Character-based Joint Segmentation and POS Tagging for Chinese using Bidirectional RNN-CRF

Yan Shao and **Christian Hardmeier** and **Jörg Tiedemann**[*] and **Joakim Nivre**
Department of Linguistics and Philology, Uppsala University
[*]Department of Modern Languages, University of Helsinki
{yan.shao, christian.hardmeier, joakim.nivre}@lingfil.uu.se
jorg.tiedemann@helsinki.fi

Abstract

We present a character-based model for joint segmentation and POS tagging for Chinese. The bidirectional RNN-CRF architecture for general sequence tagging is adapted and applied with novel vector representations of Chinese characters that capture rich contextual information and lower-than-character level features. The proposed model is extensively evaluated and compared with a state-of-the-art tagger respectively on CTB5, CTB9 and UD Chinese. The experimental results indicate that our model is accurate and robust across datasets in different sizes, genres and annotation schemes. We obtain state-of-the-art performance on CTB5, achieving 94.38 F1-score for joint segmentation and POS tagging.

1 Introduction

Word segmentation and part-of-speech (POS) tagging are core steps for higher-level natural language processing (NLP) tasks. Given the raw text, segmentation is applied at the very first step and POS tagging is performed on top afterwards. As by convention the words in Chinese are not delimited by spaces, segmentation is non-trivial, but its accuracy has a significant impact on POS tagging. Moreover, POS tags provide useful information for word segmentation. Thus, modelling word segmentation and POS tagging jointly can outperform the pipeline models (Ng and Low, 2004; Zhang and Clark, 2008).

POS tagging is a typical sequence tagging problem over segmented words, while segmentation also can be modelled as a character-level tagging problem via predicting the labels that identify the word boundaries. Ng and Low (2004) propose a joint model which predicts the combinatory labels of segmentation boundaries and POS tags at the character level. Joint segmentation and POS tagging becomes a standard character-based sequence tagging problem and therefore the general machine learning algorithms for structured prediction can be applied.

The bidirectional recurrent neural network (RNN) using conditional random fields (CRF) (Lafferty et al., 2001) as the output interface for sentence-level optimisation (Bi-RNN-CRF) achieves state-of-the-art accuracies on various sequence tagging tasks (Huang et al., 2015; Ma and Hovy, 2016) and outperforms the traditional linear statistical models. RNNs with gated recurrent cells, such as long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent units (GRU) (Cho et al., 2014) are capable of capturing long dependencies and retrieving rich global information. The sequential CRF on top of the recurrent layers ensures that the optimal sequence of tags over the entire sentence is obtained.

In this paper, we model joint segmentation and POS tagging as a fully character-based sequence tagging problem via predicting the combinatory labels. The Bi-RNN-CRF architecture is adapted and applied. The Chinese characters are fed into the neural networks as vector representations. In addition to utilising the pre-trained character embeddings, we propose a concatenated n-gram-representation of the characters. Furthermore, lower-than-character level information, namely radicals and orthographical features extracted by convolutional neural networks (CNNs), are also incorporated and tested. Three datasets of different sizes, genres and with different annotation schemes are employed for evaluation. Our model is thoroughly evaluated and compared with the joint segmentation and POS tagging model in

ZPar (Zhang and Clark, 2010), which is a state-of-the-art joint tagger using structured perceptron and beam decoding. According to the experimental results, our proposed model outperforms ZPar on all the datasets in terms of accuracy.

The main contributions of this work include: 1. We apply the Bi-RNN-CRF model for general sequence tagging to joint segmentation and POS tagging for Chinese and achieve state-of-the-art accuracy. The experimental results show that our tagger is robust and accurate across datasets of different sizes, genres and annotation schemes. 2. We propose a novel approach for vector representations of characters that leads to substantial improvements over the baseline model. 3. Additional improvements are obtained via exploring the feasibility of utilising lower-than-character level information. 4. We provide an open-source implementation of our method along with pre-trained character embeddings.[1]

2 Model

2.1 Neural Network Architecture

Our baseline model is an adaptation of Bi-RNN-CRF. As illustrated in Figure 1, the Chinese characters are represented as vectors and fed into the bidirectional recurrent layers. The character representations will be described in detail in the following sections. For the recurrent layer, we employ GRU as the basic recurrent unit as it has similar functionalities but fewer parameters compared to LSTM (Chung et al., 2014). Dropout (Srivastava et al., 2014) is applied to the outputs of the bidirectional recurrent layers. The outputs are concatenated and passed to the first-order chain CRF layer. The optimal sequence of the combinatory labels is predicted at the end. There is a post processing step to retrieve both segmentation and POS tags from the combinatory tags.

2.2 Tagging Scheme

Following the work of Kruengkrai et al. (2009a), the employed tags indicating the word boundaries are `B`, `I`, `E`, `S` representing a character at the beginning, inside, end of a word or as a single-character word. The CRF layer models conditional scores over all possible combinatory labels given the input characters. Incorporating the transition scores between the successive labels, the optimal sequence can be obtained efficiently via the Viterbi algorithm both for training and decoding.

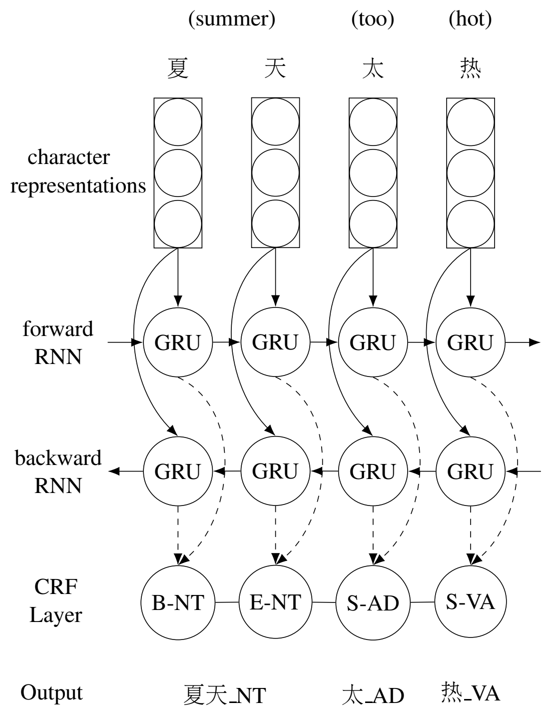

Figure 1: The Bi-RNN-CRF model for joint Chinese segmentation and POS tagging. The dashed arrows indicate that dropout layers are applied to the outputs of the recurrent layers.

The time complexity for the Viterbi algorithm is linear with respect to the sentence length n as $\mathcal{O}(kn)$, where k is constant and equals to the total number of combinatory labels. The efficiency can be improved if we reduce k. For some POS tags, combining them with the full boundary tags is redundant. For instance, only the functional word 的 can be tagged as `DEG` in Chinese Treebank (Xue et al., 2005). Since it is a single-word character, combinatory tags of `B-DEG`, `I-DEG`, and `E-DEG` never occur in the experimental data and should therefore be pruned to reduce the search space. In this paper, we prune the combinatory labels accordingly if the maximum length of the words in the training set under the corresponding POS tags is smaller than three.

2.3 Character Representations

We propose three different approaches to effectively represent Chinese characters as vectors for the neural network.

[1] https://github.com/yanshao9798/tagger

这是个特点

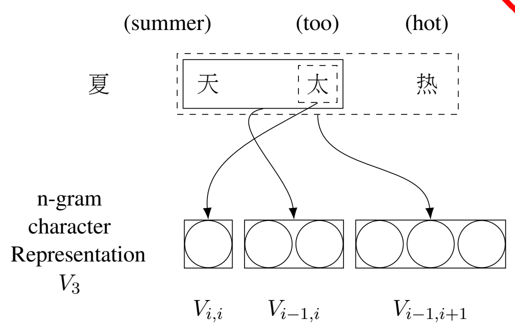

Figure 2: Vector representations of the Chinese characters as incrementally concatenated n-gram vectors in a given context.

2.3.1 Concatenated N-gram

The prevalent character-based neural models assume that larger spans of text, such as words and n-grams, can be represented by the sequence of characters that they consist of. For example, the vector representation $V_{m,n}$ of a span $c_{m,n}$ is obtained by passing the vector representations v_i of the characters c_i to a functions f as:

$$V_{m,n} = f(v_m, v_{m+1}, ..., v_n) \tag{1}$$

where f is usually an RNN (Ling et al., 2015) or a CNN (dos Santos and Zadrozny, 2014).

In this paper, instead of completely relying on the Bi-RNN to extract contextual features from context-free character representations, as demonstrated in Figure 2, we encode rich local information in the character vectors via employing the incrementally concatenated n-gram representation. In the example, the vector representation of the pivot character 太 in the given context is the concatenation of the context-free vector representation $V_{i,i}$ of 太 itself along with $V_{i-1,i}$ of the bigram 天太 as well as $V_{i-1,i+1}$ of the trigram 天太热.

Instead of constructing the vector representation $V_{m,n}$ of an n-gram $c_{m,n}$ from the character representations as in Equation 1, $V_{m,n}$ in different orders, such as $V_{i,i}$, $V_{i-1,i}$, and $V_{i-1,i+1}$, are randomly initialised separately. For each order, we use a single special vector to represent all the n-grams that only appear in the development and test sets but not in the training set. The n-grams in different orders are then concatenated incrementally to form up the vector representations of a Chinese character in the given context, which is passed further to the recurrent layers. As shown in Figure 2, the neighbouring characters on both sides of the pivot character are taken into account.

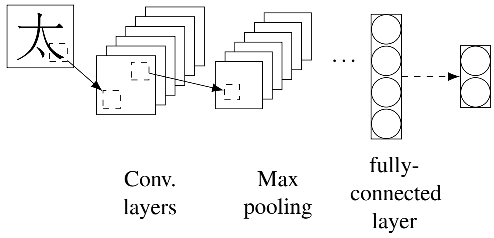

Figure 3: Convolutional Neural Networks for orthographical feature extraction. Only the first convolutional layer and its following max-pooling layer are presented.

2.3.2 Radicals and Orthographical Features

Chinese characters are logograms. As opposed to alphabetical languages, there is rich information encrypted in the graphical components. For instance, the Chinese characters that share the same part 钅 (gold) are all somewhat related to metals, such as 银 (silver), 铁 (iron), 针 (needle) and so on. The shared part 钅 is known as the radical, which functions as a semantic indicator. Hence, we investigate the effectiveness of using the information below the character level for our task.

First, the radicals are used as extra features. They are represented as randomly initialised vectors and concatenated as parts of the character representations. Radicals are traditionally used as indices in Chinese dictionaries. In our approach, radicals are retrieved via the unicode representation of Chinese characters as the characters that share the same radical are grouped together. They are organised in consistent with the categorisation in Kangxi Dictionary (康熙字典), in which all the Chinese characters are grouped under 214 different radicals. We only employ the radicals of the common characters in the unicode range of (U+4E00, U+9FFF). For the characters out of the range and the non-Chinese characters, we use a single special vector as their radical representations.

Additionally, instead of presuming that only radicals encode lower-than-character level information, we use convolutional neural networks (CNNs) to extract graphical features from scratch by regarding the Chinese characters as pictures and feed their pixels as the input. As illustrated in Figure 3, there are two convolutional layers, both followed by a max-pooling layer. The output of the second max-pooling convolutional layer is reshaped and passed to a regular fully-connected layer. Dropout is applied to the output of the fully-connected layer. The output is then concatenated as parts of the character representation. The CNNs are trained jointly with the main network.

2.3.3 Pre-trained Character Embeddings

The context-free vector representations of single characters introduced in section 2.3.1 can be replaced by pre-trained character embeddings retrieved from large corpora. We employ GloVe (Pennington et al., 2014) to train our character embeddings on Wikipedia[2] and the freely available Sogou News Corpora (SogouCS)[3]. We use a single random vector as the representation for the characters that are not in the embedding vocabulary. Pre-trained embeddings for higher-order n-grams are not employed in this paper.

2.4 Ensemble Decoding

At the final decoding phase, we use ensemble decoding, a simple averaging technique, to mitigate the deviations led by random weight initialisation of the neural network. For the chain CRF decoder, the final sequence of the combinatory tags y is obtained via the conditional scores $S(y_i|x_i)$ and the transition scores $T(y_i, y_j)$ given the input sequence x. Instead of computing the optimal sequence with respect to the scores returned by a single model, both the conditional scores and transition scores are averaged over four models with identical parameter settings that are trained independently:

$$y^* = \underset{y \in L(x)}{\operatorname{argmax}}\, p(y|x; \{\bar{S}\}, \{\bar{T}\}) \qquad (2)$$

Ensemble decoding is only applied to the best performing model according to the feature experiments at the final testing phase in this paper.

3 Implementation

Our neural networks are implemented using the TensorFlow r0.11 library (Abadi et al., 2016). We group the sentences with similar lengths into the same buckets and the sentences in the same bucket are padded to the same length accordingly. We construct sub-computational graphs respectively for each bucket. The training speed of our neural network on GPU devices can be drastically improved thanks to the bucket model. The training time is proportional to both the size of the training set and the number of POS tags.

Char. embedding size	64
n-gram embedding size	64
Radical embedding size	30
Character font	simsun (宋体)
Character size	30 × 30
GRU state size	200
Conv. filter size	5 × 5
Conv. filter number	32
Max pooling size	2 × 2
Fully-connected size	100
Optimiser	Adagrad
Initial learning rate	0.1
Decay rate	0.05
Gradient Clipping	5.0
Dropout rate	0.5
Batch size	10

Table 1: Hyper-parameters.

Table 1 shows the adopted hyper-parameters. We use one set of parameters for all the experiments on different datasets. The weights of the neural networks, including the embeddings, are initialised using the scheme introduced in Glorot and Bengio (2010). The network is trained with the error back-propagation algorithm. All the embeddings are fine-tuned during training by back-propagating gradients. Adagrad (Duchi et al., 2011) with mini-batches is employed for optimisation with the initial learning rate $\eta_0 = 0.1$, which is updated with a decay rate $\rho = 0.05$ as $\eta_t = \frac{\eta_0}{\rho(t-1)+1}$, where t is the index of the current epoch.

We use early stopping (Yao et al., 2007) with respect to the performance of the model on the development sets. F1-scores of both segmentation ($F1_{Seg}$) and joint POS tagging ($F1_{Seg\&Tag}$) are employed as $F1_{Seg} * F1_{Seg\&Tag}$ to measure the performance of the model after each epoch during training. In our experiments, the models are trained for 30 epochs. To ensure that the weights are well optimised, we only adopt the best epoch after the model is trained at least for 5 epochs.

4 Experiments

4.1 Datasets

We employ three different datasets for our experiments, namely Chinese Treebank (Xue et al., 2005) 5.0 (CTB5) and 9.0 (CTB9) along with the Chinese section in Universal Dependencies (UD Chinese) (Nivre et al., 2016) of version 1.4.

[2] https://dumps.wikimedia.org/
[3] http://www.sogou.com/labs/resource/cs.php

CTB5 is the most employed dataset for joint segmentation and POS tagging in previous research. It is composed of newswire data. We follow the conventional split of the dataset as in Jiang et al. (2008); Kruengkrai et al. (2009a); Zhang and Clark (2010). CTB9 consists of source texts in various genres. We split the dataset by referring to the partition of CTB7 in Wang et al. (2011). We extend the training, development and test sets from CTB5 by adding 80% of the new data in CTB9 to training and 10% each to development and test. The double-checked files are all placed in the test set. The detailed splitting information can be found in Table 10 in Appendix. UD Chinese has both universal and language-specific POS tags. They are not predicted jointly in this paper. For the sake of convenience, we refer the universal tags as UD1 and the language-specific ones as UD2 in the following sessions. In order to make the model benefit more from the pre-trained character embeddings, we convert the texts in UD Chinese from traditional Chinese into simplified Chinese.

Table 2 shows brief statistics of the employed datasets in numbers of words. The out-of-vocabulary (OOV) words are counted regardless of the POS tags. We can see that the size of UD Chinese is much smaller and it has a notably higher OOV rate than the two CTB datasets.

	CTB5	CTB9	UD Chinese
Train	493,935	1,696,322	98,608
Dev	6,821	136,468	12,663
Test	8,008	242,317	12,012
OOV rate (dev)	8.11	2.93	12.13
OOV rate (test)	3.47	3.13	12.46

Table 2: Statistics of the employed datasets in numbers of words.

4.2 Experimental Results

Both segmentation (Seg) and joint segmentation and POS tagging (Seg&Tag) are evaluated in our experiments.[4] We employ recall (R), precision (P) and F1-score (F) as the evaluation metrics. A series of feature experiments are carried out on the development sets to evaluate the effectiveness of the proposed approaches for vector representations of the characters. Finally, the best performing model according to the feature experiment is applied to the test sets in the forms of single as well as ensemble and compared with ZPar.

4.2.1 Feature Experiments

Table 3 shows the evaluation results of using concatenated n-grams up to different orders as the character representations. By introducing 2-grams, we can obtain vast improvements over solely using the conventional character embeddings, which indicates that not all the local information can be effectively captured by the BiRNN using context-free character representations. Utilising the concatenated n-grams ensures that the same character has different but yet closely related representations in different contexts, which is an effective way to encode contextual features.

From the table, we see that notable improvements can be achieved further via employing 3-grams. 4-grams still help but only to CTB9 while adding 5-grams achieves almost no improvement on any of the datasets. The results imply that concatenating higher-order n-grams can be detrimental, especially on datasets in smaller sizes due to the fact that higher-order grams are more sparse in the training data and their vector representations cannot be trained well enough. Besides, adopting higher-order grams also substantially increases the numbers of weights and therefore both training and decoding become less efficient. Under the circumstances, we consider that 3-gram model is optimal for our task and it is employed in the following experiments for all the datasets.

The concatenated n-grams has a bigger size compared to the basic character representation. We conduct one additional experiment using a basic 1-gram character model with a larger character vector size of 300. The evaluation scores are similar to the basic character model with the size of 64, which shows that the improvements obtained by the n-gram model is not provoked by enlarging the size of the vector representation.

The evaluation scores of the lower-than-character level features are reported in Table 4. The relevant features are added on top of the 3-gram model. Employing radicals and graphical features achieves similar improvements for segmentation while utilising radicals obtains better results for joint POS tagging on CTB5. However, radicals are not a very effective feature on CTB9, UD1 and UD2 whereas a notable enhancement is observed when employing graphical fea-

[4]The evaluation script is downloaded from: http://people.sutd.edu.sg/ yue_zhang/doc/doc/joint_files /evaluate.py

	CTB5		CTB9		UD1		UD2	
	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag
size = 300	95.22	91.71	95.53	90.89	91.84	85.43	92.40	85.63
1-gram	95.14	91.52	95.25	90.43	91.74	85.07	91.83	84.93
2-gram	97.08	93.72	96.30	91.66	**94.50**	88.36	94.42	88.14
3-gram	**97.14**	94.01	96.47	91.75	94.36	88.27	**94.43**	**88.32**
4-gram	97.13	**94.02**	96.48	**91.89**	94.25	88.37	94.16	88.24
5-gram	96.94	93.84	**96.50**	91.88	94.40	**88.47**	94.25	88.03

Table 3: Evaluation of concatenated n-gram representations on the development sets in F1-scores

	CTB5		CTB9		UD1		UD2	
	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag
3-gram	97.14	94.01	96.47	91.75	94.36	88.27	**94.43**	88.32
+radicals	**97.26**	**94.42**	96.42	91.74	94.37	88.21	94.39	**88.36**
+graphical	97.25	94.08	**96.50**	**91.78**	**94.50**	**88.59**	94.23	87.95

Table 4: Evaluation of lower-than-character level features on the development sets in F1-scores.

	CTB5		CTB9		UD1		UD2	
	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag
1-gram	95.14	91.52	95.25	90.43	91.74	85.07	91.83	84.93
+GloVe	95.82	92.45	95.44	90.57	92.77	86.48	93.01	86.48
3-gram, radicals	97.26	94.42	96.42	91.74	94.37	88.21	94.39	88.36
+GloVe	**97.42**	**94.58**	**96.56**	**91.96**	**95.12**	**89.69**	**95.02**	**89.20**

Table 5: Evaluation of the pre-trained character embeddings on the development sets in F1-scores.

tures on UD1. Using CNNs to extract graphical features is computationally much more expensive than simply adopting radicals via a lookup table, especially when GPU is not available. Adopting radicals should therefore be prioritised if the computational resources are limited.

From Table 5, we can learn that employing pre-trained embeddings as initial vector representations for the characters achieves improvements in general, whereas the improvements are comparatively smaller if the the concatenated n-gram representations and the radicals are added. Additionally, the improvements obtained on UD Chinese are more significant than on CTBs, which indicates that the pre-trained character embeddings are more beneficial to the datasets in smaller sizes.

In general, the feature experiments indicate that the proposed Chinese character representations are all sensitive to dataset size. Using higher-order n-grams requires more data for training. On the other hand, the pre-trained embeddings are more vital if the dataset is small. In addition, the different representations are sensitive to tagging schemes as the evaluation results on UD1 and UD2 are quite diverse. Taking both robustness and efficiency into consideration, we select 3-grams along with radicals and pre-trained character embeddings as the best setting for final evaluation.

4.2.2 Final Results

Table 6 shows the final scores on the test sets. The complete evaluation results in precision, recall and F1-scores are contained in Table 11 and Table 12 in Appendix. Our system is compared with ZPar. We retrained a ZPar model on CTB5 that reproduces the evaluation scores reported in Zhang and Clark (2010). We also modified the source code so that it is applicable to CTB9 and UD Chinese. In addition, we perform the mid-p McNemar's test (Fagerland et al., 2013) to examine the statistical significances.

As shown in Table 6, the single model is worse than the ensemble model but still outperforms ZPar on all the tested datasets. ZPar incorporates discrete local features at both character and word levels and employs structured perceptron for global optimisation, whereas we encode rich local information in the character representations and employ BiRNN to effectively extract global features and capture long term dependencies. The chain CRF layer is used for sentence-level optimisation, which functions similarly to structured perceptron. As opposed to the taggers built with traditional machine learning algorithms,

		CTB5		CTB9		UD1		UD2	
		Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag	Seg	Seg&Tag
ZPar		97.77	93.82	96.28	91.62	93.75	88.11	93.98	88.16
Single (3-gram, rad., GloVe)		97.89	94.07**	96.47**	91.89**	94.85**	89.41**	94.93**	89.00**
Ensemble (4 models)		**98.02***	**94.38****	**96.67****	**92.34****	**95.16****	**89.75***	**95.09***	**89.42****
OOV recall	ZPar	76.98	68.34	**75.83**	63.71	78.69	64.40	79.56	64.86
	Single	**78.78**	69.78	74.16	62.58	81.36	67.40	81.16	66.73
	Ensemble	77.34	**70.50**	75.52	**64.14**	**82.16**	**68.14**	**81.56**	**68.00**

Table 6: Evaluations of the best model on the final test sets in F1-scores as well as the recalls of out-of-vocabulary words. Significance tests for Single are in comparison to ZPar, while tests for Ensemble are in comparison to Single ($^{**}p < 0.01$, $^{*}p < 0.05$)

our model avoids heavy feature engineering and benefits from large plain texts via utilising pre-trained character embeddings. It is also very flexible to add lower-than-character level features as parts of the character representations. The model performs very well despite being fully character based. Moreover, it has clear advantages when applied to smaller datasets like UD Chinese, while the prevalence is much smaller on CTB5.

Both our model and ZPar segment OOV words in UD Chinese with higher accuracies than the ones in CTBs despite that UD Chinese is notably smaller and the overall OOV rate is higher. Compared to CTB, the words in UD Chinese are more fine-grained and the average word length is shorter, which makes it easier for the tagger to correctly segment the OOV words as Zhang et al. (2016) show that the longer words are more difficult to be segmented correctly. For joint POS tagging for OOV words, the two systems both perform significantly better on CTB5 as it is only composed of news text.

In general, our model is more robust to OOV words than ZPar, except that ZPar yields better result for segmentation by a small margin on CTB9. ZPar also obtains higher accuracy for joint POS tagging than the single model on CTB9. The differences between ZPar and our model for both segmentation and POS tagging are more substantial on UD Chinese, which indicates that our model is relatively more advantageous for handling OOV words when the training sets are small, whereas ZPar is able to perform equally well when substantial amount of training data is available as they achieve similar results on the CTB sets.

The single model is further improved by ensemble-averaging four independently trained models. The improvements are not drastic but they are observed systematically across all the datasets. In general, ensemble decoding is beneficial to handling OOV words as well except that a small drop for segmentation on CTB5 is observed.

	Ensemble		ZPar	
	Seg	Seg&Tag	Seg	Seg&Tag
BN	**97.89***	**94.48****	97.68	94.22
CS	**96.67****	**91.78****	95.61	90.15
FM	**96.54****	**91.92****	96.30	91.51
MG	**94.54****	**89.23****	94.22	88.60
NS	**97.56**	**93.92****	97.49	93.70
SM	**96.43****	**91.78****	96.13	90.32
SP	**97.29****	**93.93****	96.69	93.35
WB	**94.27****	**88.44****	93.38	86.88

Table 7: Evaluation on Broadcast News (BN), Conversations (CS), Forum (FM), Magazine (MG), News (NS), Short Messages (SM), Speech (SP) and Weblogs (WB) in CTB9. ($^{**}p < 0.01$, $^{*}p < 0.05$)

	Words	UW rate	OOV rate	OOV recall
BN	29,829	14.30	2.61	80.03
CS	42,713	10.40	2.50	79.85
FM	37,409	16.08	2.61	70.52
MG	29,003	22.95	5.96	74.28
NS	42,351	14.24	2.97	80.48
SM	28,683	13.50	3.06	72.41
SP	27,797	7.41	2.29	71.54
WB	4,532	38.83	5.49	75.10

Table 8: Numbers of words, UW rate, OOV rate and OOV recall of the decomposed test sets in different genres of CTB9. OOV recalls are obtained by the ensemble model.

Table 7 displays the evaluation of the ensemble model and ZPar on the decomposed test sets of CTB9 in different genres. Our model surpasses ZPar on all the genres in both segmentation and joint POS tagging. The differences are subtle on the genres in which the texts are normalised, such as News and Broadcast News. This, to a very large extent, explains why our model is only marginally better than ZPar on CTB5, whereas the experimental results reveal that our model is substantially better at processing non-standard text as it yields significantly higher scores on Conversations, Short Messages and Weblogs.

The evaluation results of both our model and ZPar vary substantially across different genres, implying that the texts in some genres are fundamentally more challenging to process. We report unique word (UW) rate and OOV rate in Table 8 as we see that they are correlated with the evaluation scores in Table 7. For instance, the models achieve higher scores on Broadcast News that has both low UW rate and OOV rate (the OOV recall is relatively high), which is in contrast to Magazine. In addition, regardless of the very low OOV recall, the model attains very nice performance on Speech because of the low UW rate and OOV rate.

Our models are compared with the previous best-performing systems on CTB5 in Table 9. Our models are not optimised particularly with respect to CTB5 but still yield competitive results, especially for joint POS tagging. We are the first to report evaluation scores on CTB9 and UD Chinese.

	Seg	Seg&Tag
Kruengkrai et al. (2009b)	0.9798	0.9400
Zhang and Clark (2010)	0.9778	0.9367
Sun (2011)	**0.9817**	0.9402
Wang et al. (2011)	0.9811	0.9418
Shen et al. (2014)	0.9802	0.9380
Single	0.9789	0.9407
Ensemble	0.9802	**0.9438**

Table 9: Result comparisions on CTB5 in F1-scores.

5 Related Work

The fundamental BiRNN-CRF architecture is task-independent and has been applied to many sequence tagging problems on Chinese. Peng and Dredze (2016) adopt the model for Chinese segmentation and named entity recognition in the context of multi-task and multi-domain learning. Dong et al. (2016) employ a character level BiLSTM-CRF model that utilises radical-level information for Chinese named entity recognition. Ma and Sun (2016) use a similar architecture but feed the Chinese characters pair-wise as edge embeddings instead. Their model is applied respectively to chunking, segmentation and POS tagging.

Zheng et al. (2013) model joint Chinese segmentation and POS tagging via predicting the combinatory segmentation and POS tags. They employ the adaptation of the feed forward neural network introduced in Collobert et al. (2011) that only extracts local features in a context window. A perceptron-style training algorithm is employed for sentence level optimisation, which is the same as the training algorithm of the BiRNN-CRF model. Kong et al. (2015) apply segmental recurrent neural networks to joint segmentation and POS tagging but the evaluation results are substantially below the state-of-the-art on CTB5.

Sun et al. (2014) attempt to encode radical information into the conventional character embeddings. The radical-enhanced embeddings are employed and evaluated for Chinese segmentation. The results show that radical-enhanced embeddings outperform both skip-ngram and continues bag-of-word (Mikolov et al., 2013) in word2vec.

6 Conclusion

We adapt and apply the BiRNN-CRF model for sequence tagging in NLP to joint Chinese segmentation and POS tagging via predicting the combinatory tags of word boundaries and POS tags. Concatenated n-grams as well as lower-than-character features are employed along with the conventional pre-trained character embeddings as the vector representations for Chinese characters. The feature experiments indicate that concatenated n-grams contribute substantially. However, both radicals and graphical features as lower-than-character level information are less effective. How to incorporate the lower-than-character level information more effectively will be further explored in the future.

The proposed model is extensively evaluated on CTB5, CTB9 and UD Chinese. Despite the fact that different character representation approaches are sensitive to data size and tagging schemes, we use one set of hyper-parameters and universal feature settings so that the model is robust across datasets. The experimental results on the test sets show that our model outperforms ZPar which is built on structured perceptron on all the datasets. We obtain state-of-the-art performances on CTB5. The results on UD Chinese and CTB9 also reveal that our model has great advantages in processing non-standard text, such as weblogs, forum text and short messages.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. Savannah, Georgia, USA.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(August):2493–2537.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *International Conference on Computer Processing of Oriental Languages*. Springer, pages 239–250.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of The 31st International Conference on Machine Learning*. pages 1818–1826.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Morten W Fagerland, Stian Lydersen, and Petter Laake. 2013. The mcnemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC medical research methodology* 13(1):91.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. pages 249–256.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

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 *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Citeseer.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2015. Segmental recurrent neural networks. *arXiv preprint arXiv:1511.06018* .

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009a. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, Singapore, pages 513–521.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, WANG Yiou, Kentaro Torisawa, and Hitoshi Isahara. 2009b. Joint chinese word segmentation and pos tagging using an error-driven word-character hybrid model. *IEICE transactions on information and systems* 92(12):2298–2305.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*. volume 1, pages 282–289.

Wang Ling, Chris Dyer, W. Alan Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1520–1530.

Shuming Ma and Xu Sun. 2016. A new recurrent neural crf for learning non-linear edge features. *arXiv preprint arXiv:1611.04233* .

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, page 10641074.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the*

2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain, pages 277–284.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. pages 1659–1666.

Nanyun Peng and Mark Dredze. 2016. Multi-task multi-domain representation learning for sequence tagging. *arXiv preprint arXiv:1608.02689* .

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, pages 1532–1543.

Mo Shen, Hongxiao Liu, Daisuke Kawahara, and Sadao Kurohashi. 2014. Chinese morphological analysis with character-level pos tagging. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 253–258.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, Portland, Oregon, USA, pages 1385–1394.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *International Conference on Neural Information Processing*. Springer, pages 279–286.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, pages 309–317.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering* 11(02):207–238.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation* 26(2):289–315.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 421–431.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Columbus, Ohio, pages 888–896.

Yue Zhang and Stephen Clark. 2010. A fast decoder for joint word segmentation and pos-tagging using a single discriminative model. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Massachusetts, USA, pages 843–852.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging pages 647–657.

Appendix

Dataset	CTB chapter IDs
Train	0044-0143, 0170-0270, 0400-0899, 1001-1017, 1019, 1021-1035, 1037-1043, 1045-1059, 1062-1071, 1073-1117, 1120-1131, 1133-1140, 1143-1147, 1149-1151, 2000-2915, 4051-4099, 4112-4180, 4198-4368, 5000-5446, 6000-6560, 7000-7013
Dev	0301-0326, 2916-3030, 4100-4106, 4181-4189, 4369-4390, 5447-5492, 6561-6630, 7013-7014
Test	0001-0043, 0144-0169, 0271-0301, 0900-0931, 1018, 1020, 1036, 1044, 1060, 1061, 1072, 1118, 1119, 1132, 1141, 1142, 1148, 3031-3145, 4107-4111, 4190-4197, 4391-4411, 5493-5558, 6631-6700, 7015-7017

Table 10: The split of Chinese Treebank 9.0

		P	R	F
CTB5	Single	97.49	98.30	97.89
	Ensemble	97.57	98.47	98.02
CTB9	Single	96.38	96.55	96.47
	Ensemble	96.61	96.74	96.67
UD1	Single	94.71	94.99	94.85
	Ensemble	95.07	95.27	95.17
UD2	Single	94.98	94.93	94.93
	Ensemble	95.00	95.22	95.11

Table 11: Evaluation of segmentations in precision, recall and F1-scores

		P	R	F
CTB5	Single	93.68	94.47	94.07
	Ensemble	93.95	94.81	94.38
CTB9	Single	91.81	91.97	91.89
	Ensemble	92.28	92.40	92.34
UD1	Single	89.28	89.54	89.41
	Ensemble	89.67	89.86	89.77
UD2	Single	88.95	89.04	89.00
	Ensemble	89.33	89.54	89.43

Table 12: Evaluation of joint segmentations and POS tagging in precision, recall and F1-scores