A Chinese Input Edit Method (IEM) based on HMM

Sijie Ju

1. Introduction

This project aims to create an Input Edit Method (IEM) for Chinese. In Exercise 2, one of the shortcomings of my Chinese name generator was the lack of a reasonable solution for converting pinyin to Chinese characters. This inspired me to work on an IEM for Chinese. The mainstream method to build a Chinese IEM is based on Hidden Markov Models (HMM) and the Viterbi algorithm, which are also used in Part-of-Speech (POS) tagging. This attracted my interest, so in this project, I am going to train a model . The quality of this model will be tested on different corpus texts. I expect its performance to be better on texts from the same genre due to similar writing styles and vocabulary used. What's more, its accuracy will decrease as the texts get longer because the generation of the next hanzi is affected by the selection of previous hanzi. If a word or phrase is not covered by the training data, the model may generate the wrong hanzi for the pinyin. A longer sequence may aggravate this problem.

2. Material and methods

The process of converting pinyin into hanzi can be understood as finding the most possible hidden sequence (hanzi sequence) based on the given observable sequence (pinyin sequence). The hanzi sequence is predictable due to the limited number of hanzi that typically appear after a certain hanzi. The transmission probability is the probability of the appearance of a hanzi after the previous one. Additionaly, the pronunciation of hanzi is conventional, so the emission probability can be calculated by the frequency of hanzi corresponding to a given pinyin. Given these factors, HMM proves to be a good approach for converting pinyin into hanzi.

For the training data, I used a word dictionary (mapull, 2023) as a basic vocabulary, and a news corpus (brightmart, 2023) to improve the coherence of predicted sequence. The reports in the news corpus covers a wide range of topics, including daily life, entertainment, politics, and economics, which means it can provide a rich resource of vocabulary. The dictionary has totally 320349 words, while the news corpus used for training has 576635 short sentences. To ensure the dataset contains only Chinese characters, all data was filtered using Unicode.

Following this, I counted the initial probability, transmission probability and emission probability. The frequency was firstly counted and then was normalized by using logarithms to obtain the probability. Then I collected all the hanzi with same pronunciation (pinyin) into a dictionary. With these parameters, I developed a function utilizing the Viterbi algorithm to decode the pinyin sequence. The model was tested on two corpora of different genres. The first dataset was constructed with the news corpus used in the training. I choose the next 100 documents in the corpus, which includes 6406 pinyin sequences. The second dataset was derived from a Chinese wiki corpus (brightmart, 2023). The first 30 documents with a total of 6,585 pinyin sequences was used for the dataset. compare the model's performance across these different text types. The wiki corpus contains more technical terms, so I expected it to be more difficult for the model and lead to a lower performance. For the evaluation, I compared the

predicted texts and the original texts and calculate the accuracy to see to which extent the model operated correctly.

3. Results

According to the evaluation results, the model accuracy on the news corpus is 80.773%, while its accuracy on wiki corpus is 80.579%. The IEM in real word can provide multiple options for the users to choose from. However, the predicted sequence from our model is just the most possible hanzi sequence for the pinyin input. Thus, the model performance is good according to the figures. The model's performance across different genre is very similar, which contradicts with my expectation.

As is shown in the Figure 1, the texts in wiki corpus are mainly combined with 2 to 20 characters, while the texts in news corpus is shorter. Its length mainly focused on 5 to 10 characters. In both corpora, it's rare to see texts with more than 35 characters. Thus, these texts can only provide limited insight into the relationship between length and accuracy. In terms of the influence of length on the precision of prediction, the result (Figure 2) indicates no clear correlation between the length of pinyin sequence and the precision. The accuracy of the predicted sequences is all around 80%, which also diverges from my expectation.

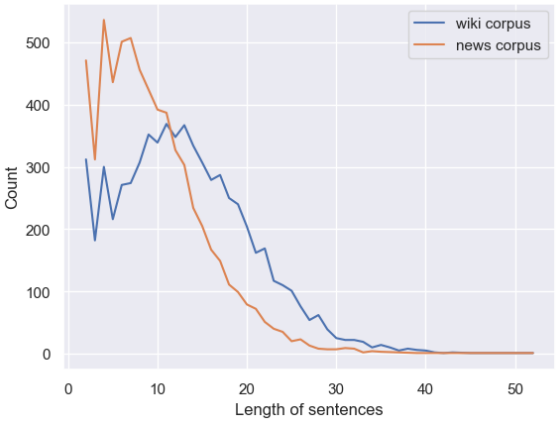

Figure 1 Length of texts in corpus

Figure 2 Accuracy of predictions with different length

This may be attributed to the fact that most of the words in the test texts are known to the model. With Viterbi algorithm, the model can minimize the number of errors, thereby enhancing its overall performance.

The present IEM only has the most basic function: convert the pinyin to hanzi. For a more powerful IEM, more functions should be added. Firstly, in practical situation, the input pinyin sequence is a complete string and should be parsed and segmented by the IEM. However, our model can only convert the segmented pinyin list into hanzi. What's more, our current model lacks the ability to detect and correct errors in the pinyin sequence. It would break down if the input pinyin sequence contains pinyin that is not included in the pinyin list. Hence, it would be a necessity to equip the model with the function of automatic error-correction .

References

brightmart, (2023). *NLP Chinese Corpus*. https://github.com/brightmart/nlp_chinese_corpus

iseesaw, 2023. *Pinyin2ChineseChar.* https://github.com/iseesaw/Pinyin2ChineseChars

letiantian, 2015. *Pinyin2Hanzi*. https://github.com/letiantian/Pinyin2Hanzi?tab=readme-ov-file

mapull, zeffon (2023). *Chinese Dictionary*. https://github.com/mapull/chinese-dictionary/blob/main/word/word.json

The code can be viewed from:
https://github.com/Kamemii/A-Chinese-Input-Edit-Method/tree/main