Part-of-Speech Tagging for Chinese-English Mixed Texts with Dynamic Features

Jiayi Zhao[†] **Xipeng Qiu**[‡] **Shu Zhang**[§] **Feng Ji**[‡] **Xuanjing Huang**[‡]
School of Computer Science, Fudan University, Shanghai, China [†] [‡]
Fujitsu Research and Development Center, Beijing, China[§]
zjy.fudan@gmail.com[†]
{xpqiu,fengji,xjhuang}@fudan.edu.cn[‡]
zhangshu@cn.fujitsu.com [§]

Abstract

In modern Chinese articles or conversations, it is very popular to involve a few English words, especially in emails and Internet literature. Therefore, it becomes an important and challenging topic to analyze Chinese-English mixed texts. The underlying problem is how to tag part-of-speech (POS) for the English words involved. Due to the lack of specially annotated corpus, most of the English words are tagged as the oversimplified type, "foreign words". In this paper, we present a method using dynamic features to tag POS of mixed texts. Experiments show that our method achieves higher performance than traditional sequence labeling methods. Meanwhile, our method also boosts the performance of POS tagging for pure Chinese texts.

1 Introduction

Nowadays, Chinese-English mixed texts are prevalent in modern articles or emails. More and more English words are used in Chinese texts as names of organizations, products, terms and abbreviations, such as "eBay", "iPhone", "GDP", "Android" etc. On the other hand, it is also a common phenomenon to use Chinese-English mixed texts in daily conversation, especially in communication among employers in large international corporations.

There are some challenges for analyzing Chinese-English mixed texts:

1. How to define the POS tags for English words in these mixed texts. Since the standard of POS tags for English and Chinese are different, we cannot use English POS to tag the English words in mixed texts.
2. Due to lack of annotated corpus for mixed texts, most of the English words are tagged as "foreign words", which is oversimplified. So we cannot use them in further processing for the syntactic and semantic analysis.
3. Most English words used in mixed texts are often out-of-vocabulary (OOV), which thus increases the difficulties to tag them.

Currently, the mainstream method of Chinese POS tagging is joint segmentation & tagging with cross-labels, which can avoid the problem of error propagation and achieve higher performance on both subtasks(Ng and Low, 2004). Each label is the cross-product of a segmentation label and a tagging label, e.g. {B-NN, I-NN, E-NN, S-NN, ...}. The features are generated by position-based templates on character-level.

Since the main part of mixed texts is in Chinese and the role of English word is more like Chinese, we use Chinese POS tags (Xia, 2000) to tag English words. Since the categories of the most commonly used English words are nouns, verbs and adjectives, we can use "NN", "NR", "VV", "VA", "JJ" to label their POS tags.

For the English proper nouns and verbs, there are no significant differences in Chinese and English POS tags except that English features plural and tense forms.

For the English nouns, these are some English nouns used as verbs, such as "我很 [fan/VV] 他。(*I adore him very much.*)" where "fan" means "adore" and is used as a verb.

For the English adjectives, there are two corresponding Chinese POS tags "VA" and "JJ". For example, the roles of some English words in Table 1,

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1379–1388, Jeju Island, Korea, 12–14 July 2012.

Table 1: The POS tags of English Adjectives in Mixed Texts

Chinese	English
我 非 常 [professional/VA]。	I am very professional.
感觉很 [high/VA]。	Feel very high.
他是 [super/JJ] [star/NN]	He is a super star.

such as "professional" and "high", are different with their original ones.

Therefore, the POS tagging for mixed texts cannot be settled with simple methods, such as looking up in a dictionary.

One of the main differences between Chinese and English in POS tagging is that the two languages have character-based features and word-based features respectively. To ensure the consistency of tagging models, we prefer to use word-level information in Chinese, which is both useful for Chinese-English mixed texts and Chinese-only texts. For instance, in a sentence "X 或者 Y ... (X or Y ...)", the word Y ought to have the same POS tag as the word X. Another example is that the word following a pronoun is usually a verb, and adjectives often describe nouns. Some related works show that word-level features can improve the performance of Chinese POS tagging (Jiang et al., 2008; Sun, 2011).

In this paper, we propose a method to tag mixed texts with dynamic features. Our method combines these dynamic features, which are dynamically generated at the decoding stage, with traditional static features. For Chinese-English mixed texts, the traditional features cannot yield a satisfied result due to lack of training data. The proposed dynamic features can improve the performance by using the information of a word, such as POS tag or length of the whole word, which is proven effective by experiments.

The rest of the paper is organized as follows: In section 2, we introduce the sequence labeling models, then we describe our method of dynamic features in section 3 and analyze its complexity in section 4. Section 5 describes the training method. The experimental results are manifested in section 6. Finally, We review the relevant research works in section 7 and conclude our work in section 8.

2 Sequence Labeling Models

Sequence labeling is the task of assigning labels $\mathbf{y} = y_1, \ldots, y_n$ to an input sequence $\mathbf{x} = x_1, \ldots, x_n$. Given a sample $\mathbf{x}$, we define the feature $\Phi(\mathbf{x}, \mathbf{y})$. Thus, we can label $\mathbf{x}$ with a score function,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \quad (1)$$

where $\mathbf{w}$ is the parameter of function $F(\cdot)$.

For sequence labeling, the feature can be denoted as $\phi_k(y_i, y_{i-1}, \mathbf{x}, i)$, where i stands for the position in the sequence and k stands for the number of feature templates.

we use online Passive-Aggressive (PA) algorithm (Crammer and Singer, 2003; Crammer et al., 2006) to train the model parameters. Following (Collins, 2002), the average strategy is used to avoid the overfitting problem.

3 Dynamic Features

The form of traditional features is shown in Table 2, where C represents a Chinese character, and T represents the character-based tag. The subscript i indicates its position related to the current character.

Table 2: Traditional Feature Templates

$C_i, T_0 (i = -2, -1, 0, 1, 2)$
$C_i, C_j, T_0 (i, j = -2, -1, 0, 1, 2 \text{ and } i \neq j)$
T_{-1}, T_0

Traditional features are generated by position-fixed templates. Since the length of Chinese word is unfixed, their meanings are incomplete. We categorize them as "static" features since they can be calculated before tagging (except "T_{-1}, T_0").

The form of dynamic features is shown in Table 3, where $WORD$ represents a Chinese word, and POS (LEN) is the POS tag (length) of the word. The subscript of dynamic feature template indicates its position related to the current word.

Table 4 shows an example. If the current position is " Apple", then {POS_{-1}=CC, POS_{-2}=NR, $WORD_{-1}$="和", LEN_{-2}=2}. Since these features are unavailable before tagging, we call them "dynamic" features.

Table 3: Examples of Dynamic Feature Templates

$POS_i, POS_j, T_0(i, j = -2, -1, 0 \text{ and } i \neq j)$
$POS_i, WORD_j, T_0(i, j = -2, -1, 0)$
$WORD_i, LEN_j, POS_k, T_0(i, j, k = -2, -1, 0)$
...

Dynamic features are more flexible because the number of involved characters is dependent on the length of previous words. Unlike static features, dynamic features do not merely rely on the input sequence $C_{1:n}$, so the weights of dynamic features, in which POS/LEN are involved, can be trained by Chinese-only texts and used by mixed texts, which resolve the problem of the lack of training data.

4 Tagging with Dynamic Features

In the tagging stage, we use the current best result to approximately calculate the unknown tag information. For an input sequence $C_{1:n}$, the current best tags from index 0 to $i-1$ can be calculated by Viterbi algorithm and they can be used to generate dynamic features for index i. The specific algorithm is shown in Algorithm 1.

Here is an example to explain the time complexity of the dynamic features. Normal template $x_{i-2}x_{i-1}y_i$ requires to look for the positions of $i-2$ and $i-1$ related to the current character x_i, but dynamic template $pos_{i-2}pos_{i-1}y_i$ needs to know the pos tags of two words. If the length of $word_{i-1}/word_{i-2}$ is 2, then the positions of $i-4, i-3, i-2, i-1$ are needed to generate the dynamic features.

For all dynamic features, it is unnecessary to repetitively calculate the $POS/WORD/LEN$ array. Apart from that one time calculation of the array, no distinction can be found between the time complexity of the dynamic features and the traditional features. For input $C_{1:n}$, the time complexity is $O(n*[O(op.2)+(T_s.num+T_d.num)*O(op.1)+O(op.4)])$, n.b. $O(op.1) = O(op.3)$. Universally the dynamic features only require the information of position $i-2$ and $i-1$, so the time complexity of calculating the $POS/WORD/LEN$ array can be ignored as compared with the complexity of Viterbi algorithm and feature extraction. The approximate algorithm is thus faster than the Brute-Force way by using word-level information.

input : character sequence $C_{1:n}$
static templates T_s
dynamic templates T_d
number of labels m
trans matrix M
output: results Max & V_p

Initialize: weight matrix W ($n \times m$)
viterbi score matrix V_s ($n \times m$)
viterbi path matrix V_p ($n \times m$)
the index of current best label Max
for $i = 1 \cdots n$ **do**
  for t_s *in* T_s **do**
    // create feature string F_s $(Op.1)$
    F_s = createFeature($C_{1:n}$, t_s);
    $W[i]$ += getWeightVector(F_s);
  end
  // create a list of <pos_k,$word_k$,len_k>
  // $(k = 0, -1, -2 \ldots)$ $(Op.2)$
  $dList$ = getCurrentBestPath(Max, V_p);
  for t_d *in* T_d **do**
    // create dynamic features string F_d
    // $(Op.3)$
    F_d = createFeature($C_{1:n}$, t_d, $dList$);
    $W[i]$ += getWeightVector(F_d);
  end
  // Update $V_s[i]$, $V_p[i]$ $(Op.4)$
  viterbi_OneStep($V_s[i-1]$, $W[i]$, M);
  $Max = \arg\max_{\mathbf{i}}(V_s[i])$;
end

Algorithm 1: Tagging Algorithm with Dynamic Features

5 Training

Given an example $(\mathbf{x}, \mathbf{y})$, $\hat{\mathbf{y}}$ are denoted as the incorrect labels with the highest score

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{z} \neq \mathbf{y}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{z}). \quad (2)$$

The **margin** $\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$ is defined as

$$\gamma(\mathbf{w}; (\mathbf{x}, \mathbf{y})) = \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) - \mathbf{w}^T \Phi(\mathbf{x}, \hat{\mathbf{y}}). \quad (3)$$

Thus, we calculate the **hinge loss** $\ell(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$, (abbreviated as ℓ_w) by

Table 4: Example for Chinese-English Mixed POS Tagging

微	软	和	Apple	的	OS	风	格	不	同	。
B-NR	E-NR	S-CC	S-NR	S-DEG	S-NN	B-NN	E-NN	B-VA	E-VA	S-PU

$$\ell_w = \begin{cases} 0, & \gamma(\mathbf{w};(\mathbf{x},\mathbf{y})) > 1 \\ 1-\gamma(\mathbf{w};(\mathbf{x},\mathbf{y})), & \text{otherwise} \end{cases} \quad (4)$$

In round k, the new weight vector $\mathbf{w}_{k+1}$ is calculated by

$$\mathbf{w}_{k+1} = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}-\mathbf{w}_k||^2 + \mathcal{C}\cdot\xi, \quad \textbf{s.t. } \ell(\mathbf{w};(\mathbf{x}_k,\mathbf{y}_k)) <= \xi \textbf{ and } \xi >= 0 \quad (5)$$

where ξ is a non-negative slack variable, and $\mathcal{C}$ is a positive parameter which controls the influence of the slack term on the objective function.

Following the derivation in PA (Crammer et al., 2006), we can get the update rule,

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \tau_k(\Phi(\mathbf{x}_k,\mathbf{y}_k) - \Phi(\mathbf{x}_k,\hat{\mathbf{y}_k})), \quad (6)$$

where

$$\tau_k = \min(\mathcal{C}, \frac{\ell_{w_k}}{\|\Phi(\mathbf{x}_k,\mathbf{y}_k) - \Phi(\mathbf{x}_k,\hat{\mathbf{y}_k})\|^2}) \quad (7)$$

Our algorithm based on PA algorithm is shown in Algorithm 2.

input : training data sets:
$(\mathbf{x}_i,\mathbf{y}_i), i = 1,\cdots,N$, and parameters:
$\mathcal{C}, K$
output: $\mathbf{w}_K$

Initialize: **wTemp** $\leftarrow 0, \mathbf{w} \leftarrow 0$;
for $k = 0 \cdots K-1$ **do**
  for $i = 1 \cdots N$ **do**
    receive an example $(\mathbf{x}_i,\mathbf{y}_i)$;
    predict: $\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y}} \langle \mathbf{w}_k, \Phi(\mathbf{x}_i,\mathbf{y})\rangle$;
    if $\hat{\mathbf{y}}_i \neq \mathbf{y}_i$ **then**
      update $\mathbf{w}_{k+1}$ with Eq. 6;
    end
  end
  wTemp = **wTemp** + $\mathbf{w}_{k+1}$;
end
$\mathbf{w}_K$ = **wTemp**$/K$;

Algorithm 2: Training Algorithm

6 Experiments

We implement our system based on FudanNLP[1]. We employ the commonly used label set {B, I, E, S} for the segmentation part of cross-labels. {B, I, E} represent *Begin*, *Inside*, *End* of a multi-node segmentation respectively, and S represents a *Single* node segmentation.

The $F1$ score is used for evaluation, which is the harmonic mean of precision P (percentage of predict phrases that exactly match the reference phrases) and recall R (percentage of reference phrases that returned by system).

The feature templates, which are used to extract features, are listed in Table 5. We set traditional method (static features) as the baseline. The detailed experimental settings and results are reported in the following subsections.

[1]Available at `http://code.google.com/p/fudannlp/`

Table 5: Feature Templates

Static	$x_{i-2}y_i, x_{i-1}y_i, x_iy_i, x_{i+1}y_i, x_{i+2}y_i$
	$x_{i-1}x_iy_i, x_{i+1}x_iy_i, x_{i-1}x_{i+1}y_i,$
	$y_{i-1}y_i$
Dynamic	$pos_{i-2}pos_{i-1}y_i, pos_{i-1}pos_iy_i$
	$pos_{i-2}word_{i-1}y_i, pos_{i-1}word_iy_i$
	$pos_{i-1}word_{i-1}y_i, pos_iword_iy_i$
	$word_{i-2}word_{i-1}y_i, word_{i-1}word_iy_i$
	$word_ilen_iy_i$

6.1 POS Tagging for Chinese-only Texts

Before the experiments on Chinese-English mixed texts, we evaluate the performance of our method on Chinese-only texts. We use the CTB dataset from the POS tagging task of the Fourth International Chinese Language Processing Bakeoff (SIGHAN Bakeoff 2008)(Jin and Chen, 2008). The details are shown in Table 6.

The performance comparison on joint segmentation & POS tagging is shown in Table 7. Our method obtains an error reduction of 6.7% over the baseline. The reason is that our dynamic features can utilize

Table 6: POS Tagging Dataset in SIGHAN Bakeoff 2008

		Train Set (number)	Test Set (number)
Sentence		23444	2079
Word	Total	642246	59955
	NN	168896	16793
	NR	42906	3970
	VV	92887	8641
	VA	9106	649
	JJ	15640	1581

word-level information effectively and the feature templates are more flexible.

Table 7: Performances of POS Tagging on Chinese-only Texts with Static and Dynamic Features

Method	P	R	F1
Baseline	89.68	89.60	89.64
Our	**90.35**	**90.31**	**90.33**

6.2 POS Tagging for Chinese-English Mixed Texts

Without annotated corpus for Chinese-English mixed texts, we use synthetic data as the alternative. In Chinese-English mixed texts, English words of noun(NN/NR), verb(VV/VA) and adjective(JJ) categories are the most commonly used, so we randomly transform a certain percentage of Chinese words with these POS tags in the SIGHAN Bakeoff 2008 dataset(Jin and Chen, 2008) into their English counterparts.

6.2.1 Synthetic Data

Before trying out an experiment, we first study how to generate the data of mixed texts.

We use two ways to produce the synthetic data: "Respective Replacement" and "Unified Replacement".

Respective Replacement We replace the selected Chinese words into their corresponding English counterparts.

Unified Replacement We replace the selected Chinese words with a unified label ENG. The reason we use the label ENG instead of real words is that we want to consider the context of these words but not the words themselves and overcome the problem of out-of-vocabulary (OOV) English words.

For our experiments, we just select 5% of the Chinese nouns and verbs from SIGHAN dataset, and replace them in the above two ways. After replacement, the training and test data have 12780 and 1254 English words, respectively. 5189 words are generated by way of "Respective Replacement". In the test data, 326 words are OOV, which comprises 25% of the whole vocabulary. The information of generated data is shown in Table 8.

Table 8: The Synthetic Chinese-English Mixed Dataset H

Dataset		Numbers of ENG	
		NN	VV
H	Train Set	8191	4589
	Test Set	842	412

We use H_1 to represent the dataset generated by way of "Respective Replacement", and H_2 for the dataset by way of with "Unified Replacement". The experimental results on these two datasets are shown in Table 9.

Table 9: Performances of POS Tagging on Dataset H_1 and H_2

Method	Dataset	ENG	OOV	Total
		F1	$F1_{oov}$	F1
Baseline	H_1	73.60	54.91	88.93
	H_2	77.59	73.93	89.11
Our	H_1	75.60	54.60	89.79
	H_2	**79.82**	**77.61**	**89.81**

From Table 9, we can see that the "Unified Replacement" way is better than the "Respective Replacement" way for both the baseline and our method. The main reason is that the "Unified Replacement" way can greatly improve the tagging performance of OOV words.

6.2.2 Detail Comparisons

For detail comparisons of all situations of mixed texts, we design six synthetic datasets, $A/B/C/D_1/D_2/E$ by randomly selecting 10% or 15% of Chinese words ("NN/NR/VV/VA/JJ") in the

above SIGHAN Bakeoff 2008 dataset, and replacing them with English label ENG.

The differences of these datasets are as following:

- Dataset A only contains English words with tags "NN/VV".
- Dataset B contains English words with tags "NN/VV/VA".
- Dataset C contains one more tag "NR" than Dataset B.
- Datasets D_1 and D_2 contain one more tag "JJ" than Dataset B. The difference between D_1 and D_2 is that D_2 has about 50% more English words than D_1 in training set.
- Dataset E contains English words with all the tags "NN/NR/VV/VA/JJ".

The detailed information of datasets $A/B/C/D_1/D_2/E$ is shown in Table 10.

Table 10: The Synthetic Chinese-English Mixed Dataset

Dataset		Numbers of ENG				
		NN	NR	VV	VA	JJ
A	Train	16302	0	9007	0	0
	Test	1675	0	841	0	0
B	Train	16116	0	8882	906	0
	Test	1573	0	830	58	0
C	Train	16312	4057	9067	899	0
	Test	1549	400	795	61	0
D_1	Train	16042	0	8957	855	1539
	Test	1588	0	845	58	150
D_2	Train	23705	0	13154	1300	2211
	Test	1588	0	845	58	150
E	Train	16066	4162	9156	886	1547
	Test	1647	415	809	57	141

The results are shown in Table 11. On dataset E, our method achieves 6.78% higher performance on tagging ENG labels than traditional static features. This result is reasonable because our model can use more flexible feature templates to extract features and reduce the problem of being dependent on specific English words.

Tables 12/13/14/15/16/17 show the detailed results on datasets $A/B/C/D_1/D_2/E$.

Table 11: Performances of POS Tagging on Datasets $A/B/C/D_1/D_2/E$

Dataset	Method	ENG labels	Total
		F1	F1
A	Baseline	80.25	88.74
	Our	**83.03**	**89.72**
B	Baseline	76.72	88.51
	Our	**80.54**	**89.55**
C	Baseline	68.16	88.13
	Our	**70.34**	**88.99**
D_1	Baseline	71.30	88.33
	Our	**74.02**	**89.15**
D_2	Baseline	69.59	88.09
	Our	**74.10**	**89.15**
E	Baseline	61.58	87.71
	Our	**68.36**	**88.83**

Experiment on dataset A gets the best result because "NN" and "VV" can be easily distinguished by its context. Sometimes, "VA" has the similar context with "VV", experiment on dataset B shows its influence. The performances on datasets $B/C/E$ descend in turn. The reason is that words with tag "NN" or "NR/JJ" have the similar usage/contexts in Chinese. Since we use the same form ENG instead of real words, there are no differences between these words, which leads to some errors. Though the datasets is generated randomly, we can see our method perform better on every dataset than the baseline.

Table 12: Performances on Dataset A

POS tag	Method	P	R	F1
NN	Baseline	84.36	86.33	85.33
	Our	85.37	89.91	**87.58**
VV	Baseline	71.45	68.13	69.75
	Our	77.53	69.32	**73.20**

Table 13: Performances on Dataset B

POS tag	Method	P	R	F1
NN	Baseline	84.89	80.36	82.56
	Our	83.51	88.87	**86.11**
VV	Baseline	65.90	72.65	69.11
	Our	75.75	67.35	**71.30**
VA	Baseline	36.84	36.21	36.52
	Our	51.02	43.10	**46.73**

Table 14: Performances on Dataset C

POS tag	Method	P	R	F1
NN	Baseline	73.77	78.24	75.94
	Our	76.84	77.99	**77.41**
VV	Baseline	61.67	66.79	64.13
	Our	64.94	67.80	**66.34**
NR	Baseline	55.22	37.00	44.31
	Our	55.65	50.50	**52.95**
VA	Baseline	63.64	34.43	44.68
	Our	60.00	39.34	**47.52**

Table 15: Performances on Dataset D_1

POS tag	Method	P	R	F1
NN	Baseline	77.15	81.42	79.23
	Our	76.70	88.54	**82.20**
VV	Baseline	67.53	64.50	65.98
	Our	79.65	59.76	**68.29**
JJ	Baseline	25.00	18.00	**20.93**
	Our	22.92	14.67	17.89
VA	Baseline	36.00	31.03	**33.33**
	Our	28.57	37.93	32.59

Table 16: Performances on Dataset D_2

POS tag	Method	P	R	F1
NN	Baseline	79.11	74.87	76.93
	Our	79.29	82.68	**80.95**
VV	Baseline	55.77	72.78	65.64
	Our	69.17	70.89	**70.02**
JJ	Baseline	27.27	12.00	16.67
	Our	34.38	22.00	**26.83**
VA	Baseline	37.21	27.59	**31.68**
	Our	52.17	20.69	29.63

6.3 POS Tagging for Mixed Texts with a Real Dataset

To investigate the actual performance, we collect a real dataset from Web, which consists of 142 representative Chinese-English mixed sentences. This dataset contains $4,238$ Chinese characters and 275 English words. Since we focus on the performance for English words, we only label the POS tags of the English words. Table 18 shows some examples in the real dataset of mixed texts.

Table 17: Performances on Dataset E

POS tag	Method	P	R	F1
NN	Baseline	72.41	68.85	70.59
	Our	71.18	84.88	**77.43**
VV	Baseline	63.65	59.09	61.28
	Our	76.19	55.38	**64.14**
JJ	Baseline	28.57	25.53	**26.97**
	Our	30.21	20.57	24.47
VA	Baseline	44.83	45.61	45.22
	Our	60.42	50.88	**55.24**
NR	Baseline	38.03	52.05	43.95
	Our	52.01	46.75	**49.24**

Table 18: Examples in Real Dataset of Mixed Texts

通过 [Ninja Cloud/NR] 的云服务，[Ninja Blocks/NR] 能与 [Facebook/NR]、[Twitter/NR]、[Dropbox/NR] 等无缝连接。
By using [Ninja Cloud/NR], [Ninja Blocks/NR] can connect to [Facebook/NR], [Twitter/NR], [Dropbox/NR].
你去 [follow/VV] 一下这个人的工作。
You should [follow/VV] this man's work.
强烈的视觉震撼!! 很 [COOL/VA]!
... very [COOL/VA]!

The information of the real dataset is shown in Table 19. If all involved English words are tagging as "NN", the precision is just 56%.

Table 19: The Numbers of English Words with Different Tags in Dataset R

Dataset	NN	VV	VA	NR
R	154	58	28	35

Since there is no noun-modifier "JJ" in our collected data. We use the models trained on dataset B and C to tag the real data. The results are shown in Table 20. The difference between model B and C is that model B regards all words with tag "NR" as "NN". Since it is difficult to distinguish between "NR" and "NN" merely according to the context, model B performs better than model C.

The detail results of model B and C are shown in Table 21 and 22.

Table 20: Performances of POS Tagging on R

Model	Method	ENG
		F1
B	Baseline	74.91
	Our	**82.55**
C	Baseline	70.91
	Our	**74.91**

Table 21: Performances of Model B on Dataset R

POS tag	Method	P	R	F1
NN	Baseline	88.62	78.31	83.15
	Our	91.67	87.30	**89.43**
VV	Baseline	48.31	74.14	58.50
	Our	60.53	79.31	**68.66**
VA	Baseline	78.95	53.57	63.83
	Our	84.21	57.14	**68.09**

Table 22: Performances of Model C on Dataset R

POS tag	Method	P	R	F1
NN	Baseline	80.25	81.82	81.03
	Our	84.56	81.82	**83.17**
VV	Baseline	54.88	77.59	64.29
	Our	61.25	84.48	**71.01**
VA	Baseline	84.62	39.29	53.66
	Our	88.24	53.57	**66.67**
NR	Baseline	56.52	37.14	44.83
	Our	55.17	45.71	**50.00**

7 Related Works

In recent years, POS tagging has undergone great development. The mainstream method is to regard POS tagging as sequence labeling problems (Rabiner, 1990; Xue, 2003; Peng et al., 2004; Ng and Low, 2004).

However, the analysis of Chinese-English mixed texts is rarely involved in previous literature. In the aspect of the general multilingual POS tagging, most works focus on modeling cross-lingual correlations and tagging multilingual POS on respective monolingual texts, not on mixed texts (Cucerzan and Yarowsky, 2002; Yarowsky et al., 2001; Naseem et al., 2009).

Since we choose to use dynamic word-level features to improve the performance of POS tagging, we also review some works on word-level features.

Semi-Markov Conditional Random Fields (semi-CRF) (Sarawagi and Cohen, 2004) is a model in which segmentation task is implicitly included into the decoding algorithm. In this model, feature representation would be more flexible than traditional CRFs, since features can be extracted from the previous/the next segmentation within a window of variable size. The problem of this approach lies in that the decoding algorithm depends on the predefined window size to exploit the boundaries of segmentations but not the real length of words.

Bunescu (2008) presents an improved pipeline model in which the output of the previous subtasks are considered as hidden variables, and the hidden variables together with their probabilities denoting the confidence are used as probabilistic features in the next subtasks. One shortcoming of this method is inefficiency caused by the calculation of marginal probabilities of features. The other disadvantages of the pipeline method are error propagation and the need of separate training of different subtasks in the pipeline. Another disadvantage of pipeline method is error propagation.

Jiang et al. (2008) proposes a cascaded linear model for joint Chinese word segmentation and POS tagging. With a character-based perceptron as the core, combined with real-valued features such as language models, the cascaded model can efficiently utilize knowledge sources that are inconvenient to incorporate into the perceptron directly. However, they use POS tags or word information in a Brute-Force way, which may suffer from the problem of time complexity.

Sun (2011) presents a stacked sub-word model for joint Chinese word segmentation and POS tagging. By merging the outputs of the three predictors (including one word-based segmenter) into sub-word sequences, rich contextual features can be approximately derived. The experiments are conducted to show the effectiveness of using word-based information.

The difference between the above methods and ours is that our word-level features are dynamically generated in the decoding stage without exhaustive or preprocessed word segmentation.

8 Conclusion

In this paper, we focus on Chinese-English mixed texts and use dynamic features for POS tagging. To overcome the problem of the lack of annotated corpus on mixed texts, our features use both local and non-local information and take advantage of the characteristics of Chinese-English mixed texts. The experiments demonstrate the effectiveness of our method. It should be noted that our method is also effective for the mixed texts of Chinese and any foreign languages since we use "Unified Replacement".

For future works, we plan to improve our approximate tagging algorithm to reduce error propagation. In addition, we will refer to an English dictionary to generate some useful features to distinguish between "NR" and "NN" in Chinese-English mixed texts and add some statistical features derived from English resources, such as the most common tag of each English word. We would also like to investigate these features in more applications of natural language processing, such as name entity recognition, information extraction, etc.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. We also thanks Amy Zhou for her help in spell and grammar checking. This work was funded by NSFC (No.61003091 and No.61073069), 863 Program (No.2011AA010604) and 973 Program (No.2010CB327900).

References

Razvan C. Bunescu. 2008. Learning with probabilistic features for improved pipeline models. In *EMNLP*, pages 670–679. ACL.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, March.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December.

Silviu Cucerzan and David Yarowsky. 2002. Bootstrapping a multilingual part-of-speech tagger in one person-day. In *proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL*, pages 897–904. The Association for Computer Linguistics.

C. Jin and X. Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Sixth SIGHAN Workshop on Chinese Language Processing*, page 69.

T. Naseem, B. Snyder, J. Eisenstein, and R. Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36(1):341–385.

H.T. Ng and J.K. Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based. In *Proceedings of EMNLP*, volume 2004, page 277.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lawrence R. Rabiner. 1990. Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sunita Sarawagi and William W. Cohen. 2004. Semimarkov conditional random fields for information extraction. In *NIPS*.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *ACL*, pages 1385–1394. The Association for Computer Linguistics.

F. Xia. 2000. The part-of-speech tagging guidelines for the Penn Chinese Treebank (3.0).

N. Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of*

the first international conference on Human language technology research, pages 1–8. Association for Computational Linguistics.