CS221 Project Proposal

Julia Gong, Benjamin Newman

October 2018

1 Introduction and Problem Definition

Our proposal is to create a system that, given an English or anglicized name rendered in Latin letters (particularly focusing on traditionally American names), produces the approximate phonetic Chinese transcription of the name in Chinese characters. This is a machine transcription problem that involves three key steps, two of which will be our primary focus.

1. **Segmentation**: This step involves breaking the English name into discrete parts whose sounds likely exist or can be approximated in individual Chinese characters (e.g. Jake → Ja | ke).

2. **Transcription**: This step consists of choosing the best Chinese pinyin (Chinese phonetics written in Latin characters with tones) that best match the segmented words (e.g. Ja | ke → jie | ke → jié | kè).

   This task can be modeled in a number of ways. We will plan to focus on two.

   - We can formulate a search problem, with the costs being a variant of the edit distance between the English and pinyin strings, as well as a bigram cost of the co-occurrence likelihoods of consecutive pinyin transcriptions.
   - Model a sequence matching problem and use a deep neural model with the English segments as inputs.

3. **Characters**: The final step is choosing the Chinese characters that best reflect the chosen pinyin (e.g. jie | ke → jié | kè → 杰克). This may depend on factors external to language, such as gender of the name or character meanings. (Of course, there is some interaction between this step and the previous one - if there is a better character with a slightly different pinyin, it might be preferred despite poorer edit distance.)

   The character matching component will not be the focus of this project unless we have enough time; instead, we will initially use a lookup table with standard pinyin to character mappings rather than learning these associations from scratch.

The following depicts an example of what this process might look like in full:

Alice → A | LI | CE → ai | li | si → ài | lì | sī → 爱丽丝

2 Literature Review

In many ways, the task we are trying to accomplish is analogous to the grapheme to phoneme (g2p) task. In this task, given a group of written words (graphemes), the goal is to predict how the words sound (their phonemes). This is a relatively simple task for relatively phonetic languages like Spanish, somewhat more difficult for languages like English, and almost impossible for languages like Chinese. This g2p task is most prevalent in text-to-speech software used in accessibility readers or in personal assistant technologies like Siri or Alexa. There have been a variety of methods proposed for solving these kinds of problems: linear classifiers such as support vector machines, finite state automata-based methods such as conditional random fields and weighted finite state transducers, deep learning models like RNNs and LSTMs, and the popular joint-sequence model and hidden markov model (both of which are Bayesian approaches) [3, 4].

Our task is slightly different because, instead of abstracting phonemes into a universal phonetic script (e.g. the International Phonetic Alphabet), we must limit ourselves to phonemes in Chinese. Others have attempted similar versions of this problem. Wan and Verspoor use a rules-based method with modest success [5]. Shao et

al. use a modified version of the M2M aligner to segment the English words [6]. Upadhyay et al. have had good success using attention-based RNN methods with low resource languages, a simpler version of which we may try to adopt [7].

3 Data

We have collected data from two places. Some English names have officially agreed upon Chinese transcriptions, and we have obtained approximately 1500 of those transcriptions for male and female names online [1]. The rest of our data consists of ∼13500 American names, also found online. They are taken from public records and consist of names of people of multiple races (White, Black, Indian, and Hispanic) and with names that are traditionally male or female. (Unfortunately, the dataset is heavily biased toward male names (∼11000) as opposed to female names (∼2500). This should not matter too much for the pinyin task, but might affect the outcome of a later character assignment task, if we end up pursuing it).

4 Metric

We use two modified edit distances as our metrics. First, we find that, when dealing with shorter letter sequences (on the word segment level), normal edit distance (i.e. the minimum number of insertions/deletions to transform one word into another) is not enough to distinguish between word segments, so we also subtract the number of shared unique characters. This is used to come up with our baseline (see more below).

Second, for assessing our oracles and comparing them to the baseline, we use a different modified edit distance metric that penalizes different tones on the same vowels half as much as different characters. We refer to this modified pinyin distance metric as the 'distance metric' below.

5 Oracle and Baseline

5.1 Oracle

For our oracle, we asked two native Chinese speakers to independently transcribe a list of 30 anglicized (e.g. 'Golrokh') or English (e.g. 'Carly') names to the best of their knowledge. Of the 30 names, 12 names differed between the two oracles, and the average distance metric between the oracles' answers was only 0.8333. The oracles were very consistent with one another for the majority of the names, and the distance metric reflects how minute the differences were when they arose (often being merely a tone difference). Names such as 'Alice' or 'Jake', which have known standard transcriptions, were all correctly identified by both oracles.

5.2 Baseline

Our baseline algorithm uses the following rules-based method to transcribe a name. An English string can be segmented into either a consonant (C), a vowel (V), a consonant followed by vowel (CV), or a consonant followed by a vowel and another consonant (CVC). CVC is always preferred over CV-C when the character after the second consonant is another consonant (i.e. CVCV will be parsed as CV-CV, while CVCC will be parsed as CVC-C). When neither the CV nor the CVC rules apply, the default value of the lone character is used.

The lookup table is a standard transcription spreadsheet where rows contain vowel or word-ending categories corresponding to the International Phonetic Alphabet (IPA) (e.g. AA, UW, AEN, IHNG) and columns contain consonant or word-beginning categories in IPA (e.g. K, R, ZH, SH). Each column-row intersection contains a common Chinese character corresponding to the combination of these sounds (including cases where only a vowel or a consonant is provided). The algorithm converts the CV, CVC, C, or V segment from the original English string into the corresponding IPA column and row that has the lowest distance metric. For example, the 'LI' in 'ALICE' has the lowest distance metric to 'LIY', which is the chosen lookup in the table that yields '利'.

Of the 30 names assessed by the oracles, none of the baseline transcriptions perfectly matched the oracles' answers. The average distance metric (averaged for each word between the distance to each oracle) was 3.4417. Compared to the distance metric of 0.8333 between the two oracles, this shows that there is much potential for algorithm improvement and development of a better transcription system from English to a segmented pinyin string.

6 References

[1] Mack, L. (n.d.). Ever Wonder What Your Name Translates to in Chinese? Retrieved from https://www.thoughtco.com/chinese-and-english-names-688196

[2] Bejda, M. (n.d.). List of Datasets. Retrieved from http://mbejda.github.io

[3] Thu, Y. K., Pa, W.P., Sagisaka, Y., and Iwahashi, N. (2016). Comparison of Grapheme–to–Phoneme Conversion Methods on a Myanmar Pronunciation Dictionary. *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing*. 11–22. Retrieved from http://www.aclweb.org/anthology/W16-3702

[4] Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using Long Short-Term Memory recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4225-4229. doi: 10.1109/ICASSP.2015.7178767

[5] Wan, S., Verspoor, C. M. (1998). Automatic english-chinese name transliteration for development of multilingual resources. In *COLING-ACL* (pp. 1352-1356). Retrieved from https://pdfs.semanticscholar.org/ecdc/66be284fd8d03d5325c403635ea2c8dc0bc6.pdf

[6] Shao, Y., Tiedemann, J., Nivre, J. (2015). Boosting English-Chinese Machine Transliteration via High Quality Alignment and Multilingual Resources. In *Proceedings of the Fifth Named Entity Workshop* (pp. 56–60). Association for Computational Linguistics. Retrieved from http://anthology.aclweb.org/W/W15/W15-39.pdf#page=66

[7] Upadhyay, S., Kodner, J., Roth, D. (2018). Bootstrapping transliteration with constrained discovery for low-resource languages. Retrieved from https://arxiv.org/pdf/1809.07807.pdf