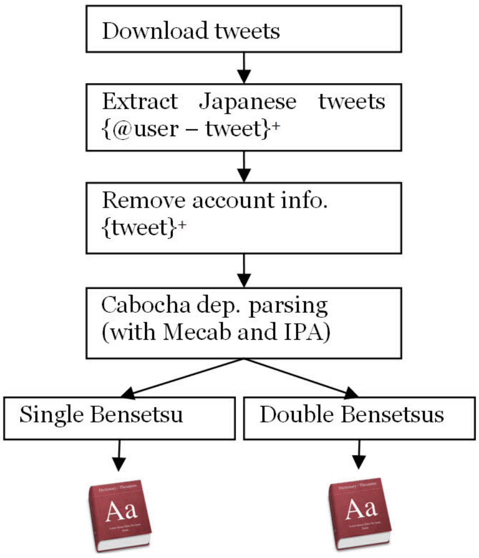

Figure 1: The mining process.

consideration also suitable for other types of compound words, such as personal names. In Mecab, one personal name is frequently separated into two individual words, family name and given name. In Cabocha, family name and given name are frequently re-combined into one Bensetsu. Thus, we can re-combine these two parts into one complete personal name.

The consequent problem is that the Kana pronunciation of the new combined word is not necessary to be the combination of the Kana pronunciations of the old individual words. For example, when two words “kabushiki/かぶしき/株式” (stock) and “kaisya/かいしゃ/会社” (company) are combined together, the result pronunciation is “kabushiki-gaisya/かぶしきがいしゃ”, where “ka/か” is changed into “ga/が”. Another category is that, the new pronunciation has no direct relation to the old individual pronunciations any more. For example, when “ichi/いち/一” (one) and “niti/にち/日” (day) are combined together, the result pronunciation can be “ichiniti/いちにち” (one day) or “tuitati/ついたち” (specially refer to the first day of every month).

2.2 Mining double Bensetsus

In case of double Bensetsus, we only extract compound words from two Bensetsus with dependency relations. That is, one Bensetsu takes as the head (node) and the other takes as the child (node) in the dependency tree. Note that this strategy does not limit the position of the head node, i.e., not matter being the left-hand-side or right-hand-side Bensetsu. Through this mining method, we can easily obtain relatively long distance dependencies, such as determining the verb by given its argument.

Recall that Japanese is a typical Subject-Object-Verb (SOV) language. Thus, the direct object phrase appears before the verb. For example, for two input Kana sequences “yasaiwoitameru/やさいをいためる” (for “yasai/やさい/野菜/vegetables wo/を/particle itameru/いためる/炒める/cooking”, i.e., stir-fried vegetables) and “atamawoitameru/あたまをいためる” (atama/あたま/head wo/を/particle itameru/いためる/痛める/pain, i.e., got a headache), even “itameru/いためる” takes the similar keyboard typing, the first-choose Kanji forms are totally different. The pre-verb objects determines this kind of dynamic choosing of Kanji characters during Japanese IME typing.

2.3 Filtering the lexicons

The original entries mined from sing/double Bensetsus are not guaranteed to be well-formed compound words. We further use the following strategies for filtering the original entries:

- remove compound words start with a stop character/word/POS list, the list includes characters such as “ん, 々, っ, ッ”; words such as “です, ない”; and POSs such as “れんたいし, せつぞくし, じょどうし”;
- remove compound words end with a stop character/word/POS list, the list includes characters such as “っ, ッ”; words such as “よかった, なりたい”; and POSs such as “せっとうじ”;
- compound words are allowed to contain numbers and English letters for compound words such as “YouTube再生リスト, AKB48”.

2.4 The mining process

Figure 1 shows the major mining process. Here, we use the twitter4j package[6], a Java library for the Twitter API, especially the twitter Streaming API[7],

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

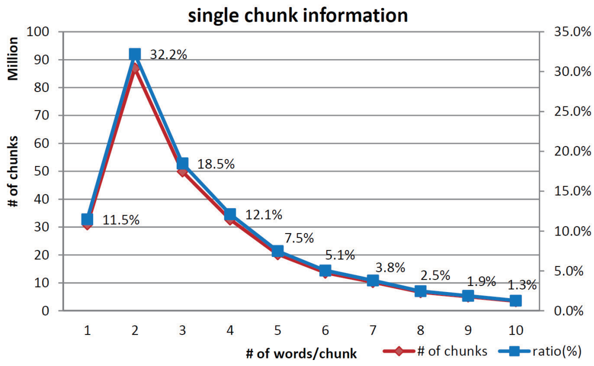

Figure 2: The distribution of the number of words per chunk in tweets.

to download tweets. Since the downloaded tweets starts with user accounts and their tweet sentences, we further get rid of the user account information and only keep the Japanese sentences. Here, we use a greedy strategy to collect Japanese tweets: if at least one katakana or hirakana appears in the tweet, then it is legal. Then, we use Cabocha which integrated Mecab and IPA dictionary for chunk-level Japanese dependency parsing. The single/double Bensetsus in the dependency trees are used to mine compound words. During the mining/generating of final lexicons, the filtering strategies are performed.

3 Experiments

Using the twitter4j package, we downloaded 44,700,736 Japanese sentences (we call this corpus "tweet" hereafter). There are totally 1,287,800,193 words in these sentences, averagely 28.8 words for each sentence. Figure 2 shows the distribution of the number of words per chunk in these Japanese sentences. From the figure, we can observe that 32.2% chunks contain two Japanese words. Chunks that contain from two to four words take a coverage of 62.8% of the total chunks.

In order to verify the novelty of the compound words mined from tweets, we also apply the similar single/double Bensetsus mining algorithm to another 200G data (we call this corpus "200G" hereafter) which are automatically crawled from the Japanese Web (other than those tweets).

Tabel 1 shows the number of compound words mined from tweets' single/double Bensetsus. In order to control the quality of the lexicons, we respectively used 1, 20, and 500 as the frequency thresholds for lexicon filtering. From the table, we can observe that:

	cut.1	cut.20	cut.500
single (200G)	-	9,823,176	685,363
double (200G)	-	20,698,683	794,605
single (tweets)	16,497,474	337,727	15,044
+ filtered	9,048,185	156,506	6,131
+ filtered(-200G)	-	21,370 (13.7%)	492 (8.0%)
double (tweets)	40,030,048	295,541	4,791
+ filtered	19,671,721	160,968	2,446
+ filtered(-200G)	-	35,474 (22.0%)	443 (18.1%)

Table 1: The number of compound words mined from single/double Bensetsus (of the "tweets" data and the "200G" web data), using a threshold of 1, 20, and 500. Here, "filtered" stands for using the post-filtering strategies, "-200G" stands for the entries that are not existing in the corresponding lexicons mined using the 200G web data.

Compound Words	Pronunciation	Description
ツイ廃	ついはい	ツイッター廃人
共感したら RT	きょうかんしたら RT	RT=re-tweet
規制垢	きせいあか	規制されたアカウント
鍵垢	かぎあか	非公開アカウント
女子力高い	じょしりょくたかい	バイドゥ IME の 女子力高い絵文字
福島原発事故	ふくしまげんぱつじこ	福島県の原子力 発電所の事故
復興予算	ふっこうよさん	東日本大震災の復興 に使う政府予算
野田総理	のだそうり	野田佳彦内閣総理大臣
うたぷり	うたぷり	うたのプリンスさま

Table 2: Examples of compound words extracted from single Bensetsu.

- the filtering strategies can remove nearly a half of the entries;
- there are still 8% to 22% of the filtered entries that do not appear in the 200G's lexicons;
- we can averagely mine 16,497,474/44,700,736=0.369 single Bensetsu entries per sentence and 40,030,048/44,700,736=0.896 double Bensetsu entries per sentence. These numbers reflect the large variance of the information contained in tweets.

Table 2 lists several examples of compound words extracted from single Bensetsu. We can observe

Compound Words	Pronunciation	Freq.
人RT	ひとRT	49,134
人全員フォローする	ひとぜんいんふぉろーする	28,558
赤司様	あかしさま	8,501
人rt	ひとrt	7,710
黒バスクラスタさん	くろばすくらすたさん	5,320
超激レアモンスター	ちょうげきれあもんすたー	5,222
赤司くん	あかしくん	5,194
卵ドロップ	たまごどろっぷ	4,840
てらあり	てらあり	4,701
SJペンさん	SJぺんさん	4,358
全力でフォローし	ぜんりょくでふぉろーし	18,473
今繋がってる	いまつながってる	16,301
RTもしくはフォローし	RTもしくはふぉろーし	12,672
RTした人全員フォローする	RTしたひとぜんいんふぉろーする	11,439
わたしの今日	わたしのきょう	11,040
RTした人	RTしたひと	8,049
軽い容量	かるいようりょう	6,387
RTで拡散し	RTでかくさんし	6,034
フォロアーをフォロー	ふぉろあーをふぉろー	5,583
今日の今	きょうのいま	5,198

Table 3: High frequency examples (top-10) of compound words extracted from single/double Bensetsus.

	cut.20	**cut.500**
single/double (tweets)	38.71%	18.06%
+ filtered	30.97%	13.55%
+ filtered(-200G)	12.26%	9.03%

Table 4: The coverage rates of the compound word lexicons to an existing twitter lexicon.

that most of these compound words are abbreviations. Also, the compound words can briefly be separated into two categories. One category includes compound words that are strongly related to twitter service, such as "ツイ∼, ∼RT, ∼垢". The other category includes compound words that are strongly related to a special period, such as "女子力高い(文字)/girls powerful (face-style characters)"[8]. Easy to say that these *hot* compound words can be dynamically mined from tweets and sent to the IME users everyday.

We further lists the top-10 (sorted by frequency) compound words mined from single/double Bensetsus in Table 3. Since we distinguish from uppercases to lowercases, words of "人RT" and "人rt" are taken as different compound words. One interesting thing in this table is that, most high frequency words contain both kana/kanji and English abbreviations, such as "RT, rt, SJ".

Besides these closed tests, we also use an existing twitter lexicon to testify the lexicons mined. The ex-

[8] http://ime.baidu.jp/type/lp/girlspower_kaomoji/

	Top1	**Top3**	**Top5**
baseline IME	38.93%	63.76%	70.47%
+ single/double Bensetsus	48.99%	65.77%	70.47%

Table 5: The top 1/3/5 precision changes of appending the mined single/double Bensetsu lexicons to a baseline IME system.

isting twitter lexicon[9] contains 155 entries. Table 4 shows the coverage rates. Even we removed nearly half of the entries using the filtering strategies, the coverage rates do not drop that much (nearly 5% to 8%). The highest coverage rate belongs to the single/double Bensetsu lexicon with a filtering threshold of 20.

Finally, we append the mined single/double Bensetsu lexicons (cut.20) to the Baidu Japanese IME system (Chen et al., 2012) by taking the 155 entries as a test set. The top 1/3/5 precision changes are listed in Table 5. The precision of the top-1 candidate significantly improves from 38.93% to 48.99% (+10.06%). Through these numbers, we can say that the proposed approach is helpful for improving real NLP applications, such as the Japanese IME system.

4 Conclusion

We have proposed an algorithm for mining new compound words from single/double Bensetsus. Experiments show that the algorithm can efficiently collect novel compound words from tweets and large-scale monolingual Japanese sentences. One natural extension is to mine compound words from more than two, or non-contiguous Bensetsus, such as もしかしたら... かもしれない.

References

Long Chen, Xianchao Wu, and Jingzhou He. 2012. Using collocations and k-means clustering to improve the n-pos model for japanese ime. In *Proceedings of Second Workshop on Advances in Text Input Methods (WTIM 2) (COLING 2012 Post-Conference Workshops)*.

Taku Kudo and Yuji Matsumoto. 2002. 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.

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