Transliteration Project

REPORT_lhe_0619

Introduction

The current work I have done can be divided into four parts, i.e. reading papers, learning deeplearning.ai courses, collecting data, and building the baseline for the model of the English-to-Chinese transliteration.

Papers

I mainly focused on the sequence-to-sequence paper from Google, but I could not understand some concepts such as the bidirectional encoder. Also, this paper is rather short and hides many details. What I am currently interested is its discussion and description of the transliteration dataset (which will be discussed below), as it reveals some problems that I am also concerned while gathering my first dataset. Since I was trying to run the Google's model and adjusting my data, I have not finished the reading of the paper concerning the Monolingual Corpora, I will study it after the meeting on Wednesday.

Courses

The Coursera website provides me with great materials to learn Machine Learning. I currently finished the first simple project of the binary classification using logistic regression, it is implemented on Jupyter/iPython Notebook. Since two month is a short period of time, I will accelerate the learning process, hopefully will know LSTM/GNN before building the final model.

Dataset

The dataset I have created has about 6800 word pairs, including people's names from different origins (particularly there is a small list of the ancient Greek Gods such as Zeus),

places' names from various countries (cities in America/Britain/New Zealand take up the majority), and a list of brands and borrowed words.

Problems with the dataset

As mentioned in the Seq2Seq paper, there are many exceptional words whose transliteration probably cannot be learnt at all. Although currently I can erased these exceptions by checking the data line by line, such noisiness cannot be resolved manually when the dataset becomes larger and larger. In my observation, the exceptions occur a lot in the names of places, brand or simply by convention. The reasons behind may result from historical factors or literary manipulation of words. For example, Paris is transliterated to 巴黎, but it actually sounds more like 帕里斯. Nevertheless, 巴黎 has a sense of more 'beauty" than 帕里斯 in Chinese. My idea is that should we divide the dataset into names of peoples and names of others, because exceptions are much less frequent in people's names.

The second issue is about the general noisiness among the places' names. There are dozens of names containing certain parts irrelevant to transliteration, such as 'River', 'Port', 'Mount', 'Sea', 'Island', 'North/East/West/South' , etc., which correspond to '河', '港', '山', '海', '岛, '北/东/西/南' in Chinese. Even though I can erase the irrelevant parts by seeking for some patterns, it will result in a 'collateral damage' in the sense that some words containing these seemingly irrelevant parts as sound components will lose them accidentally. For instance, the English name 'Westbrook' contains 'West', but in this case 'West' does not represent a direction. Instead, 'West' will be transliterated to '韦斯特'. Again, for a small dataset, I can resolve all the conflicts by checking it line by line. But it is impossible for a giant dataset.

The final concern is that Chinese characters are pictograms. There is no way to infer the sound by simply looking at the character itself. Also, Chinese is a many-to-many language in the sense that a single Chinese character may have several sounds and a particular sound may be possessed by several characters, whereas the sound of a single English word is almost fixed (although there exists different accents). Consequently, a model simply dealing with the textual data may treat alternative and other valid transcriptions as errors. (Mihaela Rosca et al., 2016)

Baseline

```
ERROR 2470000 0.374696     1078 2877
```

In the first round, I used 7011 word pairs to train the seq2seq model and obtained an error rate of 37.4696% for the training. However, the model behaves rather badly on the test data.

I am now re-training the model using a refined dataset (By refined I mean that I erase some exceptions manually) of the size 6800+. Hopefully it will be finished by tomorrow's meeting.