INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Deriving phonetic transcriptions and discovering word segmentations for speech-to-speech translation in low-resource settings

Andrew Wilkinson[1], *Tiancheng Zhao*[1], *Alan W Black*[1]

[1]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA
{aewilkin, tianchez, awb}@cs.cmu.edu

Abstract

We investigate speech-to-speech translation where one language does not have a well-defined written form. We use English-Spanish and Mandarin-English bitext corpora in order to provide both gold-standard text-based translations and experimental results for different levels of automatically derived symbolic representations from speech. We constrain our experiments such that the methods developed can be extended to low-resource languages. We derive different phonetic representations of the source texts in order to model the kinds of transcriptions that can be learned from low-resource-language speech data. We experiment with different methods of clustering the elements of the phonetic representations together into word-like units. We train MT models on the resulting texts, and report BLEU scores for the different representations and clustering methods in order to compare their effectiveness. Finally, we discuss our findings and suggest avenues for future research.
Index Terms: speech-to-speech, machine translation, segmentation, low-resource, unwritten languages

1. Introduction

The task of creating a speech-to-speech machine translation (MT) system generally assumes the existence of a symbolic representation schema for both the source and the target language. For the usual speech-to-speech (S2S) system, an MT system is first trained using some corpus of bitext, and then a speech recognition component for the source language and a speech synthesis component for the target language are added onto that system to complete the S2S pipeline. This model, however, is inadequate to account for cases in which a text representation for one or both languages is unavailable, or in which only an imprecise representation exists. When dealing with low-resource languages and scenarios in which only audio data is collected, it may be the case that a language lacks a standardized written form; or that no one is available to transcribe the audio; or that transcription is performed by a nonnative or nonexpert worker; or that no adequate speech recognition system exists.

If a S2S system is to be built, if no text is available, we must first have a way of inducing a text representation of the audio in order to train the MT system. Approaches have been developed to discover phonemes from audio that output strings of phonemes as atomic units. That is, the phonemes are the "words" of the resulting sentences; they are not grouped into any larger units of representation. Once we have a string of phonemes for each sentence, we can use that as the text representation of the side of the parallel corpus for the language in question (whether source or target). The resulting MT system will work, but will underperform a comparable system that is able to use orthographic or phonemic words as a representation. In this paper, we address the question: Given a phonemic string representation of a language, and no language-specific resources, what is the best way to automatically cluster phonemes into larger units such that the resulting representation optimizes MT quality?

In our investigation, we used language pairs (English-Spanish, Mandarin-English) for which a standard orthographic representation was available for both source and target, in order to compute oracle MT evaluation scores. We created phonemic representations of the source language from the original (text-only) corpora, to simulate having had only audio to begin with. In each case, the target language was represented in the conventional orthographic form, allowing MT scores to be computed over words, as is usual, rather than over phonemes.

2. Related Work

In Stahlberg et al. [1], the authors build an ASR (automatic speech recognition) system for Slovene using a phoneme recognizer for a closely related language (Croatian) and a novel string clustering method. The phoneme sequences are aligned with words in three source languages; these cross-lingual phoneme-to-word alignments are used to induce word-like units from the phonemes [2]. Similar units that may represent the same Slovene word, but differ due to recognition and alignment errors, are grouped together and used to construct a pronunciation dictionary and a unigram language model. These are used together with a Croatian acoustic model to recognize Slovene. In contrast, we explore the utility of a phoneme recognizer both for its intended language (English) and for an unrelated language (Mandarin); we use different statistical methods for clustering phonemes into words; and our goal is to optimize MT quality rather than speech recognition per se.

In Goldwater et al. [3], the problem of word segmentation is approached from the perspective of modeling the processes by which an infant language learner may identify word boundaries in speech. By computationally investigating approaches based on different assumptions about the statistical nature of phonemes and syllables, the authors show that successful word segmentation from speech relies upon modeling both word-internal and interword patterns. We use the code developed for this work in our experiments, showing applications of the authors' work to other problems in speech recognition.

The work of Muthukumar and Black [4] presents an alternate method of deriving phonetic transcriptions of speech when no regular transcriptions or linguistic knowledge are available, based on predicting articulatory features and clustering them into "inferred phonemes."

Duong et al. [5] use a neural, attentional model to learn alignments between source-language (Spanish) phonemes and

http://dx.doi.org/10.21437/Interspeech.2016-1319

target-language (English) words, outperforming other aligners and benefiting the translation task. Further, they align source speech directly with target words, without involving any source-language-specific knowledge.

3. English-Spanish

The first language pair for our experiments was English-Spanish, for which we used part of the English-Spanish Europarl corpus. Out of a total of 2.1 million sentences, we used 490,000 for training, 1000 for tuning, and 5000 for testing. Larger tuning sets were eschewed due to memory limitations in loading unusually large tables. All sentences used were over 10 tokens in length. All MT models were trained using Moses, in a standard phrase-based approach incorporating IRSTLM and mgiza. Where conventional orthography was present, sentences were truecased and tokenized.

3.1. Approaches

The overall goal is to discover the best method(s) for learning to cluster phonemes that have been derived from audio-only data, or text data in an imperfect representation of the language, into word-like units, as judged by comparing evaluations of MT system quality. For the Europarl data, no audio is available. Instead, we used two different methods of transforming the text into strings of phonemes, to simulate the representations that can be derived directly from audio. We can use the results from these experiments to inform our approach in situations where no text is present.

1. Method 1: We used a feature of the speech synthesis engine Flite [6] to produce phonemes from the English data. This produces a string of phonemes for each word of the text in a deterministic manner, and hence, does not represent the variations in pronunciation present in actual human speech, nor noise introduced by (imperfect) speech recognition. Its utility is to provide a noise-free phonetic transcription against which to compare results from other representations.
2. Method 2: We synthesized the corpus, creating an audio file for each sentence, and then performed phoneme discovery on the audio in the same manner as if it were true human speech. For synthesis we used eight Flite voices {aew, bdl, clb, eey, jmk, ljm, rms, slt}, seven from American English speakers and one from a Canadian English speaker. For phoneme discovery we used the system built by Sitaram et al. [7]. This incorporated an English phonemic language model trained on a corpus of transcribed TED talks that were converted into Arpabet notation using the CMU Pronouncing Dictionary, and an English acoustic model trained on the 1997 English Broadcast News Speech (HUB4) corpus (LDC98S71).

For comparison, we also computed scores for an oracle MT model that uses regular orthographic words for both source and target, without any phonemic transformations.

For each of the above methods, we compared four different clustering approaches:

3.1.1. Phonemes only

For our baseline, we trained a system using the "raw," unclustered phonemes generated by each method as the source text. Table 1 shows examples of text generated by the two methods.

Table 1: Examples with raw phonemes

Original	I declare resumed the session of the European Parliament adjourned on . . .
Method 1	ay d ih k l eh r r ih z uw m d dh ax s eh sh ax n aa v dh ax y uh r ax p iy ax n p aa r l ax m ax n t ax jh er n d aa n . . .
Method 2	AY D IH K L EH R IY Z D UW IH NG DH IH S AE SH AH N AH V DH AE T Y AO R P IY AE N D P AA R T L IH M AE N D IH JH ER N D AA N . . .

3.1.2. Naïve syllable clustering

We next used an approach of clustering phonemes into syllables that does not take into account any sophisticated phonotactic knowledge. While rules and constraints for English syllable structure are well understood, we wanted to model a situation where this information is *a priori* unknown. Phonemes in the inventory are identified as either "C" or "V," and a range of potential syllable structures are ranked by length, from the long and complex to the most basic (CCCVCCCC, CCCVCCC, . . . CV, VC, VV, V). For each syllable structure template in this list, patterns of individual phonemes matching the template in a sentence are clustered together. Table 2 shows example output of this approach as applied to the texts from Table 1.

Table 2: Examples with naïve clustering

Method 1	ay d_ih k_l_eh_r_r_ih z_uw m_d_dh_ax_s_eh sh_ax_n aa v_dh_ax_y uh r_ax_p iy ax n_p_aa_r_l ax m_ax n_t_ax_jh er n_d_aa_n_f_r . . .
Method 2	AY D_IH K_L_EH_R IY Z_D_UW IH NG_DH_IH_S AE SH_AH_N AH V_DH_AE_T_Y AO R_P_IY AE N_D_P_AA_R_T_L IH M_AE N_D_IH_JH ER N_D_AA_N_F_R . . .

3.1.3. Most frequent ngrams

We next used an approach in which we calculate the k most frequent ngrams in the corpus and combine those ngrams into clusters, then repeat the process on the resulting text, for a total of p iterations. Within the 50 most frequent ngrams for each iteration, values for n were almost always 2, and occasionally 3. We ran grid search experiments on the method 1 text to determine a general neighborhood for optimal values of k and p, and then used those values ($k = 10$, $p = 25$) for the method 2 text as well. Table 3 shows example output of this approach.

3.1.4. Goldwater approach

For our final approach, we used the Dirichlet process/Gibbs sampler word segmentation algorithm (version 1.2) created by Goldwater et al. [3]. Table 4 shows example output of this approach.

3.2. Results

Table 5 shows the BLEU scores obtained over our test set by each of the four approaches above, for each of the two methods of producing strings of phonemes from the original corpus.

Table 3: Examples with most-frequent-ngrams clustering

Method 1	ay_d ih_k_l_eh_r r ih_z uw m d dh_ax_s_eh_sh_ax_n aa_v_dh_ax y_uh_r_ax_p_iy_ax_n p_aa_r_l_ax_m_ax_n_t_ax_jh er n_d aa_n . . .
Method 2	AY_D IH_K_L EH_R_IY_Z D UW_IH_NG DH_IH_S AE_SH AH_N AH_V_DH_AE_T Y_AO_R P_IY AE_N_D P_AA_R_T L IH_M_AE_N_D IH_JH ER_N_D AA_N . . .

Table 4: Examples with Goldwater clustering

Method 1	aydihklehr rihzuwmddhaxseh shaxnaav dhaxyuhraxpiyaxn paarlaxmaxnt axjhernd aanfraydiy . . .
Method 2	AYDIHKL EHRIYZ DUWIHNGDHIHS AESHAHNAHV DHAETYAORPIY AEND PAARTLIHM AEND IHJHERN D AAN . . .

Table 5: English-Spanish Results (BLEU)

	Words	Raw phonemes	Naïve syllables	Ngrams	Goldwater
Oracle	35.76				
Method 1		20.45	22.81	29.12	**31.92**
Method 2		13.81	13.78	18.46	**20.20**

The relative quality of the results from the four clustering approaches is fairly consistent between the two methods. For method 1, the naïve syllable clustering approach outperformed the raw phonemes, whereas for method 2, there was no change in quality. We impute the lack of improvement here to the fact that due to the noise in the phoneme-recognized synthesized data, and the wide range of possible complex (multi-consonant) syllables allowed by the syllable template, the model is unable to learn good alignments for a significant proportion of the tokens in the input. There are also many out-of-vocabulary tokens in the test set; the output is peppered with untranslated tokens such as "N_K_AO_R_T_F_R" or "N_HH_EH_L_D_T_W."

It is evident that the most sophisticated approach, the Goldwater algorithm, does the best job at this task.

4. Mandarin-English

The second language pair in our experiments was Mandarin-English. We used the FBIS Chinese-English corpus (LDC2003E14), with 303,356 pairs of sentences. We randomly divided the corpus into 296,470 sentences for training, 1037 for tuning, and 5489 for testing; this brought the sizes of the sets roughly into proportion with the sets for English-Spanish. The corpus is mostly in the broadcast news domain.

4.1. Approaches

For this language pair, we organize our experiments along the dimensions of token representation and token granularity:

4.1.1. Token representation

The original Mandarin sentences are written in hanzi characters. In order to be able to extend our results to zero- or low-resource languages, we explored four representations of the original text with increasing abstraction and concomitant difficulty for MT, that model having different levels of knowledge about the language at hand. The representations (with examples in Table 6) are:

1. Hanzi: The original data in the form of hanzi characters.
2. Pinyin: We converted hanzi characters into pinyin (with tones) using the open-source Python library Dragon Mapper [8]. This representation is closely related to the pronunciation of the language.
3. Pinyin (toneless): We removed the tone information, since standard ASR usually does not handle tonal variations.
4. ASR: We first synthesised the Mandarin speech using Festival and then performed speech recognition with CMU Sphinx using the same phoneme recognizer used in Section 3.

4.1.2. Token granularity

Furthermore, in order to study the performance impact of missing word boundaries, we used different levels of granularity for the atomic "word" unit (with examples in Table 7):

1. Syllable: This level removes the oracle word boundary such that each token is a single hanzi character or the corresponding pinyin for it; e.g., "火车" becomes "火" and "车."
2. Phone: This level breaks down the pinyin of individual characters into consonant-vowel-consonant structure. The positions of vowels are determined based on a set of rules detecting vowels such as a, á, etc.
3. Goldwater: This level uses the Goldwater word segmentation algorithm to cluster the pinyin and ASR representations into larger units.

Table 6: Examples with different representation

English gloss	an international audience
Hanzi	国际 视听
Pinyin	guójì shìtīng
Pinyin (toneless)	guoji shiting
ASR	K L IH K S IY SH IY EY T S L IH M P

Table 7: Examples with different granularity

English gloss	an international audience
Word (hanzi)	国际 视听
Word (pinyin)	guójì shìtīng
Syllable (pinyin)	guó jì shì tīng
Phone (pinyin)	g uó j ì sh ì t ī ng
Goldwater (pinyin)	guójì shìtīng
Phone (ASR)	K L IH K S IY SH IY EY T S L IH M P
Goldwater (ASR)	KLIHKSIY SHIYEYT S LIHMP

One complication of using the Goldwater code is that it can only handle input tokens in ASCII encoding, whereas pinyin is

in UTF-8 and the number of unique tokens is far larger than the maximum number of ASCII code points (127). Therefore, we developed mapping software that first encodes all pinyin tokens into unique ASCII code and all tokens that are not in the alphabet (such as punctuation and numbers) into a special OOV token. This software successfully compresses the phone (CVC) data into 123 unique ASCII tokens. After the clustering is finished, the software maps the ASCII tokens back to original pinyin form and reconstructs the OOV tokens, so that the final sentences output by the process are suitable for input to Moses.

4.2. Results

For the above four different text representation schemas at four different levels of granularity, we trained MT models using Moses as with the English-Spanish experiments. The resulting BLEU scores over the test data are presented in Table 8.

Table 8: Mandarin-English Results (BLEU)

	Word	Syllable	Phone	Goldwater
Hanzi	29.05 (Oracle)	27.27	N/A	N/A
Pinyin	28.98	26.78	14.29	26.80
Pinyin (toneless)	28.30	25.90	18.62	25.15
ASR	N/A	N/A	4.73	6.97

For the experiments represented in the first three rows, where we have a deterministic mapping from hanzi to pinyin, the results indicate that losing word boundaries and textual representation impacts the MT performance in different ways. Converting hanzi characters to their corresponding pinyin form leads only to a small drop in the BLEU score. This is expected, since although there are situations where different hanzi characters map to the same pinyin pronunciations, there generally is a one-to-one mapping between them. Therefore, for the MT models, they are simply similar tokens in different lexical forms, which does not significantly affect the translation process.

However, in comparing columns, we observe a significant drop in score when going from syllable to phone. In this, we not only lose distinct word boundaries, but also lose distinct boundaries between characters and syllables. One explanation is that in the phone representation, the resulting text becomes much longer than its original form, which leads to a high fertility (three to six source tokens map to one English word). Fortunately, we discovered that applying Goldwater's algorithm can recover the translation quality given the phone representation, resulting in performance similar to that at the syllable level. We conducted further error analysis on the clustering result of the Goldwater algorithm, and show a example in Table 7. We can see that it successfully recovers the grouping of pinyin into the correct words.

Finally, the loss of information inherent in speech synthesis and the noise introduced by the English phoneme recognizer greatly compromise the translation quality (for a BLEU score of 4.73). We found that although performance improves by using the Goldwater algorithm on the ASR output, the translation quality is still not satisfying. Several important factors may account for the significant drop in performance. Both the acoustic model and the language model of the phoneme recognizer are trained on English corpora, while the speech is synthesized by a single Mandarin voice. The acoustic properties of Mandarin are significantly different from English; the sets of phonemes in the two languages likewise differ considerably.

5. Conclusions and Future Work

In the English-Spanish experiments, we tried several methods of combining phonemes into word-like units, and evaluated their efficacy by comparing the results from training an MT system on the different texts. The Goldwater clustering process consistently outperformed all other methods that were tried, both on the gold-standard phonetic text of the corpus and on the noisy text from phonetic ASR of synthesized speech produced from many TTS voices, which in the aggregate we take to represent natural human speech.

In the Mandarin-English experiments, we focused on the different levels of granularity possible in representing a language very different from English, to explore how different methods of representing a language given imperfect knowledge and means can affect MT quality, in combination with clustering methods. Much information initially "lost" through decomposition of hanzi into different Romanized forms is recoverable with clustering.

For the phoneme recognition system, we incorporated an English acoustic model and an English phonetic language model. As expected, the recognition performed much better on English than on Mandarin. The English-Spanish experiments, thus, give an accurate picture of the relative strengths of the clustering methods on a language for which the recognition models are well suited. The Mandarin-English experiments give a picture of what results can be expected in the extreme opposite case. This is important because it reflects the realities of performing ASR on new or low-resource languages for which well suited models are unavailable. It also underlines the importance of leveraging what transcription schemas and linguistic knowledge are available. In this case, having access to just the consonant and vowel segments of the language, in Romanized form and without tone information, resulted in a reasonable translation when paired with the Goldwater algorithm.

In future work, we want to try using models for the ASR that are closer to the language in question, to determine how much improvement can be made thereby. This can help tell whether it is more worthwhile to spend time developing or adapting such models for new languages for use in inducing phonemic transcriptions, or to work on creating alternate transcription schemas that can be improved through clustering methods. Additionally, the ASR software that we use allows for retraining to improve the language model and to bootstrap a better recognizer, which we did not perform for this work.

We have thus far used bitext corpora for which we derived an audio representation. We would like to use a corpus for which real human speech is available, in addition to an accurate transcription and a parallel translation. Furthermore, it would be beneficial to use natural or spontaneous speech, as opposed to the "performed" speech from the Europarl corpus, e.g. Our methods work best with large amounts of data, but finding adequate speech corpora, even without accurate transcription and translation, is surprisingly hard.

We would like to incorporate alignment information produced by Moses in order to help group noisy versions of the same underlying words together, similarly to [1]. We began experiments using lattice decoding of word types extracted from the phrase tables, but more work is needed to improve computational tractability.

In all our experiments, the Goldwater algorithm was run using default hyperparameters. We would like to spend time tuning the hyperparameters to further optimize MT performance.

6. References

[1] F. Stahlberg, T. Schlippe, S. Vogel, and T. Schultz, "Towards automatic speech recognition without pronunciation dictionary, transcribed speech and text resources in the target language using cross-lingual word-to-phoneme alignment," *Spoken Language Technologies for Under-Resourced Languages*, 2014.

[2] ——, "Word segmentation through cross-lingual word-to-phoneme alignment," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 85–90.

[3] S. Goldwater, T. L. Griffiths, and M. Johnson, "A bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.

[4] P. K. Muthukumar and A. W. Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 2594–2598.

[5] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics, 2016, pp. 949–959.

[6] A. W. Black and K. A. Lenzo, "Flite: a small fast run-time synthesis engine," *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

[7] S. Sitaram, S. Palkar, Y. N. Chen, A. Parlikar, and A. W. Black, "Bootstrapping text-to-speech for speech processing in languages without an orthography," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7992–7996.

[8] T. Roten, "Dragon mapper," 2014–2015. [Online]. Available: https://pypi.python.org/pypi/dragonmapper