POS tagging of Chinese Buddhist texts using Recurrent Neural Networks

Longlu Qin
Department of East Asian Languages and Cultures
longlu@stanford.edu

Abstract

Chinese POS tagging, as one of the most important problems in the NLP community, has been investigated in the past decades. This project, for the first time in the literature, tests different neural network models on a Chinese Buddhist contexts, which are the representative for the Medieval Chinese. Our results demonstrate the capacity of neural network models, and the results are than the popular trigram HMM model in the literature. Differences between the Buddhist texts and modern Chinese data are also revealed by the experiments. Lastly, we also propose several interesting topics for future research.

1 Introduction

As cited and described in [Lee and Kong, 2014],The Chinese Buddhist Canon consists of a collection of translations of Buddhist texts from Indic languages to Chinese from the 2nd to the 11th centuries CE. With a total of over 52 million characters, it is one of the most important linguistic data representing the evolution of the Chinese language from Middle Chinese (220 CE to 960 CE) to Early Modern Chinese (960 CE to 1900 CE) [Sun, 2006], including the process of disyllabication and changes in lexical meanings, as well as syntactic structures [Zhu, 2010, Jiang and Hu, 2013]. Despite its linguistic significance, the volume of the Canon makes it infeasible to manually collect quantitative evidence and analyze linguistic phenomenon over the entire corpus.

Recently, digitalized version of the Canon has enabled computation of n-gram counts and distribution [Lancaster, 2010]. However, for the research interests in Chinese historical linguistics, a better annotated corpus with information such as the syntactic annotation is required to effectively utilize digitalized historical texts for data collection and analysis. Due to the significant difference in lexical meaning and syntactic structures between Medieval Chinese and Modern Chinese, the existing parsers and POS taggers (such as the Stanford Chinese parser and the Stanford part-of-speech tagger), which were trained on Modern Chinese, do not perform well on automatically adding syntactic annotation to the Chinese Buddhist texts. As an initiative in incorporating Natural Language Processing techniques into the methodology of historical linguistics research, we hope to build parsers and POS taggers with high accuracy that can be applied to the Chinese Buddhist texts, and eventually to other historical texts of different genres and times.

As a very first step, our goal in this project is to utilize a treebank of Chinese Buddhist texts to build a POS tagger for Chinese Buddhist texts, based on the idea of neural networks. Ideally we would also like to shed some light on the difference between Medieval Chinese and Modern Chinese the the linguistic modeling level, but due to time limit, advanced analysis would be left for future work.

One of the main hurdle of this project is the lack of well-labeled data. In general, computational methods require a significant amount of training data only upon which an accurate model could be built. Recently, a manually created dependency treebank of Chinese Buddhist texts (referred to as "the treebank" hereafter), released by Lee and Kong [2014], makes this task more promising.

The rest of this report is organized as follows: Section 2 is devoted to introducing background and our problem setting. We explain our learning algorithms in Section 3. Experiments are reported in Section 4. Lastly, Section 5 concludes the report and discusses several possible directions for future work.

2 Background and problem setting

POS tagging has been investigated in the NLP literature in the past decades. Different methods have been tested on this task, including SVM[Giménez and Marquez, 2004], decision tree[Schmid and Laws, 2008], HMM[Kupiec, 1992], conditional random field autoencoders[Ammar et al., 2014] and so on. When applied to the task of Modern Chinese POS tagging, lower accuracies have been reported, although many fine-grained techniques created for Chinese have been applied [Zhao and Wang, 2002, Ng and Low, 2004, Huang et al., 2007, Huang and Harper, 2009, Hatori et al., 2011, Sun and Uszkoreit, 2012, Zhang et al., 2014]. Jointly learning the parsing structure and POS tagging has also been investigated [Wang and Xue, 2014, Li et al., 2014]. Our project only focuses on POS tagging.

Among these works, the perceptron idea is most related to our project. Recently, perceptron model is used for the task of modern Chinese POS tagging [Zhang et al., 2014]. [Zhang et al., 2014] investigates the effects of different regularizations on the neural networks model. However, no exploration about the network structure is done in the paper. We take one step further in this project to compare the performances of different neural network models.

Moreover, this project is for the first time in the literature, to our best knowledge, to investigate the performances of different neural networks models on the task of POS tagging on the Chinese Buddhist texts. We would expect POS tagging in medieval Chinese to be a more difficult task, because there can be more ambiguity of the grammatical function and lexical meaning of a word. Old Chinese and Middle Chinese (before 960 CE) in general have more possibilities in terms of the POS tagging of one word. For example, all of the cases of the word 'shi' in Penn Chinese Treebank were tagged as copula (VC) Xia [2000]. However, in the Buddhist treebank, the tagging for 'shi' includes copula (VC), determiner (DT), common nouns (NN), adverb (AD), pronoun (PN), and predicative adjective (VA). Note that the Chinese Buddhist texts, including the treebank itself, are documented in Chinese characters. Therefore, the notorious problem that the romanization of one Chinese syllable can represent many different characters across the four tones is not relevant to our task at hand.

In general, our task is to build a tagger that can add POS labels to new data based on the input data. The input data X is a list of sentences, $X = \{s_1, s_2, \dots, s_m\}$, where each s_i denotes the i-th sentence in the dataset. Each sentence consists of its words and their corresponding POS labels. We denote the i-th word in the t-th sentence by $w_i^{(t)}$. The real tag of the word w is denoted by y_w, and its prediction is denoted by $\widehat{y}_w$. Denote the word embedding in the vector space as L. The vector for word w is simply denoted by L_w. We add a "$\langle s \rangle$" (resp. "$\langle \backslash s \rangle$") to the beginning (resp. end) of every sentence.

Note that in Chinese, each word may have different number of characters. We reserve the segmentation from the treebank, i.e. all the sentences have been well segmented. Significant improvement is reported in the literature of Modern Chinese POS tagging by exploring the tree structure. However, we do not import the tree structure from the treebank, since the task of parsing is more advanced, and reserved for future work. We do import the segmentation of the characters from the treebank to omit this step before tagging.

Our tagger, in the end, will take a new sentence (well segmented) as its input, and returns POS labels for each word in the sentence.

3 Learning models

We test several models on the treebank, including majority voting(MV), trigram HMM(tri-HMM), 2-layers Neural Networks(2-layers NN), Recurrent NN (RNN), bi-directional Recurrent NN(RNN_bidirect), and trigram RNN(tri-RNN).

3.1 Baselines

We treat the majority voting method and tri-HMM as our baseline. Words that appear only once in the training set is treated as the same unknown word "UNK".

The majority voting method is a naive memorizing method. Given a test word (or words in a test sentence), the MV method randomly generate a tag according to the empirical distribution in the training data,

$$\mathbb{P}\left(\widehat{y}_w = u\right) = \mathbb{P}\left(y_w = u\right)/\mathbb{P}\left(w\right) = \frac{\sum_{i,t} I_{y_{w_i^t}=u \,\&\, w_i^t=w}}{\sum_{i,t} I_{w_i^t=w}}.$$

For a word that never appears in the training data, we treat it as a unknown word "UNK".

The trigram-HMM model is popular in the literature of POS tagging for Modern Chinese. It has been demonstrated that tri-HMM significantly outperform normal HMM. The model of trigram HMM is illustrated in 1. We use the package that is implemented by Guo [2013].

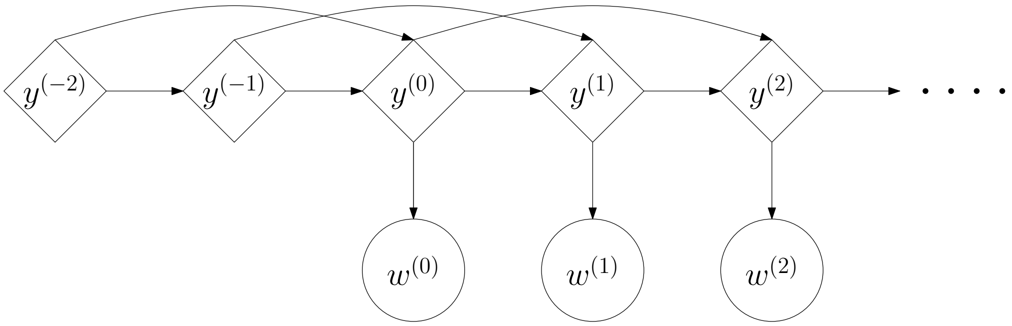

Figure 1: Trigram HMM

3.2 2-layers Neural Networks

We also try a simple neural networks for this task as a discriminative model. We use the model from Assignment 2 of this course. The window size is picked as 3. Thus, the input for the word $w_i^{(t)}$ is $(L_{w_{i-1}^{(t)}}, L_{w_i^{(t)}}, L_{w_{i+1}^{(t)}})$. The number of hidden unit will be mildly tuned.

3.3 RNN models

In total, 3 RNN models are tested in the project[1]. We use the regular RNN model from assignment 2 of this course. However, in order to get better training procedures, we add a regularization term for all the weight matrices (and the corresponding term for the gradient). We decay the learning rate of the SGD algorithm. Dropout technique is used for all the RNN models. We also add offset terms to make the models more flexible.

[1] We omit all the detailed training rules (step size, gradient update etc.) in the report. All the implementation pass the gradient check.

RNN model is then modified to a bidirectional RNN model. In particular,

$$\begin{aligned}\overrightarrow{h}^{(t)} &= \text{sigmoid}(\overrightarrow{L} x_t + \overrightarrow{H}\overrightarrow{h}^{(t-1)} + \overrightarrow{b_1});\\ \overleftarrow{h}^{(t)} &= \text{sigmoid}(\overleftarrow{L} x_t + \overleftarrow{H}\overleftarrow{h}^{(t-1)} + \overleftarrow{b_1});\\ \widehat{y}^{(t)} &= \text{softmax}(U_r \overrightarrow{h}^{(t)} + U_l \overleftarrow{h}^{(t)} + b_2).\end{aligned}$$

Intuitively, the tag of the current word not only has correlation with that of the proceeding word, but also that of the following word. Similarly, we approximate the gradient by only back-propagate two steps.

Finally, motivated by the improvement of tri-HMM from normal HMM, we propose a new model, named as trigram RNN. The model is illustrated in Figure 2. In particular,

$$\begin{aligned}h^{(t)} &= \text{sigmoid}(H_2 h^{(t-2)} + H_1 h^{(t-1)} + b - 1)\\ \widehat{y}^{(t)} &= \text{softmax}(U h^{(t)} + b_2)\end{aligned}$$

Here, $h^{(t)}$ not only depends on the previous hidden state $h^{(t-1)}$, but also directly depends on $h^{(t-2)}$. We hope that this extra dependency can help to catch longer windows in the sentence.

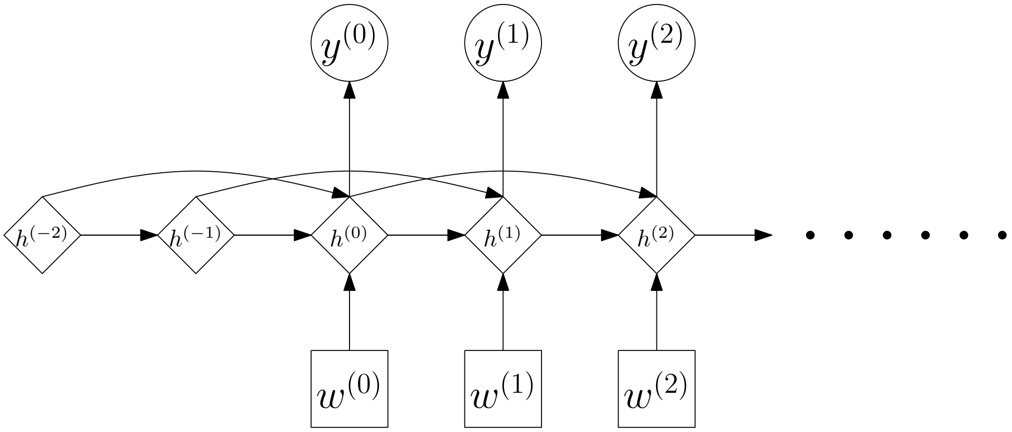

Figure 2: Trigram RNN

3.4 Other particularly designed techniques

There are also other tricks applied to the POS tagging in the literature. One of the most popular techniques is to add manually created features. However, we try to avoid human feature engineering in this project. Thus we do not follow this practice.

The only data preprocessing technique applied in this project is unsupervised word embedding. We use the skip-gram model for word embedding, and use the embedding result as the initial vector L for the neural network models. We use the structured skip-gram model of Ling et al. [2015] for the embedding. In the case of not doing this preprocessing step, a random matrix is generated as the initial value of L.

Another important question to consider is how to handle the unknown (and low frequent) words. An averaging method is proposed by Huang et al. [2007]. Due to the time limit, we instead randomly generate an embedding vector for the unknown word in the network models.

Other techniques are also proposed to achieve improvements in the literature, for example, reranking[Huang et al., 2007], latent annotations for tags[Huang and Harper, 2009], word clustering [Sun and Uszkoreit, 2012], using morphological structure to handle unknown words[Tseng et al., 2005]. These techniques are beyond the scope of this project, but we acknowledge that further improvements are possible based on these delicate ideas.

4 Experiments

We tested all the model mentioned above on a dataset of Chinese Buddhist texts.

4.1 Data statistics

We conducted a general exploration of the treebank to obtain the basic statistics of it. The treebank contains around 40K words and 8.5k sentences, drawn from four sutras in the Chinese Buddhist Canon. The total dictionary size is 3304. All the words are assigned one of the 30 tags, 3 of them being punctuations: ",", ".", "qm"(question mark). As expected, we observe a long tail distribution over the vocabulary. Among 3304 words, there are 1504 words only appear once, and 2412 words appear less than 5 times.

4.2 Measurement

The tree bank is separated into 3 parts in a ratio of 3:1:1, as training set, validation set, and test set. We mildly tune all the models, pick the best hyper-parameters based on its performance on the validation set. Finally we report its performance on the test set. All the performances are measured by classification accuracy. For the neural network model, we also report their F1 scores which take into account both classification precision and classification recall. (The F1 scores of MV and tri-HMM are omitted due to their weak performances.) To avoid overestimation, we removed all the punctuation words and tags. Note that tagging these punctuations is easy and should have much higher accuracy.

4.3 Experiment results

The final performances of all the models are reported in Table 1 and Table 2. Neural networks significantly outperforms the MV method and the HMM model. This demonstrates the capacity of neural network models. Among all the models, RNN_bidirect achieved the best performance, with classification accuracy 85.26% and F1 score 85.06%.

Among the neural network models, the 2-layers NN and the RNN-bidirect outperform the other two methods. In contrast to Modern Chinese data, it seems that modeling the sentence structure in a temporal perspective does NOT help in the Buddhist texts. One of the possible reason may be the small length of each sentence. Note that there are many short sentences in our treebank, each one consisting of less than 4 words. Average length of a sentence is less than 5. (compared to the Penn Chinese Treebank 6.0, average sentence length is around 25.) The relatively short sentence length could also explain the performance of the HMM model.

On the other hand, the bidirectional dependency (on the proceeding word and the following word) seems helpful. Note that in both models of 2-layers NN and RNN_bidirect, the following word is directly taken into account to generate the tag of the current word.

Although in the literature of Modern Chinese POS tagging, unsupervised word embedding is observed to improve the performances of the models. This effect is not significant in our experiments. With word embedding, the performances decreases in the RNN_bidirect model. We also tried to visualize the embedding result, but no clear pattern is observed.

	No embedding	skip-gram
MV	57.44%	NA
tri-HMM	42.75%	NA
2-layers NN	83.08%	83.77%
RNN	82.95%	83.18%
RNN_bidirect	**85.26%**	**84.96%**
tri-RNN	82.04%	82.57%

Table 1: Accuracies of different models

	No embedding	skip-gram
2-layers NN	82.97%	83.56%
RNN	82.78%	82.85%
RNN_bidirect	**85.06%**	**84.72%**
tri-RNN	81.87%	82.40%

Table 2: F1 scores of different models

5 Conclusion and future work

In this project, we take an initial step to investigate the performances of different models on the task of Chinese Buddhist texts POS tagging. A simple model of the temporal structure of the sentence seems not helpful in the task. On the other hand, significant improvement is observed by introducing direct bidirectional dependency.

Further improvements seem very likely and worth investigation, for example, by employing a better way of handling unknown words. Combining the ideas of trigram and bidirectional dependency in a RNN model may be another interesting investigation. Other further investigatios could be using different neural network models such as the convolution NN, since the temporal structure does not bring significant improvement in our experiment. Utilizing the tree structure of the treebank for the POS tagging task, for example, using recursive NN, could be another interesting topic. Different data augmentations techniques could also be helpful, given the small size of the dataset.

Acknowledgement

We sincerely thank Professor John Lee (Lee and Kong [2014]) for kindly providing us the Chinese Buddhist Text Treebank. Gratitudes also go to the author of the trigram HMM package, Guo [2013], and the authors of the wang2vec package, Ling et al. [2015].

References

Waleed Ammar, Chris Dyer, and Noah A Smith. Conditional random field autoencoders for unsupervised structured prediction. In *Advances in Neural Information Processing Systems*, pages 3311–3319, 2014.

Jesús Giménez and Lluis Marquez. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer, 2004.

Zach Guo. Hmm-trigram-tagger. 2013. URL `https://github.com/zachguo/HMM-Trigram-Tagger`.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. Incremental joint pos tagging and dependency parsing in chinese. In *IJCNLP*, pages 1216–1224. Citeseer, 2011.

Zhongqiang Huang and Mary Harper. Self-training pcfg grammars with latent annotations across languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 832–841. Association for Computational Linguistics, 2009.

Zhongqiang Huang, Mary P Harper, and Wen Wang. Mandarin part-of-speech tagging and discriminative reranking. In *EMNLP-CoNLL*, pages 1093–1102, 2007.

Shaoyu Jiang and Chirui Hu. *[Research papers on the grammar of Chinese Buddhist texts translations]*. The Commercial Press, 2013.

Julian Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242, 1992.

Lewis Lancaster. From text to image to analysis: Visualization of chinese buddhist canon. *Digital Humanities 2010*, page 184, 2010.

John Lee and Yin Hei Kong. A dependency treebank of chinese buddhist texts. *Literary and Linguistic Computing*, page fqu048, 2014.

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, and Wenliang Chen. Joint optimization for chinese pos tagging and dependency parsing. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(1):274–286, 2014.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. Two/too simple adaptations of word2vec for syntax problems. *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), Denver, CO*, 2015.

Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *EMNLP*, pages 277–284, 2004.

Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 777–784. Association for Computational Linguistics, 2008.

Chaofen Sun. *Chinese: A linguistic introduction.* Cambridge University Press, 2006.

Weiwei Sun and Hans Uszkoreit. Capturing paradigmatic and syntagmatic lexical relations: Towards accurate chinese part-of-speech tagging. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 242–252. Association for Computational Linguistics, 2012.

Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*, pages 32–39, 2005.

Zhiguo Wang and Nianwen Xue. Joint pos tagging and transition-based constituent parsing in chinese with non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 733–742, 2014.

Fei Xia. The part-of-speech tagging guidelines for the penn chinese treebank (3.0). 2000.

Kaixu Zhang, PR Fujian, Jinsong Su, and Changle Zhou. Regularized structured perceptron: A case study on chinese word segmentation, pos tagging and parsing. *EACL 2014*, page 164, 2014.

Jian Zhao and Xiao-long Wang. Chinese pos tagging based on maximum entropy model. In *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, volume 2, pages 601–605. IEEE, 2002.

Qingzhi Zhu. On some basic features of buddhist chinese. *Journal of the International Association of Buddhist Studies*, 31(1-2):485–504, 2010.