

Word Segmentation improvement through CRF, dictionary and Hebbian Learning Model

Tianhui Liu, Pochuan-Liang, Jiahuai Ma, Jiangan Cheng, Jingyi Xu

{Tianhui.Liu, pliang, maj2, JianganCheng, JingyiXu}@ufl.edu

Abstract

There are many differences between modern Mandarin processing and classic Chinese processing in different tasks, including segmentation, punctuation, quoting, named entity recognition and etc. In this experiment, we are trying to focus on the segmentation domain by applying a combined method of dictionary and an unsupervised learning model to improve the accuracy.

1 Introduction

Classical Chinese is a general call for the Chinese language used in the ancient dynasties through the past two millennia. Forms of classical Chinese literature include poems, Song Ci, articles, theses, couplets, etc. Classic literature is the primary cultural carrier for Ancient China, and the research on it is of significant meaning. Among the commonly applied tasks in processing Classical Chinese, segmentation is a basic but essential one. Due to the many differences existing between Modern Mandarin and Classical Chinese, it is not good enough to simply use a Chinese NLP model for Mandarin. In this experiment, the pre-trained model Guwen-Bert will be used. Using a combination of a dictionary for ontological analysis and unsupervised Hebbian learning, an improvement in the segmentation accuracy is expected.

2 Language analysis

Several problems arise when using modern Mandarin processing models for natural language processing tasks. Here are some of the major issues and their underlying reasons:

1. Ambiguity in Mandarin: In Mandarin, the pitch of a word can change its meaning. This makes Mandarin more ambiguous when doing tasks like speech recognition and machine translation. For example, the word "ma" can

mean "mother," "horse," "scold," or "numb" depending on different tones. This makes it difficult for modern Mandarin processing models to accurately recognize and translate speech.

2. Limited training data: The amount of Mandarin training data available for NLP tasks is limited compared to other languages. This makes it more challenging to develop accurate models for tasks like machine translation or sentiment analysis.
3. Complex grammar: Mandarin grammar is more complex than that of many other languages. For example, Mandarin does not have articles or plurals, and word order is more flexible. This can make it difficult for NLP models to accurately parse sentences and extract meaning.
4. Lack of context: Mandarin is a context-dependent language. The meaning of a word or phrase depends heavily on the context in which it is used. This makes it more challenging for NLP models to accurately understand and translate text, as they need to be able to accurately interpret the surrounding context.

In conclusion, while modern Mandarin processing models have come a long way in recent years, they still face several challenges. These challenges stem from the unique characteristics of the Mandarin language, such as tonality, grammar complexity, and lack of standardized romanization.

Most of the Chinese natural language processing tasks are based on word segmentation as a first step. There is one main problem in the word segmentation task. The first one is that the tokenizers in BERT family segment Chinese words in single characters by default. This is against the natural that Chinese words are potentially composed of



Figure 1: Supervised Sequence Labeling for CRF word segmentation training

multiple characters. This problem is the same in classical Chinese.

An example is "春秋." It can appear in a phrase as "春秋→ Spring and Autumn," and some state-of-the-art also would interpret this phrase as so. However, this idiom in most classical Chinese literature refers to a specific historical period. A counter-example could be "妻子." This word could appear as a single word composed of two characters, interpreted as "妻子→ wife," whereas in the earlier corpus, it is often interpreted as "妻子→ wife and son."

The problem of label bias problem on probabilistic models like HMMs and MEMMs will be also discussed in the next section.

3 CRF as a segmentation optimization method

CRF(Conditional Random Field) algorithm is a sequence labeling algorithm based, receiving an input of size n $X = \{x_1, x_2, \dots, x_n\}$ and giving a labeled output of size n text $Y = \{y_1, y_2, \dots, y_n\}$. In other words, CRF is modeling $P(Y|X)$. In this task of Chinese word segmentation, there are two kinds of labels, 'B' for the beginning of a word and 'I' for the inside of a word. By manually labeling the word segmentation on the training corpus, input a test sequence and determine the labeling sequence of the output. The idea of using a CRF(Conditional Random Field) model for segmenting and labeling sequence data was raised by Lafferty[1]. In this work, Lafferty argued that the label bias problem makes all traditional probabilistic models victims. This is precisely the problem discussed in the previous paragraph. The researchers attempted to combine the advantage of conditional models with the global normalization of random field models. The first attempt by Peng[4] on applying this model to Chinese language word segmentation tasks. Researchers did a close-and-open CRF test and com-

Closed				
	Precision	Recall	F1	R_{out}
CTB	0.828	0.870	0.849	0.550
PK	0.935	0.947	0.941	0.660
HK	0.917	0.940	0.928	0.531
AS	0.950	0.962	0.956	0.292
Open				
	Precision	Recall	F1	R_{out}
CTB	0.889	0.898	0.894	0.619
PK	0.941	0.952	0.946	0.676
HK	0.944	0.948	0.946	0.629
AS	0.953	0.961	0.957	0.403

Figure 2: Overall results of CRF segmentation on closed and open tests according to Peng[4]

pared the result on four different systems. There is a clear improvement in the open test set in different evaluation standards. Chang's work also supported evidence that CRF helps improve modern Mandarin word segmentation accuracy. [3].

4 Assumption of Using Hebb Learning to Improve Segmentation

In this phase, all the work is based on the foundation that the segmentation is already done by applying CRF. CRF uses a supervised sequence to predict the labeling of its target text sequence. This decision-making of CRF is based on probability. This probability is defined as a global probability in this project and is achieved by training using the related corpus.

Aside from this global probability, a local adjustment can be made to the probability by analyzing the context of the literature.

After running the CRF, the global probability of the task $P(Y|X)$ can be extracted; To be exact, the probability $P(y_i = 0|X, 0 < i < n)$ is computed, where to say $y_i = 0$ is the same as at position i , the CRF algorithm would label it as B(Beginning label). Here is an actual example as "闭门羹." Say '闭' is at position 0. Based on this paper [2], we model local decision-making as shown in the figure 3. The activations L is calculated by $Pr(L_a|y_0 = 0) = b_a + \sum_{i=1}^n w_{a,i} \cdot y_i$, where $w_{a,i}$ is calculated with the probability of y_i , e.g. $L_2 = Pr(y_0 = 0, y_1 = 1, y_2 = 1)$. A Set of potential stimulates $X = \{x_1, \dots, x_m\}$ is based on the potential word segmentation of the context and a $M \times N$ matrix A of actions $a_{j,k}$. An inspection on x_j and y_k , if there is a positive connec-

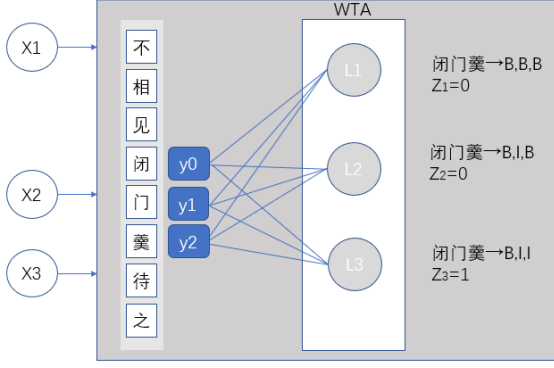


Figure 3: A base model of local Hebb decision machine

tion, for example, word x_j and word y_k are both idioms in sinicized Buddhism literature, reward $r(a_{j,k}) = 1$. The model uses a reward-modulated Bayesian Hebb rule to do weight updates by attempting to maximize the sum of reward the decision machine get. When the learner is trying to decide on three actions, a, b, and c, then the selection is made by $z_a = 1 \iff L_a = \max\{L_a, L_b, L_c\}$. In this project, we will try a naive way of Hebb Learning, which is ignoring the changes happening due to the change in word segmentation progress.

An alternative weight operation strategy is to generate the probability between two segmented words, this work has been done by Yulin Li and other researchers[7].

5 Methodology

The major difficulty in Classical Chinese is that the tokenizer of either RoBERTa or BERT treats single Chinese characters as tokens. As in Classical Chinese, it could be hard to determine whether two characters are one token, even if these two characters can form a word in a dictionary. So the attempt we want to try is to use CRF to get possible combinations of characters as potential tokens, do ontology analysis, form connection objects, and use them to adjust the initialized weight, try to achieve masking, and improve NER task.

In this experiment, we will use the original tokenizer of GuwenBert, train GuwenBert as the control group, and get a score using CClue evaluation for performance comparison. We will use the BERT tokenizer, try the Hebbian learning way of fine-tuning, then get a CClue score. If there is an improvement for most of the test set, this experiment is successful.

As for implementing the dictionary, to validate its effectiveness, we will just input the necessary info

for the test set.

For the dataset, we will first use the token the generate from the tokenizer using our dataset and input to the model to get an output, then we change some parameters of the token then input to the model to get another output, we are going to compare both output to see if our method is better than the original one.

6 Base Matrix

In this section, we will introduce the pre-trained model we selected and the environment.

For this project, we used the pre-trained model GuwenBERT[6] by Ethan Yan, which is trained with cleansed corpus and achieved a 0.89 F1-Score. This work of Yan, is based on the tokenizer of BERT. We plan to use Daizhige Version 20, a classical Chinese corpus with categorized literature for the data set. The pre-trained base model as a comparison set is invoked through Hugging Face.

7 Dataset

The dataset "daizhige20" was collected from GitHub. This is a collection of classical Chinese poems written during the Tang and Song Dynasties and is maintained by Gary Ge, a researcher at Carnegie Mellon University. The "daizhige20" dataset consists of text files of over 20,000 poems written by more than 2,000 poets. Each file in this dataset contains the poems of a single poet. These poems cover a wide range of themes, including nature, love, war, and philosophy. The poems are written in classical Chinese characters, which may require knowledge of the language to interpret and understand. Overall, the "daizhige20" dataset provides a rich source of information for researchers interested in studying classical Chinese literature and poetry. It could be useful for applications such as natural language processing, text mining, and sentiment analysis.

For pre-processing of the dataset, because what we need for input is only plain text, so we have to remove extra white space in the file, also some topics or chapters that write in the file. For those content with Punctuation, we are going to remove all the punctuation, but the version with punctuation can use to check the correctness of the output while training. Figure 4 is an example of manual labeling for word segmentation.

We will separate the dataset into two parts; one is for training, which includes the training data

1	天	B-char
2	下	I-char
3	君	B-char
4	王	I-char
5	至	B-char
6	于	B-char
7	贤	B-char
8	人	I-char
9	众	B-char
10	矣	B-char
11	当	B-char
12	时	B-char
13	则	B-char
14	荣	B-char
15	没	B-char
16	则	B-char
17	已	B-char
18	焉	B-char
19	孔	B-char
20	子	I-char
21	布	B-char
22	衣	I-char
23	传	B-char
24	十	B-char
25	个	I-char

Figure 4: Manually labeled segmentation example

and validation date, and the other is for testing data, which is used for testing the accuracy of our model.

8 Evaluation

We have to know whether the text translated by our model is accurate or not, and we also need to compare our model horizontally for completeness in text translation and punctuation recognition, which are the conditions to measure the readiness of a model. So we found a tool called CClue, which gives us a score for our model, a score about the accuracy and readability of our model.

CClue is an open source natural language processing tools that can be used to get a grade on various models which is used on translation.

CClue can evaluate the accuracy of the translation, we need to submit the unit (individual / team name), model name, project / paper address, and link to the model weights (upload to Hugging Face for documentation). Afterwards an evaluation result about our model will be obtained.

CClue Hugging Face employs five methodologies and standards to assess the performance of NLP models:

1. Loss Function: The loss function determines

how well the model performs when it is applied to the training data. A lower loss value indicates better performance on the training data.

2. Accuracy: Accuracy is the number of correctly predicted samples divided by the total number of samples. This is a common evaluation metric for classification problems.
3. F1 Score: The F1 score is calculated by taking the harmonic mean of the Precision score and the Recall score. This metric is more representative than accuracy in the case of imbalanced data.
4. BLEU Score: The BLEU Score is a statistic that is used for the assessment of machine translation, comparing the similarity between the translations by model and human translations.
5. ROUGE Score: ROUGE Score is an assessment measure that is utilized for the tasks that are associated with automated summarization, measuring the similarity between the summaries by model and human summaries.

The differences between CClue assessment and GT assessment:

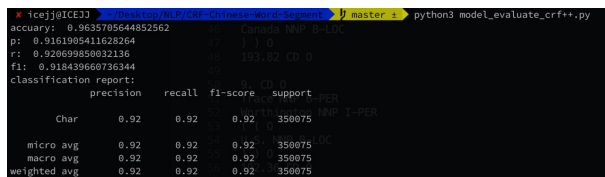
1. Both use BLEU and ROUGE, which are evaluation methods used to assess the similarity between machine translation and human summaries.
2. CClue uses loss function and Accuracy, which indicate that a lower loss value and higher accuracy means that your model performs better on the training data.
3. GT uses TER and TERP, which is a way to show Translton Edit Rate.
 - (i) TER (Translation Edit Rate): TER calculates the minimum number of edits required to convert the machine-translated text into the reference translation and normalizes it.
 - (ii) TERP (Translation Edit Rate Plus): TERP is an extension of TER that takes into account synonyms of words and reordering of phrases. TERP enables the substitution of a word with its synonym when calculating the number of revisions and enables the re-ordering of phrases without compromising

the quality of the overall translation. This makes TERP more reflective of human translator evaluations, as it allows for vocabulary and structure variation.

For evaluating the results, both BLEU and ROUGE are useful to. The reason for choosing CClue is that TER and TERP are not suitable in Classic Chinese translations, because there is no TER and TERP evaluation for translating ancient texts into modern English, when we translation form Classic Chinese to English, we do not have a standard reference translation. So we choose CClue Hugging Face.

9 Result

Result The first phase experiment is carried out by applying a dictionary with CRF, and here is our experiment result. This proves that CRF combined



```

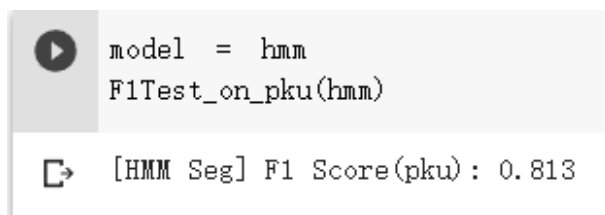
accuracy: 0.9635785644852562
p: 0.9161985411628264
r: 0.920699858032136
f1: 0.918439660736344
classification report:
      precision    recall  f1-score   support

 Char           0.92         0.92         0.92         350075
 micro avg       0.92         0.92         0.92         350075
 macro avg       0.92         0.92         0.92         350075
 weighted avg     0.92         0.92         0.92         350075

```

with a dictionary has improved the accuracy of classical Chinese segmentation tasks to a reasonable level. Using CRF to implement Chinese Segmentation, at this stage, we uses 86905 sentences of data to train our model. It iterate 919 times and the labeling error rate is 0.01816, the sentence error rate is 0.42424.

We use sequeval metrics to evaluate the model with test data, We get the accuracy of 0.935, this is the figure of the result. On the other hand, a HMM model trained with same dataset for the word segmentation task has a worse result. This result of HMM F1 score is achieved by using the published work of Wu[5].



```

model = hmm
F1Test_on_pku(hmm)

[HMM Seg] F1 Score(pku): 0.813

```

10 Limitations and future suggestions

Limitations of this project include as follows: First, during the progress of Hebb Learning, we didn't

do an update on the X vector, that is, during the progress, as local decisions are made, some elements of X vector stimulations shall be disabled and this shall also affect the decision making. Second, since we don't have resources for manually labeling a pure classical Chinese dataset, we used a dataset that includes Modern Mandarin and also classical Chinese. This may make our result not so accurate. Also, there are many kinds of CRFs, like LSTM-CRF and Bi-LSTM-CRF. A comparison shall be made between them.

11 Conclusion

Chinese word segmentation with CRF has clearly higher accuracy than those done with HMM or MEMM. And as Hebb learning always create a positive increment amount, it shall be able to increase the probability to the wanted direction.

References

- [1] A. McCallum Lafferty, John and F. Pereira. 2001. Conditional random field: Probabilistic models for segmenting and labeling sequence data. *ICML 18*.
- [2] Rodney J. Douglas Wolfgang Maass Michael Pfeiffer, Bernhard Nessler. 2010. [Reward-modulated hebbian learning of decision making](#). *Neural Computation* (2010) 22 (6): 1399–1444.
- [3] CD Manning PC Chang, M Galley. 2008. Optimizing chinese word segmentation for machine translation performance.
- [4] Fangfang Feng Peng, Fuchun and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. *COLING 2004*.
- [5] Zack Wu. 2017. [Chinesewordsegmentation-system](https://github.com/izackwu/ChineseWordSegmentationSystem). <https://github.com/izackwu/ChineseWordSegmentationSystem>.
- [6] Ethan Yan. 2021. Guwenbert. <https://github.com/Ethan-yt/guwenbert>.
- [7] Fanyu Wang Yulin Li, Zhenping Xie. 2022. [An associative knowledge network model for interpretable semantic representation of noun context](#).