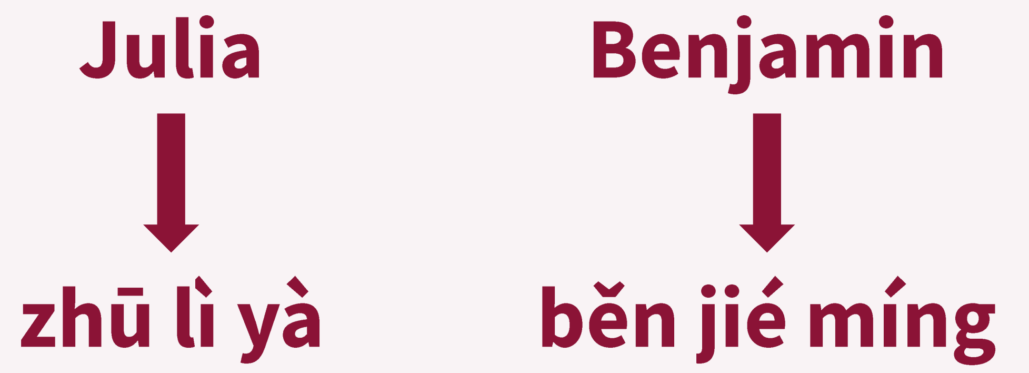
English-Chinese Name Machine Transliteration using Search and Neural Network Models

Julia Gong, Benjamin Newman

Motivation and Objective

- Machine translation focuses on meaning rather than sound, thus not directly addressing the transliteration of names between languages, especially those without standard transliterations.
- **Objective**: Finding the optimal transliterations of English names into Chinese pinyin. "Optimal" entails:
  - Sounding closest to original English,
  - Being consistent with existing standard translations (e.g. examples below).
- Translation attempted directly, without intermediate phoneme representations.



Julia → **zhū lì yà**

Benjamin → **běn jié míng**

Data

- List of English names, including both those traditionally male and traditionally female, collected from online source [1].
- Created dataset of 1510 names with:
  - English name,
  - Chinese characters of transliteration,
  - Pinyin of characters (processed from pinyin provided by Google Translate),
  - Traditional gender of name.
- Data pre-processing:
  - Remove repeated names,
  - Space-separate individual character syllables in generated pinyin using Glosbe transliteration API [2].

Example: Alice | 爱丽丝 | ài lì sī | f

I. Search Model

- To formulate the search problem, we generate all possible segmentations of an English name (set S) and iterate over S to find the minimal cost pinyin.
- The best pinyin transliteration p for a given segment $s \in S$ minimizes the cost function:

$$c(s,p) = editDistance(s,p) * syllableCost(p)$$

- *editDistance*: the minimum number of substitutions, deletions, and additions needed to turn the segment into the pinyin, with only a 0.5 penalty for wrong tones
- *syllableCost* is the surprisal of the syllable, $-\log(\hat{p})$, where $\hat{p}$ is the maximum likelihood estimate (within relevant names) of the pinyin syllable in the corpus

The cost of the entire name segmentation is then

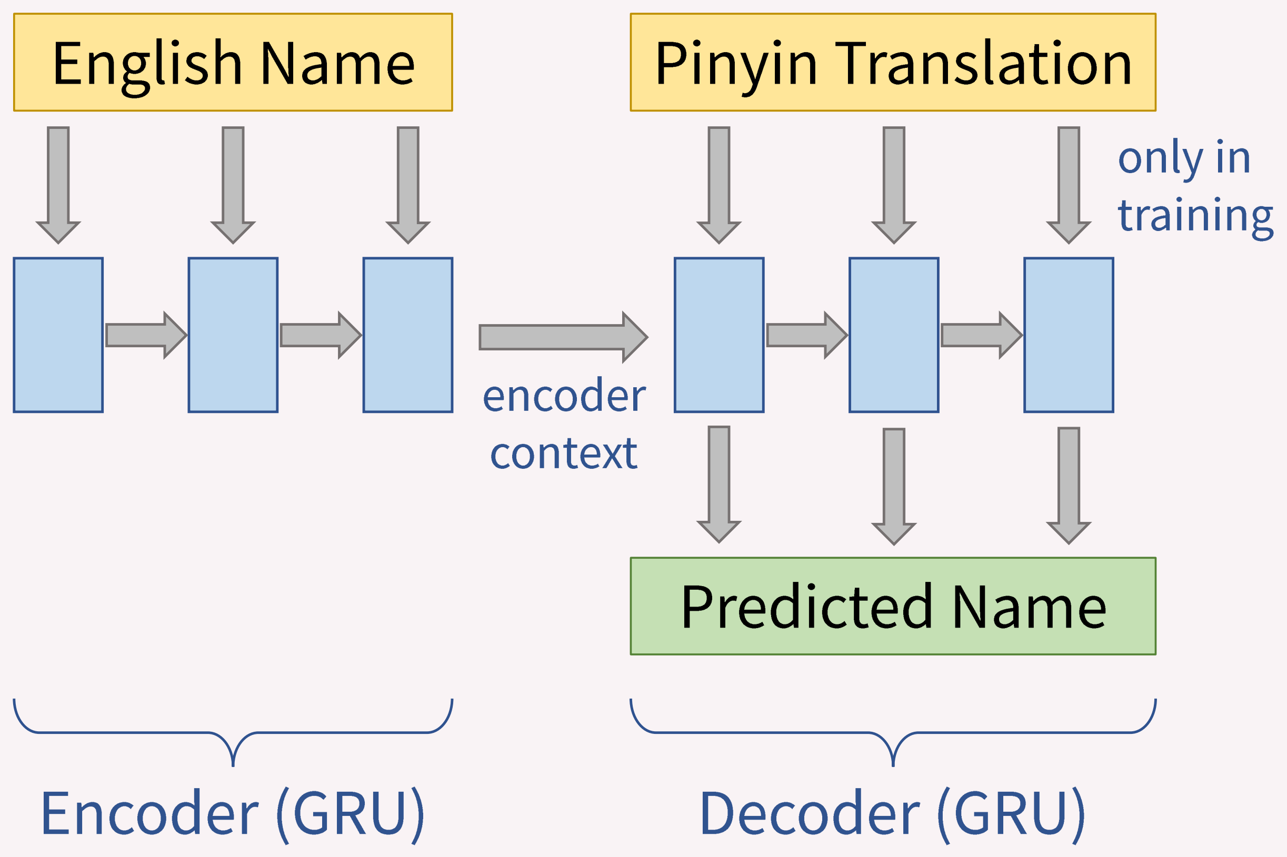
$$C(S) = \sum_{s \in S} \min_{p} c(s,p)$$

- Dynamic programming is used to extract the pinyin with the minimum cost segmentation.

II. Neural Network Model

- We use the encoder-decoder architecture to relate sequential features of English names with pinyin transliterations, with a 1410/100 train-test split.
- For the classification task, we use cross-entropy loss:

$$\mathcal{L}(\hat{y}, y) = -\mathrm{y} \cdot \log(\hat{y})$$



Results, Analysis, Discussion, Improvements

Search Model: 1488 of 1510 names differed from the ground truth, with average *editDistance* of 4.36 among differing pairs, such as:

- Benjamin, běn jié míng → běn jiā mǐn
- Julia, zhū lì yà → bù lián
- Erasmo, āi lā sī mò → ér ā sà mò

Ongoing improvements: Address issue that optimal pinyin may not be written similarly to corresponding English (e.g. *j*, *zh*; *er*, *ai* above), and may require phonetic mapping. Refine *syllableCost* with n-grams-based estimate of pinyin likelihood.

Neural Network Model: 98 of 100 test set names differed from the ground truth, with average *editDistance* of 4.13 characters among differing pairs, such as:

- Benjamin, běn jié míng → pěn méi ěr
- Julia, zhū lì yà → jié ěr u
- Henry, hēng lì → hǎn nn (illustrates that some generated character sequences were not even valid pinyin)
- Claudia, kè láo dí yà → kè lì kè (the network did learn to segment syllables using spaces, but still inserted double spaces for some names)

Ongoing improvements: predict syllables instead of characters in decoder, enforce structure constraints for valid pinyin form, combine search and neural network where the output pinyin of the search is fed as input into the neural network.

Takeaways: Overall, the search model was superior. It had slightly worse edit distance, but edit distance alone doesn't completely quantify transliteration quality, since the search model was guaranteed to choose valid pinyin. The neural network performed worse on average, committing fundamental mistakes likely due to the small dataset size, which could improve when we combine the two methods.

Challenges

- Optimal pinyin do not always align with English syllable boundaries.
- We have insufficient data to use large-scale deep learning techniques, and it is very difficult to generalize to uncommon character sequences.
- We can't assume the best pinyin that sounds most similar to the English is written similarly to English. Search models need encoded relationships between sounds and symbols.

References

[1] Mack, L. (n.d.). Ever Wonder What Your Name Translates to in Chinese?

[2] GLOSBE Partners. (2017). https://glosbe.com/

[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.

[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).

[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.

[7] Upadhyay, S., Kodner, J., & Roth, D. (2018). Bootstrapping transliteration with constrained discovery for low-resource languages.