Accurate Word Segmentation using Transliteration and Language Model Projection

Masato Hagiwara  Satoshi Sekine
Rakuten Institute of Technology, New York
215 Park Avenue South, New York, NY
{masato.hagiwara, satoshi.b.sekine}@mail.rakuten.com

Abstract

Transliterated compound nouns not separated by whitespaces pose difficulty on word segmentation (WS). Offline approaches have been proposed to split them using word statistics, but they rely on static lexicon, limiting their use. We propose an online approach, integrating source LM, and/or, back-transliteration and English LM. The experiments on Japanese and Chinese WS have shown that the proposed models achieve significant improvement over state-of-the-art, reducing 16% errors in Japanese.

1 Introduction

Accurate word segmentation (WS) is the key component in successful language processing. The problem is pronounced in languages such as Japanese and Chinese, where words are not separated by whitespaces. In particular, compound nouns pose difficulties to WS since they are productive, and often consist of unknown words.

In Japanese, transliterated foreign compound words written in Katakana are extremely difficult to split up into components without proper lexical knowledge. For example, when splitting a compound noun ブラキッシュレッド burakisshureddo, a traditional word segmenter can easily segment this as ブラキッシュ/shureddo “blackish red” since shureddo “shred” is a known, frequent word. It is only the knowledge that ブラキッシュ buraki (“blacki”) is not a valid word which prevents this. Knowing that the back-transliterated unigram “blacki” and bigram “blacki shred” are unlikely in English can promote the correct WS, ブラキッシュ/レッド “blackish red”. In Chinese, the problem can be more severe since the language does not have a separate script to represent transliterated words.

Kaji and Kitsuregawa (2011) tackled Katakana compound splitting using back-transliteration and paraphrasing. Their approach falls into an offline approach, which focuses on creating dictionaries by extracting new words from large corpora separately before WS. However, offline approaches have limitation unless the lexicon is constantly updated. Moreover, they only deal with Katakana, but their method is not directly applicable to Chinese since the language lacks a separate script for transliterated words.

Instead, we adopt an online approach, which deals with unknown words simultaneously as the model analyzes the input. Our approach is based on semi-Markov discriminative structure prediction, and it incorporates English back-transliteration and English language models (LMs) into WS in a seamless way. We refer to this process of transliterating unknown words into another language and using the target LM as LM projection. Since the model employs a general transliteration model and a general English LM, it achieves robust WS for unknown words. To the best of our knowledge, this paper is the first to use transliteration and projected LMs in an online, seamlessly integrated fashion for WS.

To show the effectiveness of our approach, we test our models on a Japanese balanced corpus and an electronic commerce domain corpus, and a balanced Chinese corpus. The results show that we achieved a significant improvement in WS accuracy in both languages.

2 Related Work

In Japanese WS, unknown words are usually dealt with in an online manner with the unknown word model, which uses heuristics
depending on character types (Kudo et al., 2004). Nagata (1999) proposed a Japanese unknown word model which considers PoS (part of speech), word length model and orthography. Uchimoto et al. (2001) proposed a maximum entropy morphological analyzer robust to unknown words. In Chinese, Peng et al. (2004) used CRF confidence to detect new words.


Kaji and Kitsuregawa (2011)'s approach is the closest to ours. They built a model to split Katakana compounds using back-transliteration and paraphrasing mined from large corpora. Nakazawa et al. (2005) is a similar approach, using a Ja-En dictionary to translate compound components and check their occurrence in an English corpus. Similar approaches are proposed for other languages, such as German (Koehn and Knight, 2003) and Urdu-Hindi (Lehal, 2010). Correct splitting of compound nouns has a positive effect on MT (Koehn and Knight, 2003) and IR (Braschler and Ripplinger, 2004).

A similar problem can be seen in Korean, German etc. where compounds may not be explicitly split by whitespaces. Koehn and Knight (2003) tackled the splitting problem in German, by using word statistics in a monolingual corpus. They also used the information whether translations of compound parts appear in a German-English bilingual corpus. Lehal (2010) used Urdu-Devnagri transliteration and a Hindi corpus for handling the space omission problem in Urdu compound words.

3 Word Segmentation Model

Our baseline model is a semi-Markov structure prediction model which estimates WS and the PoS sequence simultaneously (Kudo et al., 2004; Zhang and Clark, 2008). This model finds the best output $y^*$ from the input sentence string $x$ as: $y^* = \arg\max_{y \in Y(x)} w \cdot \phi(y)$. Here, $Y(x)$ denotes all the possible sequences of words derived from $x$. The best analysis is determined by the feature function $\phi(y)$ the weight vector $w$. WS is conducted by standard Viterbi search based on lattice, which is illustrated in Figure 1. We limit the features to word unigram and bigram features, i.e., $\phi(y) = \sum_i [\phi_1(w_i) + \phi_2(w_{i-1}, w_i)]$ for $y = w_1...w_n$. By factoring the feature function into these two subsets, argmax can be efficiently searched by the Viterbi algorithm, with its computational complexity proportional to the input length. We list all the baseline features in Table 1. The asterisks (*) indicate the feature is used for Japanese (JA) but not for Chinese (ZH) WS. Here, $w_i$ and $w_{i-1}$ denote the current and previous word in question, and $t^j_i$ and $t^j_i$ are level-$j$ PoS tags assigned to them. $l(w)$ and $c(w)$ are the length and the set of character types of word $w$.

If there is a substring for which no dictionary entries are found, the unknown word model is invoked. In Japanese, our unknown word model relies on heuristics based on character types and word length to generate word nodes, similar to that of MeCab (Kudo et al., 2004). In Chinese, we aggregated consecutive 1 to 4 characters add them as “n (common noun)”, “ns (place name)”, “nr (personal name)”, and “nz (other proper nouns),” since most of the unknown words in Chinese are proper nouns. Also, we aggregated up to 20 consecutive numerical characters, making them a single node, and assign “m” (number). For other character types, a single node with PoS “w (others)” is created.

Table 1: Features for WS & PoS tagging

<table>
<thead>
<tr>
<th>ID</th>
<th>Feature</th>
<th>ID</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$w_i$</td>
<td>13</td>
<td>$w_{i-1}w_i$</td>
</tr>
<tr>
<td>2</td>
<td>$t^1_i$</td>
<td>14</td>
<td>$t^1_{i-1}t^1_i$</td>
</tr>
<tr>
<td>3*</td>
<td>$t^2_i$</td>
<td>15*</td>
<td>$t^2_{i-1}t^2_i$</td>
</tr>
<tr>
<td>4*</td>
<td>$t^3_i$</td>
<td>16*</td>
<td>$t^3_{i-1}t^3_i$</td>
</tr>
<tr>
<td>5*</td>
<td>$t^4_i$</td>
<td>17*</td>
<td>$t^4_{i-1}t^4_i$</td>
</tr>
<tr>
<td>6*</td>
<td>$t^5_i$</td>
<td>18*</td>
<td>$t^5_{i-1}t^5_i$</td>
</tr>
<tr>
<td>7</td>
<td>$w_i$</td>
<td>19</td>
<td>$\phi_{LMPS}(w_i)$</td>
</tr>
<tr>
<td>8*</td>
<td>$w_{i-1}t^2_i$</td>
<td>20</td>
<td>$\phi_{LMPS}(w_{i-1}, w_i)$</td>
</tr>
<tr>
<td>9*</td>
<td>$w_{i-1}t^3_i$</td>
<td>21</td>
<td>$\phi_{LMPS}(w_{i-1})$</td>
</tr>
<tr>
<td>10*</td>
<td>$w_{i-1}t^4_i$</td>
<td>22</td>
<td>$\phi_{LMPS}(w_{i-1}, w_i)$</td>
</tr>
<tr>
<td>11*</td>
<td>$w_{i-1}t^5_i$</td>
<td>23</td>
<td>$\phi_{LMPS}(w_{i-1})$</td>
</tr>
<tr>
<td>12</td>
<td>$c(w_i)(l(w_i))$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1The Japanese dictionary and the corpus we used have 6 levels of PoS tag hierarchy, while the Chinese ones have only one level, which is why some of the PoS features are not included in Chinese. As character type, Hiragana (JA), Katakana (JA), Latin alphabet, Number, Chinese character, and Others, are distinguished. Word length is in Unicode.
4 Use of Language Model

Language Model Augmentation Analagous to Koehn and Knight (2003), we can exploit the fact that レッド reddo (red) in the example ブラキッシュレッド is such a common word that one can expect it appears frequently in the training corpus. To incorporate this intuition, we used log probability of n-gram as features, which are included in Table 1 (ID 19 and 20): $\phi_1^{LMS}(w_i) = \log p(w_i)$ and $\phi_2^{LMS}(w_{i-1}, w_i) = \log p(w_{i-1}, w_i)$. Here the empirical probability $p(w_i)$ and $p(w_{i-1}, w_i)$ are computed from the source language corpus. In Japanese, we applied this source language augmentation only to Katakana words. In Chinese, we did not limit the target.

4.1 Language Model Projection

As we mentioned in Section 2, English LM knowledge helps split transliterated compounds. We use (LM) projection, which is a combination of back-transliteration and an English model, by extending the normal lattice building process as follows:

Firstly, when the lattice is being built, each node is back-transliterated and the resulting nodes are associated with it, as shown in Figure 1 as the shaded nodes. Then, edges are spanned between these extended English nodes, instead of between the original nodes, by additionally taking into consideration English LM features (ID 21 and 22 in Table 1): $\phi_1^{LMP}(w_i) = \log p(w_i)$ and $\phi_2^{LMP}(w_{i-1}, w_i) = \log p(w_{i-1}, w_i)$. Here the empirical probability $p(w_i)$ and $p(w_{i-1}, w_i)$ are computed from the English corpus. For example, Feature 21 is set to $\phi_1^{LMP}$ ("blackish") for node (a), to $\phi_1^{LMP}$ ("red") for node (b), and Feature 22 is set to $\phi_2^{LMP}$ ("blackish", "red") for edge (c) in Figure 1. If no transliterations were generated, or the n-grams do not appear in the English corpus, a small frequency $\epsilon$ is assumed.

Finally, the created edges are traversed from EOS, and associated original nodes are chosen as the WS result. In Figure 1, the bold edges are traversed at the final step, and the corresponding nodes “大 - 人气 - 色 - ブラキッシュ - レッド” are chosen as the final WS result.

For Japanese, we only expand and project Katakana noun nodes (whether they are known or unknown words) since transliterated words are almost always written in Katakana. For Chinese, only “ns (place name)”, “nr (personal name)”, and “nz (other proper noun)” nodes whose surface form is more than 1-character long are transliterated. As the English LM, we used Google Web 1T 5-gram Version 1 (Brants and Franz, 2006), limiting it to unigrams occurring more than 2000 times and bigrams occurring more than 500 times.

5 Transliteration

For transliterating Japanese/Chinese words back to English, we adopted the Joint Source Channel (JSC) Model (Li et al., 2004), a generative model widely used as a simple yet powerful baseline in previous research e.g., (Hagiwara and Sekine, 2012; Finch and Sumita, 2010).\(^2\) The JSC model, given an input of source word $s$ and target word $t$, defines the transliteration probability based on transliteration units (TUs) $u_i = \langle s_i, t_i \rangle$ as: $P_{JSC}(s, t) = \prod_{i=1}^{f} P(u_i | u_{i-n+1}, ..., u_{i-1})$, where $f$ is the number of TUs in a given source / target word pair. TUs are atomic pair units of source / target words, such as “la/ザ” and “ish/ッシュ”。The TU n-gram probabilities are learned from a training corpus by following iterative updates similar to the EM algorithm\(^3\).

In order to generate transliteration candidates, we used a stack decoder described in (Hagiwara and Sekine, 2012). We used the training data of the NEWS 2009 workshop (Li et al., 2009a; Li et al., 2009b).

As reference, we measured the performance on its own, using NEWS 2009 (Li et al., 2009b) data. The percentage of correctly transliterated words are 37.9% for Japanese and 25.6%\(^3\).\(^4\) Note that one could also adopt other generative / discriminative transliteration models, such as (Jianpojarn et al., 2007; Jianpojarn et al., 2008).

\(^2\)We only allow TUs whose length is shorter than or equal to 3, both in the source and target side.
for Chinese. Although the numbers seem low at a first glance, Chinese back-transliteration itself is a very hard task, mostly because Chinese phonology is so different from English that some sounds may be dropped when transliterated. Therefore, we can regard this performance as a lower bound of the transliteration module performance we used for WS.

6 Experiments

6.1 Experimental Settings

Corpora For Japanese, we used (1) EC corpus, consists of 1,230 product titles and descriptions randomly sampled from Rakuten (Rakuten-Inc., 2012). The corpus is manually annotated with the BCCWJ style WS (Ogura et al., 2011). It consists of 118,355 tokens, and has a relatively high percentage of Katakana words (11.2%). (2) BCCWJ (Maekawa, 2008) CORE (60,374 sentences, 1,286,899 tokens, out of which approx. 3.58% are Katakana words). As the dictionary, we used UniDic (Den et al., 2007). For Chinese, we used LCMC (McEnery and Xiao, 2004) (45,697 sentences and 1,001,549 tokens). As the dictionary, we used CC-CEDICT (MDGB, 2011).

Training and Evaluation We used Averaged Perceptron (Collins, 2002) (3 iterations) for training, with five-fold cross-validation. As for the evaluation metrics, we used Precision (Prec.), Recall (Rec.), and F-measure (F). We additionally evaluated the performance limited to Katakana (JA) or proper nouns (ZH) in order to see the impact of compound splitting. We also used word error rate (WER) to see the relative change of errors.

6.2 Japanese WS Results

We compared the baseline model, the augmented model with the source language (+LM-S) and the projected model (+LM-P). Table 3 shows the result of the proposed models and major open-source Japanese WS systems, namely, MeCab 0.98 (Kudo et al., 2004), JUMAN 7.0 (Kurohashi and Nagao, 1994), and KyTea 0.4.2 (Neubig et al., 2011). We observed slight improvement by incorporating the source LM, and observed a 0.48 point F-value increase over baseline, which translates to 4.65 point Katakana F-value change and 16.0% (3.56% to 2.99 %) WER reduction, mainly due to its higher Katakana word rate (11.2%). Here, MeCab+UniDic achieved slightly better Katakana WS than the proposed models. This may be because it is trained on a much larger training corpus (the whole BCCWJ). The same trend is observed for BCCWJ corpus (Table 2), where we gained statistically significant 1 point F-measure increase on Katakana word.

Many of the improvements of +LM-S over Baseline come from finer grained splitting, for example, *レインスーツ reinsuutsu “rain suits” to レイン/スーツ, while there is wrong over-splitting, e.g., テレキャスター terekyasutaa “Telecaster” to テレ/キャスター. This type of error is reduced by +LM-P, e.g., *プラス/チップラスチック purasuchikku “*plus tick” toプラスチック purasuchikku “plastic” due to LM projection. +LM-P also improved compounds whose components do not appear in the training data, such as *ルーカスフィルム ruukasufirumu to ルーカス/フィルム “Lucas Film.” Indeed, we randomly extracted 30 Katakana differences between +LM-S and +LM-P, and found out that 25 out of 30 (83%) are true improvement. One of the proposed method’s advantages is that it is very robust to variations, such as アクティベイティッド akutibeitiddo “activated,” even though only the original form, アクティブート akutibeto “activate” is in the dictionary.

One type of errors can be attributed to non-English words such as スノコベッド sunokobeddo, which is a compound of Japanese word スノコ sunoko “duckboard” and an English word ベッド beddo “bed.”

6.3 Chinese WS Results

We compare the results on Chinese WS, with Stanford Segmenter (Tseng et al., 2005) (Table 4). Including +LM-S decreased the
performance, which may be because one cannot limit where the source LM features are applied. This is why the result of +LM-S+LM-P is not shown for Chinese. On the other hand, replacing LM-S with LM-P improved the performance significantly. We found positive changes such as *欧麦尔萨利赫 oumaier/salihe to 欧麦尔/萨利赫 oumaier/salihe “Umar Saleh” and *领导/人曼德拉 lingdaoren/renmandela to 领导人/曼德拉 lingdaoren/mandela “Leader Mandela”. However, considering the overall F-measure increase and proper noun F-measure decrease suggests that the effect of LM projection is not limited to proper nouns but also promoted finer granularity because we observed proper noun recall increase.

One of the reasons which make Chinese LM projection difficult is the corpus allows single tokens with a transliterated part and Chinese affices, e.g., 马克思主义者 makesizhuyizhe “Marxists” (马克思 makesi “Marx” + 主义者 zhuyizhe “-ist (believers)”) and 尼罗河 niluohe “Nile River” ( 尼罗 niluo “Nile” + 河 he “-river”). Another source of errors is transliteration accuracy. For example, no appropriate transliterations were generated for 维纳斯 weinasi “Venus,” which is commonly spelled 维纳斯 weinasi. Improving the JSC model could improve the LM projection performance.

7 Conclusion and Future Works

In this paper, we proposed a novel, online WS model for the Japanese/Chinese compound word splitting problem, by seamlessly incorporating the knowledge that back-transliteration of properly segmented words also appear in an English LM. The experimental results show that the model achieves a significant improvement over the baseline and LM augmentation, achieving 16% WER reduction in the EC domain.

The concept of LM projection is general enough to be used for splitting other compound nouns. For example, for Japanese personal names such as 仲里依紗 Nakariisa, if we could successfully estimate the pronunciation Nakariisa and look up possible splits in an English LM, one is expected to find a correct WS Naka Ríisa because the first and/or the last name are mentioned in the LM. Seeking broader application of LM projection is a future work.
References


