How to Speak a Language without Knowing It

Xing Shi and Kevin Knight
Information Sciences Institute
Computer Science Department
University of Southern California
{xingshi, knight}@isi.edu

Heng Ji
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
jih@rpi.edu

Abstract

We develop a system that lets people overcome language barriers by letting them speak a language they do not know. Our system accepts text entered by a user, translates the text, then converts the translation into a phonetic spelling in the user's own orthography. We trained the system on phonetic spellings in travel phrasebooks.

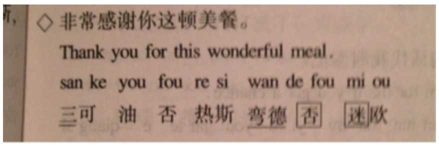
1 Introduction

Can people speak a language they don't know? Actually, it happens frequently. Travel phrasebooks contain phrases in the speaker's language (e.g., "thank you") paired with foreign-language translations (e.g., " спасибо"). Since the speaker may not be able to pronounce the foreign-language orthography, phrasebooks additionally provide phonetic spellings that approximate the sounds of the foreign phrase. These spellings employ the familiar writing system and sounds of the speaker's language. Here is a sample entry from a French phrasebook for English speakers:

English:	Leave me alone.
French:	Laissez-moi tranquille.
Franglish:	Less-ay mwah trahn-KEEL.

The user ignores the French and goes straight to the Franglish. If the Franglish is well designed, an English speaker can pronounce it and be understood by a French listener.

Figure 1 shows a sample entry from another book—an English phrasebook for Chinese speakers. If a Chinese speaker wants to say "非常感谢你这顿美餐", she need only read off the Chinglish "三可 油 否 热斯 弯德否 米欧", which approximates the sounds of "Thank you for this wonderful meal" using Chinese characters.



◇ 非常感谢你这顿美餐。
Thank you for this wonderful meal.
san ke you fou re si wan de fou mi ou
三可 油 否 热斯 弯德 否 迷欧

Figure 1: Snippet from phrasebook

Phrasebooks permit a form of accurate, personal, oral communication that speech-to-speech translation devices lack. However, the user is limited to a small set of fixed phrases. In this paper, we lift this restriction by designing and evaluating a software program with the following:

- Input: Text entered by the speaker, in her own language.
- Output: Phonetic rendering of a foreign-language translation of that text, which, when pronounced by the speaker, can be understood by the listener.

The main challenge is that different languages have different orthographies, different phoneme inventories, and different phonotactic constraints, so mismatches are inevitable. Despite this, the system's output should be both unambiguously pronounceable by the speaker and readily understood by the listener.

Our goal is to build an application that covers many language pairs and directions. The current paper describes a single system that lets a Chinese person speak English.

We take a statistical modeling approach to this problem, as is done in two lines of research that are most related. The first is machine transliteration (Knight and Graehl, 1998), in which names and technical terms are translated across languages with different sound systems. The other is respelling generation (Hauer and Kondrak, 2013), where an English speaker is given a phonetic hint about how to pronounce a rare or foreign word to another English speaker. By contrast, we aim

Chinese	已经八点了
English	It’s eight o’clock now
Chinglish	意思埃特额克劳克闹 (yi si ai te e ke lao ke nao)
Chinese	这件衬衫又时髦又便宜
English	this shirt is very stylish and not very expensive
Chinglish	迪思舍特意思危锐思掉利失安的闹特危锐伊克思班西五
Chinese	我们外送的最低金额是15美金
English	our minimum charge for delivery is fifteen dollars
Chinglish	奥儿米尼们差只佛低利沃锐意思发五听到乐思

Table 1: Examples of <Chinese, English, Chinglish> tuples from a phrasebook.

to help people issue full utterances that cross language barriers.

2 Evaluation

Our system’s input is Chinese. The output is a string of Chinese characters that approximate English sounds, which we call Chinglish. We build several candidate Chinese-to-Chinglish systems and evaluate them as follows:

- We compute the normalized edit distance between the system’s output and a human-generated Chinglish reference.
- A Chinese speaker pronounces the system’s output out loud, and an English listener takes dictation. We measure the normalized edit distance against an English reference.
- We automate the previous evaluation by replace the two humans with: (1) a Chinese speech synthesizer, and (2) a English speech recognizer.

3 Data

We seek to imitate phonetic transformations found in phrasebooks, so phrasebooks themselves are a good source of training data. We obtained a collection of 1312 <Chinese, English, Chinglish> phrasebook tuples [1] (see Table 1).

We use 1182 utterances for training, 65 for development, and 65 for test. We know of no other computational work on this type of corpus.

Our Chinglish has interesting gross empirical properties. First, because Chinglish and Chinese are written with the same characters, they render the same inventory of 416 distinct syllables. However, the distribution of Chinglish syllables differs a great deal from Chinese (Table 2). Syllables “si” and “te” are very popular, because while consonant clusters like English “st” are impossible to reproduce exactly, the particular vowels in “si” and “te” are fortunately very weak.

Frequency Rank	Chinese	Chinglish
1	de	si
2	shi	te
3	yi	de
4	ji	yi
5	zhi	fu

Table 2: Top 5 frequent syllables in Chinese (McEnery and Xiao, 2004) and Chinglish

We find that multiple occurrences of an English word type are generally associated with the same Chinglish sequence. Also, Chinglish characters do not generally span multiple English words. It is reasonable for “can I” to be rendered as “kan nai”, with “nai” spanning both English words, but this is rare.

4 Model

We model Chinese-to-Chinglish translation with a cascade of weighted finite-state transducers (wFST), shown in Figure 2. We use an online MT system to convert Chinese to an English word sequence (Eword), which is then passed through FST A to generate an English sound sequence (Epron). FST A is constructed from the CMU Pronouncing Dictionary (Weide, 2007).

Next, wFST B translates English sounds into Chinese sounds (Pinyin-split). Pinyin is an official syllable-based romanization of Mandarin Chinese characters, and Pinyin-split is a standard separation of Pinyin syllables into initial and final parts. Our wFST allows one English sound token to map

[1] Dataset can be found at `http://www.isi.edu/natural-language/mt/chinglish-data.txt`

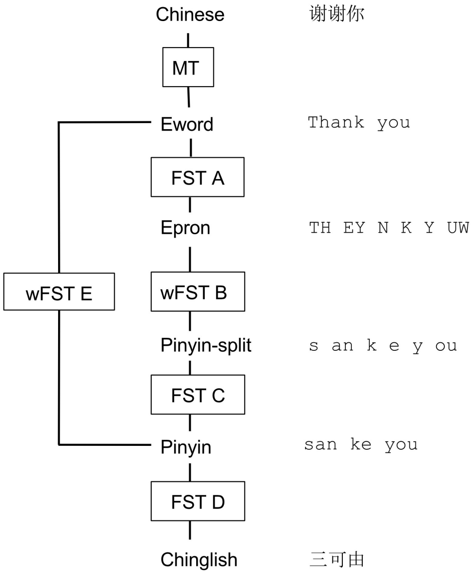


Figure 2: Finite-state cascade for modeling the relation between Chinese and Chinglish.

to one or two Pinyin-split tokens, and it also allows two English sounds to map to one Pinyin-split token.

Finally, FST C converts Pinyin-split into Pinyin, and FST D chooses Chinglish characters. We also experiment with an additional wFST E that translates English words directly into Chinglish.

5 Training

FSTs A, C, and D are unweighted, and remain so throughout this paper.

5.1 Phoneme-based model

We must now estimate the values of FST B parameters, such as P($si|S$). To do this, we first take our phrasebook triples and construct sample string pairs <Epron, Pinyin-split> by pronouncing the phrasebook English with FST A, and by pronouncing the phrasebook Chinglish with FSTs D and C. Then we run the EM algorithm to learn FST B parameters (Table 3) and Viterbi alignments, such as:

g	r	ae n	d
g e	r	uan	d e

labeled Epron	Pinyin-split	$P(p\|e)$
d	d	0.46
	d e	0.40
	d i	0.06
	s	0.01
ao r	u	0.26
	o	0.13
	ao	0.06
	ou	0.01

Table 3: Learned translation tables for the phoneme based model

5.2 Phoneme-phrase-based model

Mappings between phonemes are context-sensitive. For example, when we decode English "grandmother", we get:

g	r	ae n	d	m	ah	dh	er
g e	r	an	d e	m u	e	d	e

where as the reference Pinyin-split sequence is:

g e r uan d e m a d e

Here, "ae n" should be decoded as "uan" when preceded by "r". Following phrase-based methods in statistical machine translation (Koehn et al., 2003) and machine transliteration (Finch and Sumita, 2008), we model substitution of longer sequences. First, we obtain Viterbi alignments using the phoneme-based model, e.g.:

g	r	ae n	d	m	ah	dh	er
g e	r	uan	d e	m	a	d	e

Second, we extract phoneme phrase pairs consistent with these alignments. We use no phrase-size limit, but we do not cross word boundaries. From the example above, we pull out phrase pairs like:

g → g e
g r → g e r
...
r → r
r ae n → r uan
...

We add these phrase pairs to FST B, and call this the phoneme-phrase-based model.

5.3 Word-based model

We now turn to WFST E, which short-cuts directly from English words to Pinyin. We create <English, Pinyin> training pairs from our phrasebook simply by pronouncing the Chinglish with FST D. We initially allow each English word type to map to any sequence of Pinyin, up to length 7, with uniform probability. EM learns values for parameters like P(nai te|$night$), plus Viterbi alignments such as:

Model	Top-1 Overall Average Edit Distance	Top-1 Valid Average Edit Distance	Coverage
Word based	0.664	0.042	29/65
Word-based hybrid training	0.659	0.029	29/65
Phoneme based	0.611	0.583	63/65
Phoneme-phrase based	0.194	0.136	63/65
Hybrid training and decoding	0.175	0.115	63/65

Table 4: English-to-Pinyin decoding accuracy on a test set of 65 utterances. Numbers are average edit distances between system output and Pinyin references. Valid average edit distance is calculated based only on valid outputs (e.g. 29 outputs for word based model).

accept	tips
a ke sha pu	te ti pu si

Notice that this model makes alignment errors due to sparser data (e.g., the word “tips” and “ti pu si” only appear once each in the training data).

5.4 Hybrid training

To improve the accuracy of word-based EM alignment, we use the phoneme based model to decode each English word in the training data to Pinyin. From the 100-best list of decodings, we collect combinations of start/end Pinyin syllables for the word. We then modify the initial, uniform English-to-Pinyin mapping probabilities by giving higher initial weight to mappings that respect observed start/end pairs. When we run EM, we find that alignment errors for “tips” in section 5.3 are fixed:

accept	tips
a ke sha pu te	ti pu si

5.5 Hybrid decoding

The word-based model can only decode 29 of the 65 test utterances, because wFST E fails if an utterance contains a new English word type, previously unseen in training. The phoneme-based models are more robust, able to decode 63 of the 65 utterances, failing only when some English word type falls outside the CMU pronouncing dictionary (FST A).

Our final model combines these two, using the word-based model for known English words, and the phoneme-based models for unknown English words.

6 Experiments

Our first evaluation (Table 4) is intrinsic, measuring our Chinglish output against references from the test portion of our phrasebook, using edit distance. Here, we start with reference English and measure the accuracy of Pinyin syllable production, since the choice of Chinglish character does not affect the Chinglish pronunciation. We see that the Word-based method has very high accuracy, but low coverage. Our best system uses the Hybrid training/decoding method. As Table 6 shows, the ratio of unseen English word tokens is small, thus large portion of tokens are transformed using word-based method. The average edit distance of phoneme-phrase model and that of hybrid training/decoding model are close, indicating that long phoneme-phrase pairs can emulate word-pinyin mappings.

	Unseen	Total	Ratio
Word Type	62	249	0.249
Token	62	436	0.142

Table 6: Unseen English word type and tokens in test data.

Model	Valid Average Edit Distance
Reference English	0.477
Phoneme based	0.696
Hybrid training and decoding	0.496

Table 7: Chinglish-to-English accuracy in dictation task.

Our second evaluation is a dictation task. We speak our Chinglish character sequence output aloud and ask an English monolingual person to transcribe it. (Actually, we use a Chinese synthesizer to remove bias.) Then we measure edit distance between the human transcription and the reference English from our phrasebook. Results are shown in Table 7.

Chinese	年夜饭都要吃些什么
Reference English	what do you have for the Reunion dinner
Reference Chinglish	沃特 杜 又 海夫 佛 则 锐又尼恩 低呢
Hybrid training/decoding Chinglish	我忒 度 优 嗨佛 佛 得 瑞优你恩 低呢
Dictation English	what do you have for the reunion dinner
ASR English	what do you high for 43 Union Cena
Chinese	等等我
Reference English	wait for me
Reference Chinglish	唯特 佛 密 (wei te fo mi)
Hybrid training/decoding Chinglish	位忒 佛 密 (wei te fo mi)
Dictation English	wait for me
ASR English	wait for me

Table 5: Chinglish generated by hybrid training and decoding method and corresponding recognized English by dictation and automatic synthesis-recognition method.

Model	Valid Average Edit Distance
Word based	0.925
Word-based hybrid training	0.925
Phoneme based	0.937
Phoneme-phrase based	0.896
Hybrid training and decoding	0.898

Table 8: Chinglish-to-English accuracy in automatic synthesis-recognition (ASR) task. Numbers are average edit distance between recognized English and reference English.

Finally, we repeat the last experiment, but removing the human from the loop, using both automatic Chinese speech synthesis and English speech recognition. Results are shown in Table 8. Speech recognition is more fragile than human transcription, so edit distances are greater. Table 5 shows a few examples of the Chinglish generated by the hybrid training and decoding method, as well as the recognized English from the dictation and ASR tasks.

7 Conclusions

Our work aims to help people speak foreign languages they don't know, by providing native phonetic spellings that approximate the sounds of foreign phrases. We use a cascade of finite-state transducers to accomplish the task. We improve the model by adding phrases, word boundary constraints, and improved alignment.

In the future, we plan to cover more language pairs and directions. Each target language raises interesting new challenges that come from its natural constraints on allowed phonemes, syllables, words, and orthography.

References

Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*, pages 13–18.

Bradley Hauer and Grzegorz Kondrak. 2013. Automatic generation of English respellings. In *Proceedings of NAACL-HLT*, pages 634–643.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Anthony McEnery and Zhonghua Xiao. 2004. The lancaster corpus of Mandarin Chinese: A corpus for monolingual and contrastive language study. *Religion*, 17:3–4.

R Weide. 2007. The CMU pronunciation dictionary, release 0.7a.