Combination of Machine Learning Methods for Optimum Chinese Word Segmentation

Masayuki Asahara **Kenta Fukuoka** **Ai Azuma** **Takashi Tsuzuki**
Chooi-Ling Goh **Yotaro Watanabe** **Yuji Matsumoto** Matsushita Electric
Nara Institute of Science and Technology, Japan Industrial Co., Ltd.
E-mail: cje@is.naist.jp

Abstract

This article presents our recent work for participation in the Second International Chinese Word Segmentation Bakeoff. Our system performs two procedures: Out-of-vocabulary extraction and word segmentation. We compose three out-of-vocabulary extraction modules: Character-based tagging with different classifiers – maximum entropy, support vector machines, and conditional random fields. We also compose three word segmentation modules – character-based tagging by maximum entropy classifier, maximum entropy markov model, and conditional random fields. All modules are based on previously proposed methods. We submitted three systems which are different combination of the modules.

1 Overview

We compose three systems: *Models a*, *b* and *c* for the closed test tracks on all four data sets.

For *Models a* and *c*, three out-of-vocabulary (OOV) word extraction modules are composed: 1. Maximum Entropy (MaxEnt) classifier-based tagging; 2. Maximum Entropy Markov Model (MEMM)-based word segmenter with Conditional Random Fields (CRF)-based chunking; 3. MEMM-based word segmenter with Support Vector Machines (SVM)-based chunking. Two lists of OOV word candidates are constructed either by voting or merging the three OOV word extraction modules. Finally, a CRFs-based word segmenter produces the final results using either of the voted list (*Model a*) or the merged list (*Model c*).

Most of the classifiers use surrounding words and characters as the contextual features. Since word and character features may cause data sparse problem, we utilize a hard clustering algorithm (K-means) to define word classes and character classes in order to overcome the data sparse problem. The word classes are used as the hidden states in MEMM and CRF-based word segmenters. The character classes are used as the features in character-based tagging, character-based chunking and word segmentation.

Model b is our previous method proposed in (Goh et al., 2004b): First, a MaxEnt classifier is used to perform character-based tagging to identify OOV words in the test data. In-vocabulary (IV) word list together with the extracted OOV word candidates is used in Maximum Matching algorithm. Overlapping ambiguity is denoted by the different outputs from Forward and Backward Maximum Matching algorithm. Finally, character-based tagging by MaxEnt classifier resolves the ambiguity.

Section 2 describes *Models a* and *c*. Section 3 describes *Model b*. Section 4 discusses the differences among the three models.

2 Models a and c

Models a and *c* use several modules. First, a hard clustering algorithm is used to define word classes and character classes. Second, three OOV extraction modules are trained with the training data. These modules, then, extract the OOV words in the test data. Third, the OOV word candidates produced by the three OOV extraction modules are refined by voting (*Model a*) or merging (*Model c*) them. The final word list is composed by appending the OOV word candidates to the IV word list. Finally, a CRF-based word segmenter analyzes the sentence based on the new word list.

2.1 Clustering for word/character classes

We perform hard clustering for all words and characters in the training data. K-means algorithm is utilized. We use R 2.2.1 (http://www.r-project.org/) to perform k-means clustering.

Since the word types are too large, we cannot run k-means clustering on the whole data. Therefore, we divide the word types into 4 groups randomly. K-means clustering is performed for each group. Words in each group are divided into 5 disjoint classes, producing 20 classes in total. Preceding and succeeding words in the top 2000 rank are used as the features for the clustering. We define the set of the OOV words as the 21st class. We also define two other classes for the begin-of-sentence (BOS) and end-of-sentence (EOS). So, we define 23 classes in total.

20 classes are defined for characters. K-means clustering is performed for all characters in the training data. Preceding and succeeding characters and BIES position tags are used as features for the clustering: "B" stands for 'the first character of a word'; "I" stands for 'an intermediate character of a word'; "E" stands for 'the last character of a word'; "S" stands for 'the single character word'. Characters only in the test data are not assigned with any character class.

2.2 Three OOV extraction modules

In *Models a* and *c*, we use three OOV extraction modules.

First and second OOV extraction modules use the output of a Maximam Entropy Markov Model (MEMM)-based word segmenter (McCallum et al., 2000) (Uchimoto et al., 2001). Word list is composed by the words appeared in 80% of the training data. The words occured only in the remaining 20% of the training data are regarded as OOV words. All word candidates in a sentence are extracted to form a trellis. Each word is assigned with a word class. The word classes are used as the hidden states in the trellis. In encoding, MaxEnt estimates state transition probabilities based on the preceding word class (state) and observed features such as the first character, last character, first character class, last character class of the current word. In decoding, a simple Viterbi algorithm is used.

The output of the MEMM-based word segmenter is splitted character by character. Next, character-based chunking is performed to extract OOV words. We use two chunkers: based on SVM (Kudo and Matsumoto, 2001) and CRF (Lafferty et al., 2001). The chunker annotates BIO position tags: "B" stands for 'the first character of an OOV word'; "I" stands for 'other characters in an OOV word'; "O" stands for 'a character outside an OOV word'.

The features used in the two chunkers are the characters, the character classes and the information of other characters in five-character window size. The word sequence output by the MEMM-based word segmenter is converted into character sequence with BIES position tags and the word classes. The position tags with the word classes are also introduced as the features.

The third one is a variation of the OOV module in section 3 which is character-based tagging by MaxEnt classifier. The difference is that we newly introduce character classes in section 2.1 as the features.

In summary, we introduce three OOV word extraction modules: "MEMM+SVM", "MEMM+CRF" and "MaxEnt classifier".

2.3 Voting/Merging the OOV words

The word list for the final word segmenter are composed by voting or merging. Voting means the OOV words which are extracted by two or more OOV word extraction modules. Merging means the OOV words which are extracted by any of the OOV word extraction modules. The model with the former (voting) OOV word list is used in *Model a*, and the model with the latter (merging) OOV word list is used in *Model c*.

2.4 CRF-based word segmenter

Final word segmentation is carried out by a CRF-based word segmenter (Kudo and Matsumoto, 2004) (Peng and McCallum, 2004). The word trellis is composed by the similar method with MEMM-based word segmenter. Though state transition probabilities are estimated in the case of MaxEnt framework, the probabilities are normalized in the whole sentence in CRF-based method. CRF-based word segmenter is robust to length-bias problem (Kudo and Matsumoto, 2004) by the global normalization. We will discuss the length-bias problem in section 4.

2.5 Note on MSR data

Unfortunately, we could not complete *Models a* and *c* for the MSR data due to time constraints. Therefore, we submitted the following 2 fragmented models: *Model a* for MSR data is MEMM-based word segmenter with OOV word list by voting; *Model c* for MSR data is CRF-based word segmenter with no OOV word candidate.

3 Model b

Model b uses a different approach. First, we extract the OOV words using a MaxEnt classifier with only the character as the features. We did not use the character classes as the features. Each character is assigned with BIES position tags. Word segmentation by character-based tagging is firstly introduced by (Xue and Converse, 2002). In encoding, we extract characters within five-character window size for each character position in the training data as the features for the classifier. In decoding, the BIES position tag is deterministically annotated character by character in the test data. The

words that appear only in the test data are treated as OOV word candidates.

We can obtain quite high unknown word recall with this model but the precision is a bit low. However, the following segmentation model will try to eliminate some false unknown words. In the next step, we append OOV word candidates into the IV word list extracted from the training data. The segmentation model is similar to the OOV extraction method, except that the features include the output from the Maximum Matching (MaxMatch) algorithm. The algorithm runs in both forward (FMaxMatch) and backward (BMaxMatch) directions using the final word list as the references. The outputs of FMaxMatch and BMaxMatch are also assigned with BIES tags. The differences between the FMaxMatch and BMaxMatch outputs indicate the positions where the overlapping ambiguities occur. The final word segmentation is carried out by MaxEnt classifier again.

Note, both procedures in *Model b* use whole training data in the training phase. The dictionary used in the MaxMatch algorithm is extracted from the training data only during the training phase. So, the training of segmentation model does not explicitly consider OOV words. We did not use the word and character classes as features in *Model b* unlike in the case of *Models a* and *c*. The details of the model can be found in (Goh et al., 2004b). The difference is that we do not provide character types here because it is forbidden in this round. Besides, we also did not prune the OOV words because this step involve the intervention of human knowledge.

4 Discussions and Conclusions

Table 1 summarizes the results of the three models. The proposed systems employ purely corpus-based statistical/machine learning method. Now, we discuss what we observe in the three models. We remark two problems in word segmentation: OOV word problem and length-bias problem.

OOV word problem is that simple word-based Markov Model family cannot analyze the words not included in the word list. One of the solutions is character-based tagging (Xue and Converse, 2002) (Goh et al., 2004a). The simple character-based tagging (*Model b*) achieved high R_{OOV} but the precision is low. We tried to refine OOV extraction by voting and merging (*Model a* and *c*). However, the R_{OOV} of *Models a* and *c* are not as good as that of *Model b*. Figure 1 shows type-precision and type-recall of each OOV extraction modules. While voting helps to make the precision higher, voting deteriorates the recall. Defining some hand written rules to prune false OOV words will help to improve the IV word segmentation (Goh et al., 2004b), because the precision of OOV word extraction becomes higher. Other types of OOV word extraction methods should be introduced. For example, (Uchimoto et al., 2001) embeded OOV models in MEMM-based word segmenter (with POS tagging). Less than six-character substrings are extracted as the OOV word candidates in the word trellis. (Peng and McCallum, 2004) proposed OOV word extraction methods based on CRF-based word segmenter. Their CRF-based word segmenter can compute a confidence in each segment. The high confident segments that are not in the IV word list are regarded as OOV word candidates. (Nakagawa, 2004) proposed integration of word and OOV word position tag in a trellis. These three OOV extraction method are different from our methods – character-based tagging. Future work will include implementation of these different sorts of OOV word extraction modules.

Length bias problem means the tendency that the locally normalized Markov Model family prefers longer words. Since choosing the longer words reduces the number of words in a sentence, the state-transitions are reduced. The less the state-transitions, the larger the likelihood of the whole sentence. Actually, the length-bias reflects the real distribution in the corpus. Still, the length-bias problem is nonnegligible to achieve high accuracy due to small exceptional cases. We used CRF-based word segmenter which relaxes the problem (Kudo and Matsumoto, 2004). Actually, the CRF-based word segmenter achieved high R_{IV}.

We could not complete *Model a* and *c* for MSR. After the deadline, we managed to complete *Model a* (CRF + Voted Unk.) and *c* (CRF + Merged Unk.) The result of *Model a* was precesion 0.976, recall 0.966, F-measure 0.971, OOV recall 0.570 and IV recall 0.988. The result of *Model c* was precesion 0.969, recall 0.963, F-measure 0.966, OOV recall 0.571 and IV recall 0.974. While the results are quite good, unfortunately, we could not submit the outputs in time.

While our results for the three data sets (AS, CITYU, MSR) are fairly good, the result for the PKU data is not as good. There is no correlation between scores and OOV word rates. We investigate unseen character distributions in the data set. There is no correlation between scores and unseen character distributions.

We expected *Model c* (merging) to achieve higher recall for OOV words than *Model a* (voting). However, the result was opposite. The noises in OOV word candidates should have deteriorated the F-value of overall word segmentation. One reason might be that our CRF-based segmenter could not encode the occurence of OOV words. We defined the 21st word class for OOV words. However, the training data for CRF-based segmenter did not contain the 21st class. We should include the 21st class in the training data

Table 1: Our Three Models and Results: F-value/R_{OOV}/R_{IV} (Rank of F-value)

	AS	CITYU	MSR	PKU
Model a	CRF + Voted Unk.	CRF + Voted Unk.	MEMM + Voted Unk.	CRF + Voted Unk.
	0.947/0.606/0.971	0.942/0.629/0.967	0.949/0.378/0.971	0.934/0.521/0.955
	(2/11)	(2/15)	(16/29)	(10/23)
Model b	Char.-based tagging	Char.-based tagging	Char.-based tagging	Char.-based tagging
	0.952/0.696/0.963	0.941/0.736/0.953	0.958/0.718/0.958	0.941/0.760/0.941
	(1/11)	(3/15)	(6/29)	(7/23)
Model c	CRF + Merged Unk.	CRF + Merged Unk.	CRF + No Unk.	CRF + Merged Unk.
	0.939/0.445/0.967	0.928/0.598/0.940	0.943/0.025/0.990	0.917/0.325/0.940
	(7/11)	(8/15)	(21/29)	(14/23)

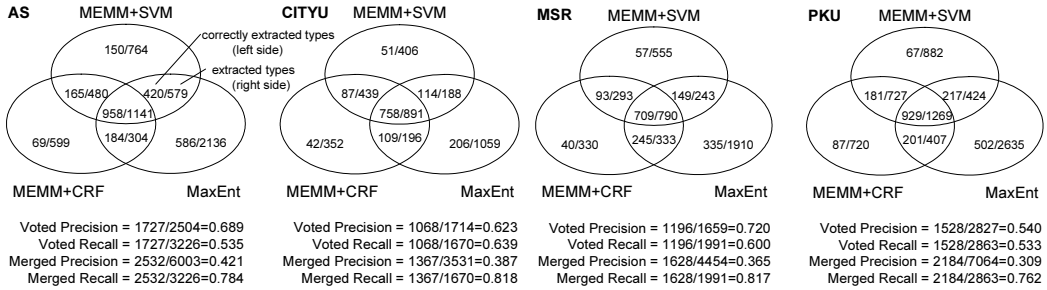

Figure 1: OOV Extraction Precision and Recall by Type

by regarding some words as pseudo OOV words.

We also found a bug in the CRF-based OOV word extration module. The accuracy of the module might be slightly better than the reported results. However, the effect of the bug on overall F-value might be limited, since the module was only part of the OOV extraction module combination – voting and merging.

Acknowledgement

We would like to express our appreciation to Dr. Taku Kudo who developed SVM-based chunker and gave us several fruitful comments.

References

Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2004a. Chinese Word Segmentation by Classification of Characters. In *Proc. of Third SIGHAN Workshop*, pages 57–64.

Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2004b. Pruning False Unknown Words to Improve Chinese Word Segmentation. In *Proc. of PACLIC-18*, pages 139–149.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with Support Vector Machines. In *Proc. of NAACL-2001*, pages 192–199.

Taku Kudo and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proc. of EMNLP-2004*, pages 230–237.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML-2001*, pages 282–289.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. of ICML-2000*, pages 591–598.

Tetsuji Nakagawa. 2004. Chinese and Japanese Word Segmentation Using Word-Level and Character-Level Information. In *Proc. of COLING-2004*, pages 466–472.

Fuchun Peng and Andrew McCallum. 2004. Chinese Segmentation and New Word Detection using Conditional Random Fields. In *Proc. of COLING-2004*, pages 562–568.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The Unknown Word Problem: a Morphological Analysis of Japanese Using Maximum Entropy Aided by a Dictionary. In *Proc. of EMNLP-2001*, pages 91–99.

Nianwen Xue and Susan P. Converse. 2002. Combining Classifiers for Chinese Word Segmentation. In *Proc. of First SIGHAN Workshop*, pages 63–70.