Semi-automatic Annotation of Chinese Word Structure

Jianqiang Ma† **Chunyu Kit‡** **Dale Gerdemann†**

† Department of Linguistics
University of Tübingen
Tübingen, Germany

‡Department of Chinese, Translation and Linguistics
City University of HK, HKSAR, China

{jma,dg}@sfs.uni-tuebingen.de ctckit@cityu.edu.hk

Abstract

Chinese word structure annotation is potentially useful for many NLP tasks, especially for Chinese word segmentation. Li and Zhou (2012) have presented an annotation for word structures in the Penn Chinese Treebank. But they only consider words that have productive affixes, which covers 35% of word types in that corpus. In this paper, we propose a linguistically inspired annotation that covers various morphological derivations of Chinese in a more general way, such that almost all multiple-character words can be structurally analyzed. As manual annotation is expensive, we propose a semi-supervised approach to automatic annotation, which combines the maximum entropy learning and the EM iteration for the Gaussian mixture model. The proposed method has achieved an accuracy of 90% on the testing set.

1 Introduction

In contrast to the pervasive success in creation and use of various language resources for corpus linguistics and natural language processing (NLP), *Chinese word structure annotation* has rarely been studied, although it is likely to be particularly useful to many NLP tasks, especially to Chinese word segmentation (CWS). In this paper, we propose a semi-supervised approach to automatic annotation of Chinese word structures.

Li (2011) shows many problems in CWS, including wordhood, granularity of lexical units for different applications, as well as several other linguistic phenomena, such as the so-called separable words, and points out that they can only be solved with adequate knowledge of word structure.

Our motivation for creating such an annotation is to test the usefulness of morphological information for the Out-Of-Vocabulary word (OOV) detection, a major challenge in CWS (Huang and Zhao, 2007). All state-of-the-art word segmenters (Zhao and Liu, 2010) based on classification (Berger et al., 1992; Xue, 2003) and sequence labeling (Lafferty et al, 2001; Peng et al., 2004) have to rely on using character n-grams as features. Despite recent advances in model combination (Wang et al., 2010; Sun, 2010), joint learning (Jiang et al., 2008; Zhang and Clark, 2008; Sun, 2011) and integration of supervised and unsupervised methods (Zhao and Kit, 2008; Sun and Xu, 2011), etc., an inherent problem with OOV words is that they are novel character combinations seldom occurring in a training corpus, giving machine learning methods little evidence for prediction. Like other linguistic elements, the distribution of character n-grams also obeys Zipf's (1949) law, indicating that *exponentially* more tokens have to occur before more distinct types are encountered. In other words, we need an exponential growth of annotated corpora to offset the *data sparseness problem* (Zhao et al., 2010), which is certainly expensive and impractical.

Morphology, on the other hand, offers a principled way to capture internal word structure and model the dynamic and productive *word formation process* for all words, including OOV ones. In this work, we will adhere to the conventional linguistic analysis of Chinese morphology (Packard, 2000; Xue, 2001). Chinese words are known to be poor in inflections and rich in derivations, including compounding, affixation and abbreviation, among many others. Li and Zhou (2012) introduce an affixation annotation on the Penn Chinese Treebank version 6.0 (CTB, Xue et al., 2005), which covers 35% word types.

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 9–17,
Tianjin, China, 20-21 DEC. 2012

The annotation to be addressed in this paper goes beyond affixation and explores for a general approach to accommodating more predominant processes including *compounding*. Our linguistically inspired annotation scheme (Section 3) is based on part-of-speech (POS) like tags for both characters and words, together with syntactic and morphological rules to derive these tags. In principle, our annotation covers most multiple-character words, except multi-char morphemes or binomes, such as 葡萄 'grape'.

Manual annotation is expensive and inefficient. To get around this problem, we propose a semi-supervised learning approach to automatic annotation of Chinese word structures, with a focus on two-character words. This method combines the maximum entropy learning and the EM iteration for Gaussian mixture models (Section 5). Our experiments show that it works significantly better than (1) two classic semi-supervised learning algorithms, self-training and co-training (Section 6), and (2) the supervised learning baseline (Section 4). The accuracy of the 1-best assignment of char tags by our approach is 90%. It is expected that the probabilistic nature of this approach can lead to an even lower error rate in real applications. To the best of our knowledge, this is the first attempt on wide-coverage semi-supervised automatic annotation of Chinese word structures.

2 Related Work

The morphology of Chinese has been studied in early works such as (Zhao, 1968; Lü, 1979) and more recently in the framework of generative linguistics, such as (Huang, 1984; Dai, 1992; Duanmu, 1997; Packard, 2000; Xue, 2001). Packard (2000) treats the morphology as an extension of syntax at the word (X0) level. Having a lexicalism flavor, it considers both morphemes and complex words with their "precompiled" morphological structures in the lexicon, except for complex words containing grammatical affixes.

In contrast, Xue (2001) has proposed a system that derives virtually *all* the complex words *with syntactic rules* or with the morphology module after syntactic analysis. The boundary of syntax and morphology further blurs and the operation scope of syntax rules expands most part of the morphology. Both Packard (2000) and Xue (2001) adopt form class descriptions, which assign words and their components (characters) POS-like tags called *form classes*. Also, rules in both systems are more or less syntactic.

Computational linguists have also started rethinking the limitations of feature-based machine learning approaches to CWS and have called for morphology-based analysis of OOV words (Dong et al., 2010). There are a few pivotal works in this direction, such as Zhao (2009), Li (2011) and Li & Zhou (2012). Zhao (2009) has proposed a character-based dependency parsing model, based on the annotation of unlabeled in-word character dependencies. While this is a valuable investigation, the deadlock of OOV word detection suggests that pure character-wise dependencies may be inadequate to model the morphological process.

Li (2011) and Li & Zhou (2012) have proposed models of joint morphological and syntactical analysis, for constituent and dependency parsing, respectively. Both are based on the same annotation of word structures for CTB. Influenced by Packard (2000), they only annotated words that contain productive affixes, which are only a *small subset* of words formed by morphological derivations. With a low coverage of the word formation phenomena, their models do not improve OOV word detection. The morphological model is expected to be effective in improving the performance of OOV word recognition, once syntax-like rules can be used to analyze most of, rather than a small portion of complex words, as illustrated in Xue (2001).

Our annotation differs from Li & Zhou's (2012) in that our annotation goes beyond affixation and aims at a thorough description of the derivational morphology in Chinese. Its ultimate goal is to construct a linguistic resource for training wide coverage word formation analyzers for Chinese.

3 Manual Annotation

3.1 Form-class description

Following Packard (2000) and Xue (2001), we adopt the *form class description* to describe the word formation analysis, as opposed to other possible descriptions of word structures, such as relational description, modification structure descriptions[1]. Character form classes refer to POS-like class identities for component morphemes of a word. For example, the word 吃饭 'to dine' can be analyzed as a verb [ ]$_V$ made of a verbal and a nominal element [V N]$_V$ 吃 'to eat' and 饭

[1] See Packard (2000) for a detailed discussion

'rice', where character form classes are denoted by the symbols inside the bracket while the word classes/POS tags are denoted by the subscript symbol of the bracket. Another example is the analysis of the adjective 先进 'advanced' as $[A\ V]_J$. In addition to form class identities, longer words have hierarchies in their elements as well.

The existence of monosyllabic words, with or without ambiguous POS tags, provides the initial link between character and word form classes (Packard, 2000). The form classes of bounded morphemes are more difficult to determine and requires extra clues such as morpho/lexical semantics.

3.2 Words to be annotated

Our annotation is carried out on CTB 5.0. Since longer words can be recursively analyzed similarly to single- and two-character words, we have chosen to focus on two-character words, which are shortest words that have inner structures. Note that the annotation of single-character words is trivial. Another reason for giving this priority to two-character words is mono- and bi-syllabic words together account for 64% and 92% word types and tokens in CTB 5.0, respectively. Our annotation has covered all 21151 open-class two-character words in CTB 5.0.

3.3 The annotation scheme

With form class description, annotating a two-character word equals to specifying its POS tags, form class co-occurrences of component characters and the association of the two. We have written programs to (1) extract the possible word and character form classes from CTB 5.0 and online resources[2], and (2) generate all the possible structures for a two-character word by calculating the Cartesian product of the sets of possible form classes of its left and right character, respectively.

The task of a human annotator is to choose the best structure for a <Word, POS> entry from computer generated candidates, if there are multiple ones. An annotator needs to figure out the optimal structure analysis, considering various information and constraints, including:

- *Semantic compatibility*. For example, word 发展 'to develop; development' $[V\ V]_V$ $[V\ V]_N$ can be interpreted as $[N\ V]_?$, if the nominal form of 发 'hair' is assumed. But this is incompatible with the overall word meaning, compared with the verbal form of 发 'open; send; get started'
- *Syntactic patterns*. Certain patterns such as N+N, V+V and J+J compounding are more likely than others, e.g. V+ C (verb + classifier) combination.
- *Word POS influence*. In many cases, the form class identity of a word may largely determine the form class identity of one or both of its constituents.

It is often necessary to refer to classic Chinese to properly use semantics clues. And note that most entries with the same word form but distinct POS tags can be captured by zero derivations and thus share the same structures as well. For example, word 发展 'to develop; development' has a base form with POS of verb $[V\ V]_V$, which zero-derives the noun form $[V\ V]_N$. As for the actual manual annotation, we have manually analyzed the 600 most frequent words in CTB 5.0. The whole annotation took about 30 annotator hours.

4 Supervised Annotation with ME

The number of manually annotated two-character words is less than 3% of the those in CTB 5.0. Given the limited resource, we have opted for training machine learning models from manual annotation to *automatically* annotate the rest 20551 two-character words. As described in Section 3.3, the annotation can be viewed as a tagging task that assigns each word entry a tag from a finite tag set of possible words structures, such as [V N] [V V]. In our annotation, the majority of the words turn out to be tagged as one of 14 most popular structures.

Tagging is a typical NLP problem that can be well solved by supervised classification. We have chosen the maximum entropy model (ME, Berger et al., 1992) to do the task, for its ability of accommodating overlapping features to achieve the state-of-the-art empirical performance.[3]

4.1 Features

For ME modeling, the choice of features strongly affects the result. As semantic features are more difficult to obtain and encode, we have mostly utilized *word POS tags* and *character syntactic patterns* as features, as shown in Table 1. In Table 1, $i(y)$ denotes the indicator function, which

[2] Mostly from http://www.zdic.net/

[3] We used Le Zhang's implementation in our experiments, available at: https://github.com/lzhang10/ME

Feature Type	Feature Group	Representative Feature
Word POS tag	Individual POS tags	$i(NN)$, $i(VV)$, $i(JJ)$, $i(AD)$, $i(VA)$, most_frequent_tag
	POS tag co-occurrence	$i(NN\ \&\ VV)$, $i(NN\ \&\ JJ)$, $i(JJ\ \&\ AD)$, $i(JJ\ \&\ VA)$
	Set of POS tags	set_of_all_possible_tags, $i(VV\ or\ NN\ or\ NR\ or\ NT)$
Left character form class	Individual form class	$i(N)$, $i(V)$, $i(J)$, $i(A)$, most_frequent_form_class
	Form class co-occurrence	$i(N\&\ V)$, $i(N\ \&\ J)$, $i(J\ \&\ A)$
	Set of form classes	set_of_all_possible_form_classes
Right character form class	*Similar to left character form class features*	
Possible structure	*Possible word structures both character classes of which are in the set of open-class*	

Table 1 Features of the ME based automatic annotator

represents whether the current feature matches pattern y. For example, $i(NN)$ in the first row says that "*NN* is a possible tag of the current word". We have systematically explored various feature configurations within these categories, among which the current feature set has achieved a better result.

4.2 Evaluation

We assume that (1) the word structures are independent and identically distributed variables and (2) automatic annotator's performance on samples of the complete set of two-character words, e.g. the manually annotated ones may reflect the performance on the complete set. We randomly split the manual annotation into a training set and a testing set, of 500 and 100 words, respectively. The performance of the model trained on the training set is measured by its *accuracy* on the testing set, which is calculated as follows:

$$Accuracy = \frac{number\ of\ correctly\ annotated\ words}{number\ of\ total\ words} \quad (1)$$

The average accuracy with 6-fold cross validation is *81%*. Note that the popular pair of metrics, *precision* and *recall* for binary classification does not apply for the evaluation of the collective result of multiple tags, as the original difference in denominators of the two metric formula no longer exists.

4.3 Discussion of ME results

In the incorrectly tagged cases, a few are impossible to learn, due to unseen classification tags. The majority are, however, related to *inherent ambiguities* of word structures, such as 完全 'complete(ly)' [J J] [A A], 实行 'to implement' [A V] [V V], and 影响 'to influence; impact' [N N] [V V]. Although one structure may be more plausible than the other for a word, the distinction is somehow inconclusive. This suggests that it is probably NOT the best to assign a single structure analysis for every case.

From a machine learning perspective, the model is characterized by *high variance* or overfitting, indicated by the big performance gap between the training (97%~92%) and testing (81%) accuracy. Besides the optimized regulation factors and the feature set, the only next thing that can improve the accuracy is probably to significantly increase the size of the annotated training set. In fact, the accuracy of 81% is a *reasonably good* result that can be obtained by ME with a relatively small set of available annotated examples.

5 Semi-supervised Annotation with Gaussian Mixture Model

5.1 Soft assignment of structures

Section 4.3 shows that many words are inherently ambiguous in structure. A better way of structure tagging may be soft assignment, i.e. allowing assignment of multiple structures to a word and using probabilities to indicate the likelihood of each assignment. For example, a soft assignment for 实行 'to implement' may look like:

[V V] : 0.8, [A V] : 0.15, [A N] : 0.01 ...

5.2 POS fingerprint features

POS features used in the ME model are discrete tag co-occurrence indicators. A drawback is that the distribution of POS tags is ignored. A better feature set is the distribution of the probabilities of seeing a certain POS tag T, given that the word is W, which can be estimated by normalized empirical counts with maximum likelihood estimation as follows:

$$P(T|W) = \frac{C(T,W)}{\sum_{T'} C(T',W)} \quad (2)$$

In practice, we only consider 10 open-class POS tags: *AD, CD, JJ, M, NN, NR, NT, OD, VA* and *VV*. The POS fingerprint, is a 10-dimensional vector that represents a word, each element of which is the conditional probability of the corresponding POS. With the model described in section 4, using original word POS features alone achieves an accuracy of 70%, while using POS fingerprint features alone achieves 74%.[4]

5.3 The generative model

Word POS tags strongly correlate with word structures (Packard, 2000). Human annotators use the single base POS tag to help annotate a word and utilize zero-derivation to generate ambiguous POS tags. But a computational model may need to keep POS ambiguities and use the distributions as features, as both base POS finding and zero-derivation probability estimation can be tricky. Even if a model can find the correct base POS for a word, the word structure may still be ambiguous in many cases, such as $[\text{V V}]_{\text{V}}$, $[\text{V N}]_{\text{V}}$ and $[\text{N V}]_{\text{V}}$. In short, it is an m-to-n non-deterministic mapping between an observable POS tag T and the latent structure S. A generative model that captures the joint distribution, $P(S,T)$ can generate all words represented by their POS fingerprints in repeated two steps:

1. Randomly choose a structure according to the structure distribution $P(S)$.
2. Draw a POS fingerprint data point according to the POS fingerprint distribution $P(T|S)$ given the chosen structure.

Each structure S determines a POS fingerprint distribution, which should somehow differ from the distributions of other structures, yet might considerable *overlaps* with that of others. This trait formalizes the observation that POS distribution has a significant correlation with structures, although words of different structures may show up with the same POS.

$P(T|S)$ should be a continuous distribution, as the data points, i.e. POS fingerprints, are continuous values. We choose the *Gaussian distribution*, following the central limit theorem stating that the average of a sufficiently large number of independent random variables can be approximated by the Gaussian. The prior distribution of structures $P(S)$ is a multinomial distribution, which neatly describes the random choice of discrete categories. An advantage of the generative model, as opposed to zero derivation, is that all possible POS tags of a word are treated in a similar way, which avoids the problems of base POS selection and derivation probability estimation.

5.4 Gaussian mixture model

The unsupervised version of the generative model can be formally described as a Gaussian mixture model (GMM, Bishop, 2006). The training data is a set of POS fingerprints $\{t^{(1)},..,t^{(m)}\}$ representing the word forms. The structures of these words, $\{s^{(1)},..,s^{(k)}\}$, are unknown, i.e. there is no structure annotation for any word. The data is specified by a joint distribution:

$$p(t^{(i)},s^{(i)}) = p(s^{(i)})p(t^{(i)}|s^{(i)}) \quad (3)$$

$$s^{(i)} \sim Multinominal\ (\emptyset)$$

$$t^{(i)} \mid (s^{(i)} = j\ ) \sim Gaussian\ (\mu_j, \Sigma_j)$$

where the parameter of the multinomial distribution $\emptyset_j = p(s^{(i)} = j) \geq 0, \sum_{j=1}^{k} \emptyset_j$. And μ and Σ are the vector of mean and variance of the Gaussian distribution, respectively.

The EM algorithm (Dempster et al., 1977) is the standard technique to estimate the parameters that maximize the likelihood of the data distribution with latent variables $s^{(i)}$. The algorithm runs the E-step and M-step iteratively until coverage:

1. E-step**:**

For each i and j, set:

$$w_j^{(i)} = p(s^{(i)} = j \,|t^{(i)};\ \emptyset, \mu, \Sigma\ ) =$$

$$= \frac{p(s^{(i)} = j;\ \emptyset\ )p(t^{(i)}|s^{(i)} = j; \mu, \Sigma)}{\sum_{l=1}^{k} p(s^{(i)} = l;\ \emptyset\ )p(t^{(i)}|s^{(i)} = l; \mu, \Sigma)} \quad (4)$$

2. M-step:

$$\emptyset_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)} \quad (5)$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} t^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}} \quad (6)$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (t^{(i)} - \mu_j)(t^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}} \quad (7)$$

The quantity that we calculate in the E-step, the posterior probability of the structure $s^{(i)}$, given $t^{(i)}$, the POS fingerprint that represents the word is exactly the soft assignment of structures

[4] Note that simple substituting original POS features with POS fingerprints leads to little performance improvement in our supervised annotation experiment.

that we need. The $P(S|T)$ values obtained in the last iteration make the final annotation.

5.5 Semi-supervised GMM

A problem with EM is that there is no guarantee of finding the global optima, i.e. it often suffers from local optima. So EM is usually sensitive to the initialization and the default random initialization often leads to poor results, which has also been observed in NLP tasks (Lamar et al., 2010; Peng and Schuurmans, 2001). To solve this problem, we propose a semi-supervised version of GMM that uses the probabilistic output of the ME model for the EM initialization.

We train the ME model for automatic annotation in the same way as in section 4 with 500 training words. Then we apply the model to predict the structures of the all the 21151 words in this study except the 100 testing words. Instead of using the single best prediction, here we utilize the *probabilistic output* of the ME model, which gives all possible structures of the words together with their marginal probabilities.

We use this output as $p(s^{(i)}|t^{(i)})$ to initialize the E-step of the EM algorithm. Since the EM algorithm runs on GMM, from now on, POS fingerprint features represent the words instead. The following points may explain why it can improve the performance: (1) Even though the best testing accuracy with "hard assignment" given by the ME model is only 81%, the "true" structure analyses may still exist as the top-k candidate with relatively large probabilities, while irrelevant ones may have only small probability mass. (2) In general the assignments that EM induce do not necessarily correspond to the desired classification tags, but the ME outputs can give the EM a better starting point to move towards the right one among all possible local optima, given the data likelihood and the classification accuracy are well correlated. (3) From the perspective of the original ME model, the connections and similarities between data points from a much bigger sample (21151 vs. 500) may help fix the high variance problem discussed in section 4.3.

The final soft assignments for the 100 testing words are obtained by applying the E-step for them with the parameter estimated in previous iterations. To get the hard assignment, we simply select the assignment with the highest probability for each word. The evaluation for the hard assignments is still based on testing accuracy, which stays at *90%* in multiple runs that we have tried.

6 Comparison Experiments

We have tried other approaches to automatic annotation to compare with the proposed method. Since our semi-supervised approach is a combination of supervised ME model and unsupervised GMM, two natural baselines would be the performance that could be achieved by applying two models independently, the former is **81%** as shown in section 4.

6.1 Unsupervised GMM

We have run the traditional unsupervised GMM, which is characterized by the random initialization of the EM algorithm. As there is no prior mapping between assignment IDs and word structures, their optimal one-to-one mapping is found via our implementation of the Hungarian algorithm (Kuhn, 1955). With 1-to-1 mapping, the testing accuracy is *54%* for several trial of random initialization.

6.2 Self-training

Self-training is a classic semi-supervised learning approach widely used in NLP. We have implemented and experimented with the Yarowsky (1995) version. It is a meta- algorithm based on a basic learning model, for which we use the ME model with the same features, training set and testing set as described in section 4. The unlabeled data U are the rest of the two-character words. We evaluate intermediate and final models with their performance on the testing set, the best of which is kept as the result.

Other setups: (1) *Loop stopping criterion*. We choose the performance on the testing data, conditioned on the current accuracy ≥ (the previous accuracy- tolerance). The tolerance avoids stopping too early. (2) *Selection criteria*. We use the standard one, namely, the classifier's confidence on its best prediction of each instance, which is highest marginal probability for ME. The selection relies on a parameter k, which defines the minimum confidence score needed for an instance to get selected. In our experiment, we have tried scores from 0.95 to 0.5 with an interval of 0.05.

We have tested with different configurations of k, splitting of U, and regularization parameters. The result of self-learning giving an *accuracy of 82%* is not too good- one percentage point beyond that of the baseline ME model.

6.3 Co-training

Co-training (Blum and Mitchell, 1998) is another classic semi-supervised algorithm. Two classifiers trained with independent views (feature set) are expected to teach one another in the iteration. Two views that we have adopted are: 1) left char and right char derived features and 2) POS fingerprint features.

With a standard setup of the co-training experiments, we have tried different selection criteria and regularization parameters. There is also only *slightly (1%) improvement* brought by co-training. It looks like that neither feature set of the two views provides the other with much additional information for classification, as the initial classifiers trained with these two views have already reached an accuracy of 68% and 74%, respectively.

To summarize, neither self-training nor co-training is capable of enhancing their performance to a level comparable to our proposed approach, which improves the accuracy from 81% to 90%. An overview of the performance of all tested methods in our research is given in Table 2.

Methods	Test Accuracy
ME	81%
Self-training	82%
Co-training	82%
Unsupervised GMM	54%
Semi-supervised GMM	*90%*

Table 2 Performance of the tested methods

7 Discussion

The performance of the proposed semi-supervised approach suggests that the distribution of the data has good characteristics that tightly link to the underlying structures. In other words, the form class descriptions of word structures provide much information for inducing the structural *regularities* of Chinese words.

To the best of our knowledge, this is the first work on automatic annotation of Chinese word structures based on semi-supervised learning. We are unable to find any existing work to directly compare with it. However, there are previous works on semi-supervised learning for other NLP tasks, such as document classification (Nigam et al., 2006). They used naïve Bayes for both the supervised learning and unsupervised learning, whereas our supervised and unsupervised models are ME and GMM, respectively. In our design, we use ME as our initial model, because it can incorporate overlapping features to get better baseline. We could not simply keep using ME as the model for EM iterations, because it does not take probabilistic (soft) assignment for training. We use Gaussian mixture for EM iteration out of two main reasons: (1) we observe a strong correlation between POS distribution and word structures, and (2) Gaussian can deal with continuous features and suffers not too much from the data sparseness, for it has only a few parameters to estimate.

A message from Nigam et al. (2006) is that in their experiments, the performance gap between the supervised model and the semi-supervised model that utilize extra unlabeled instances decreases from initially 20%~10% to complete diminishing when there are abundant labeled data to such a degree that unlabeled data do not provide any extra information. Despite the differences in modeling and application, we assume that these semi-supervised learning algorithms follow similar tack of performance improvement over the baselines.

In this sense, the performance improvement from 81% to 90% of our semi-supervised method is *very good*, especially in view of the high baseline and the relative error reduction (52%) it has achieved. Besides, we can directly use the probabilistic annotation to train models for real applications, which is probably a more sensible way than training on the hard-assignment (top-1) of structure analyses, due to the inherent ambiguities of word structures themselves. In this probabilistic/soft mode, the error rate for applications is expected to be further decreased, as the training of probabilistic grammar can be similar to EM: Even if the top-1 candidate is incorrect in a strict sense, the correct analysis may still exist in the top-k best with considerable amount of probability mass, in contrast with truly irrelevant ones. The accumulations of a large number of instances will push the probability distribution towards the right direction.

Of course, the ultimate purpose of this automatic annotation approach is to facilitate tasks such as grammar learning, Chinese word segmentation, and joint segmentation and parsing. As for the question of how good this accuracy of 90% can be to these applications, its answer has to be explored through further experiments. The success of existing works in this direction certainly points to a promising prospect.

8 Conclusion

We have developed a semi-supervised approach to annotating Chinese word structures, based on Chinese morphology and applied it to automatic annotation of two-character Chinese words with the aid of a Gaussian mixture model, which utilizes the output of the ME model for its initialization for EM iterations. The proposed method can achieve an accuracy of 90% on a test set of 100 words, using 500 manually annotated words as training examples. This method works significantly better than pure supervised model and two other typical semi-supervised learning techniques, namely self-training and co-training.

Since this work focuses only on structure annotation of two-character words in Chinese, our plan for future work will be to semi-automatically annotate longer words. This needs to incorporate annotation techniques in Li & Zhou (2012) and develop necessary models to describe the recursive nature of word derivation in Chinese. With a complete word structure annotation of all words in CTB, we expect to have more experiments with novel word structure-driven models for Chinese word segmentation and even a joint modeling of word segmentation and parsing, with a focus on the typical problems of OOV word recognition.

Acknowledgments

The research described in this paper has received funding from the European Commission's 7th Framework Program under grant agreement n° 238405 (project CLARA, a Marie Curie ITN), and is partially supported by Research Grants Council (RGC) of Hong Kong SAR, China through the GRF Grant 9041597 (CityU 144410).

References

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1): 39-71.

Christopher Bishop. 2006. *Pattern Recognition and Machine Learning*. New York: Springer.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings* of *COLT*, pp. 92-100. Madison, USA.

Xiang-Ling Dai. 1992. *Chinese Morphology and its Interface with the Syntax*. PhD Dissertation, Ohio State University.

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

Zhendong Dong, Qiang Dong and Changling Hao. 2010. Word segmentation needs change - from a linguist's view. In *Proceedings of CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 1-7. Beijing, China.

San Duanmu. 1997. "Wordhood in Chinese", in Jerome J. Packard ed. *New Approaches to Chinese Word Formation*, pp. 135-196. Mouton de Gruyter, New York, USA.

Changning Huang and Hai Zhao. 2007. Chinese word segmentation: A-decade Review. *Journal of Chinese Information Processing*, 21(3): 8-20

James C. T. Huang. 1984. Phrase structure, lexical integrity, and Chinese compounds. *Journal of the Chinese Language Teachers Association*, 19(2): 53-78.

Wenbin Jiang, Liang Huang, Qun Liu, Yajuan Lu. 2008. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL: HLT*, pp.897-904. Columbus, USA.

Harold Kuhn. 1955. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pp. 282-289. Williamstown, MA, USA

Michael Lamar, Yariv Maron, Mark Johnson, Elie Bienenstock. 2010. SVD and clustering for unsupervised POS tagging. In *Proceedings of* ACL (Short Papers), pp. 215-219. Uppsala, Sweden.

Zhongguo Li. 2011. Parsing the internal structure of words: A new paradigm for Chinese word segmentation. In *Proceedings of ACL: HLT*, pp. 1405-1414. Portland, Oregon, USA.

Zhongguo Li and Guodong Zhou. 2012. Unified dependency parsing of Chinese morphological and syntactic structures. In *Proceedings of* EMNLP-CoNLL, pp. 1445-1454. Jeji, Korea.

Shuxiang Lü. 1979. *Hanyu Yufa Fenxi Wenti "Problems in Syntactical Analysis of Chinese"*. Shangwu Yinshuguan, Beijing, China.

Kamal Nigam, Andrew McCallum and Tom Mitchell. 2006. Semi-supervised Text Classification Using EM. In Chapelle, O., Zien, A., and Scholkopf, B. (Eds.) *Semi-Supervised Learning*, 33-56. MIT Press: Boston.

Jerome Packard. 2000. *The Morphology of Chinese: A Linguistic and Cognitive Approach*. Cambridge University Press, Cambridge, UK.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING*, pp. 562-568. Geneva, Switzerland.

Fuchun Peng and Dale Schuurmans. 2001. Self-supervised Chinese word segmentation. In *Proceedings of the Fourth International Symposium on Intelligent Data Analysis,* pp. 238-247. *Cascais, Portugal*

Weiwei Sun. 2010. Word-based and character-based word segmentation models: Comparison and combination. In *Proceedings COLING*. pp. 1211-1219. Beijing, China.

Weiwei Sun. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. *In Proceedings ACL:HLT*, pp. 1385-1394. Portland, USA.

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings EMNLP*, pp. 970-979. Edinburgh, UK.

Kun Wang, Chengqing Zong and Keh-Yih Su. 2010. A character-based joint model for Chinese word segmentation. In *Proceedings of COLING*, pp. 1173-1181. Beijing, China.

Nianwen Xue. 2001. *Defining and Automatically Identifying words in Chinese*. Phd Thesis, University of Delaware.

Nianwen Xue. 2003. Chinese Word Segmentation as Characater Tagging. *Computational Linguistics and Chinese Language Processing*, 8(1): 29-48.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Tree bank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2) 207-238.

Davide Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*, pp. 189-196. Cambridge, USA.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL: HLT*, pp. 888-896. Columbus, USA.

Hai Zhao. 2009. Character-level dependencies in Chinese: usefulness and learning, pp. 879-887. In *Proceedings of EACL*, pp. 879-887. Athens, Greece.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of word segmentation and named entity recognition. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, pp. 106-111. Hyderabad, India.

Hai Zhao, Yan Song and Chunyu Kit. 2010. How large a corpus do we need: Statistical method vs. rule-based Method. In *Proceedings of LREC*, pp. 1672-1677. Malta.

Hongmei Zhao and Qun Liu. 2010. The CIPS-SIGHAN CLP 2010 Chinese Word Segmentation Bakeoff. In *Proceedings of the First CPS-SIGHAN Joint Conference on Chinese Language Processing*, pp. 199-209. Beijing, China.

Yuen-Ren Zhao. 1968. *Grammar of Spoken Chinese*. University of California Press.

George Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* Addison-Wisley. Oxford, UK.