An Unsupervised Probability Model for Speech-to-Translation Alignment of Low-Resource Languages

Antonios Anastasopoulos and David Chiang
University of Notre Dame
{aanastas, dchiang}@nd.edu

Long Duong
University of Melbourne
lduong@student.unimelb.edu.au

Abstract

For many low-resource languages, spoken language resources are more likely to be annotated with translations than with transcriptions. Translated speech data is potentially valuable for documenting endangered languages or for training speech translation systems. A first step towards making use of such data would be to automatically align spoken words with their translations. We present a model that combines Dyer et al.'s reparameterization of IBM Model 2 (`fast_align`) and *k*-means clustering using Dynamic Time Warping as a distance measure. The two components are trained jointly using expectation-maximization. In an extremely low-resource scenario, our model performs significantly better than both a neural model and a strong baseline.

1 Introduction

For many low-resource languages, speech data is easier to obtain than textual data. And because speech transcription is a costly and slow process, speech is more likely to be annotated with translations than with transcriptions. This translated speech is a potentially valuable source of information – for example, for documenting endangered languages or for training speech translation systems.

In language documentation, data is usable only if it is interpretable. To make a collection of speech data usable for future studies of the language, something resembling interlinear glossed text (transcription, morphological analysis, word glosses, free translation) would be needed at minimum. New technologies are being developed to facilitate collection of translations (Bird et al., 2014), and there already exist recent examples of parallel speech collection efforts focused on endangered languages (Blachon et al., 2016; Adda et al., 2016). As for the other annotation layers, one might hope that a first pass could be done automatically. A first step towards this goal would be to automatically align spoken words with their translations, capturing information similar to that captured by word glosses.

In machine translation, statistical models have traditionally required alignments between the source and target languages as the first step of training. Therefore, producing alignments between speech and text would be a natural first step towards MT systems operating directly on speech.

We present a model that, in order to learn such alignments, adapts and combines two components: Dyer et al.'s reparameterization of IBM Model 2 (Dyer et al., 2013), more commonly known as `fast_align`, and *k*-means clustering using Dynamic Time Warping (Berndt and Clifford, 1994) as a distance measure. The two components are trained jointly using expectation-maximization.

We experiment on two language pairs. One is Spanish-English, using the CALLHOME and Fisher corpora. The other is Griko-Italian; Griko is an endangered language for which we created (and make freely available)[1] gold-standard translations and word alignments (Lekakou et al., 2013). In all cases, our model outperforms both a naive but strong baseline and a neural model (Duong et al., 2016).

[1] https://www3.nd.edu/~aanastas/griko/griko-data.tar.gz

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1255–1263,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

2 Background

In this section, we briefly describe the existing models that the two components of our model are based on. In the next section, we will describe how we adapt and combine them to the present task.

2.1 IBM Model 2 and `fast_align`

The IBM translation models (Brown et al., 1993) aim to model the distribution $p(\mathbf{e} \mid \mathbf{f})$ for an English sentence $\mathbf{e} = e_1 \cdots e_l$, given a French sentence $\mathbf{f} = f_1 \cdots e_m$. They all introduce a hidden variable $\mathbf{a} = a_1 \cdots a_l$ that gives the position of the French word to which each English word is aligned.

The general form of IBM Models 1, 2 and `fast_align` is

$$p(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) = p(l) \prod_{i=1}^{l} t(e_i \mid f_{a_i})\, \delta(a_i \mid i, l, m)$$

where $t(e \mid f)$ is the probability of translating French word f to English word e, and $\delta(a_i = j \mid i, l, m)$ is the probability of aligning the i-th English word with the j-th French word.

In Model 1, δ is uniform; in Model 2, it is a categorical distribution. Dyer et al. (2013) propose a reparameterization of Model 2, known as `fast_align`:

$$h(i, j, l, m) = -\left|\frac{i}{l} - \frac{j}{m}\right|$$

$$\delta(a_i \mid i, l, m) = \begin{cases} p_0 & a_i = 0 \\ (1 - p_0)\frac{\exp \lambda h(i, a_i, l, m)}{Z_\lambda(i, l, m)} & a_i > 0 \end{cases}$$

where the null alignment probability p_0 and precision $\lambda \geq 0$ are hyperparameters optimized by grid search. As $\lambda \to 0$, the distribution gets closer to the distribution of IBM Model 1, and as λ gets larger, the model prefers monotone word alignments more strongly.

2.2 DTW and DBA

Dynamic Time Warping (DTW) (Berndt and Clifford, 1994) is a dynamic programming method for measuring distance between two temporal sequences of variable length, as well as computing an alignment based on this distance. Given two sequences ϕ, ϕ' of length m and m' respectively, DTW constructs an $m \times m'$ matrix w. The warping path can be found by evaluating the following recurrence:

$$w_{i,j} = d(\phi_i, \phi'_j) + \min\{w_{i-1,j}, w_{i-1,j-1}, w_{i,j-1}\}$$

where d is a distance measure. In this paper, we normalize the cost of the warping path:

$$\mathrm{DTW}(\phi, \phi') = \frac{w_{m,m'}}{m + m'}$$

which lies between zero and one.

DTW Barycenter Averaging (DBA) (Petitjean et al., 2011) is an iterative approximate method that attempts to find a centroid of a set of sequences, minimizing the sum of squared DTW distances.

In the original definition, given a set of sequences, DBA chooses one sequence randomly to be a "skeleton." Then, at each iteration, DBA computes the DTW between the skeleton and every sequence in the set, aligning each of the skeleton's points with points in all the sequences. The skeleton is then refined using the found alignments, by updating each frame in the skeleton to the mean of all the frames aligned to it. In our implementation, in order to avoid picking a skeleton that is too short or too long, we randomly choose one of the sequences with median length.

3 Model

We use a generative model from a source-language speech segment consisting of feature frames $\boldsymbol{\phi} = \phi_1 \cdots \phi_m$ to a target-language segment consisting of words $\mathbf{e} = e_1 \ldots e_l$. We chose to model $p(\mathbf{e} \mid \boldsymbol{\phi})$ rather than $p(\boldsymbol{\phi} \mid \mathbf{e})$ because it makes it easier to incorporate DTW. The other direction is also possible, and we plan to explore it in future work.

In addition to the target-language sentence $\mathbf{e}$, our model hypothesizes a sequence $\mathbf{f} = f_1 \cdots f_l$ of source-language clusters (intuitively, source-language words), and spans (a_i, b_i) of the source signal that each target word e_i is aligned to. Thus, the clusters $\mathbf{f} = f_1 \cdots f_l$ and the spans $\mathbf{a} = a_1, \ldots, a_l$ and $\mathbf{b} = b_1, \ldots, b_l$ are the hidden variables of the model:

$$p(\mathbf{e} \mid \boldsymbol{\phi}) = \sum_{\mathbf{a},\mathbf{b},\mathbf{f}} p(\mathbf{e}, \mathbf{a}, \mathbf{b}, \mathbf{f} \mid \boldsymbol{\phi}).$$

The model generates $\mathbf{e}, \mathbf{a}, \mathbf{b}$, and $\mathbf{f}$ from $\boldsymbol{\phi}$ as follows.

1. Choose l, the number of target words, with uniform probability. (Technically, this assumes a maximum target sentence length, which we can just set to be very high.)

2. For each target word position $i = 1, \ldots, l$:
   (a) Choose a cluster f_i.
   (b) Choose a span of source frames (a_i, b_i) for e_i to be aligned to.
   (c) Generate a target word e_i from f_i.

Accordingly, we decompose $p(\mathbf{e}, \mathbf{a}, \mathbf{b}, \mathbf{f} \mid \boldsymbol{\phi})$ into several submodels:

$$\begin{aligned} p(\mathbf{e}, \mathbf{a}, \mathbf{b}, \mathbf{f} \mid \boldsymbol{\phi}) = p(l) \prod_{i=1}^{l} & u(f_i) \times \\ & s(a_i, b_i \mid f_i, \boldsymbol{\phi}) \times \\ & \delta(a_i, b_i \mid i, l, |\boldsymbol{\phi}|) \times \\ & t(e_i \mid f_i). \end{aligned}$$

Note that submodels δ and s both generate spans (corresponding to step 2b), making the model deficient. We could make the model sum to one by replacing $u(f_i)s(a_i, b_i \mid f_i, \boldsymbol{\phi})$ with $s(f_i \mid a_i, b_i, \boldsymbol{\phi})$, and this was in fact our original idea, but the model as defined above works much better, as discussed in Section 7.4. We describe both δ and s in detail below.

Clustering model The probability over clusters, $u(f)$, is just a categorical distribution. The submodel s assumes that, for each cluster f, there is a "prototype" signal $\boldsymbol{\phi}^f$ (cf. Ristad and Yianilos, 1998). Technically, the $\boldsymbol{\phi}^f$ are parameters of the model, and will be recomputed during the M step. Then we can define:

$$s(a, b \mid f, \boldsymbol{\phi}) = \frac{\exp(-\mathrm{DTW}(\boldsymbol{\phi}^f, \phi_a \cdots \phi_b)^2)}{\sum_{a,b=1}^{m} \exp(-\mathrm{DTW}(\boldsymbol{\phi}^f, \phi_a \cdots \phi_b)^2)}$$

where DTW is the distance between the prototype and the segment computed using Dynamic Time Warping. Thus s assigns highest probability to spans of $\boldsymbol{\phi}$ that are most similar to the prototype $\boldsymbol{\phi}^f$.

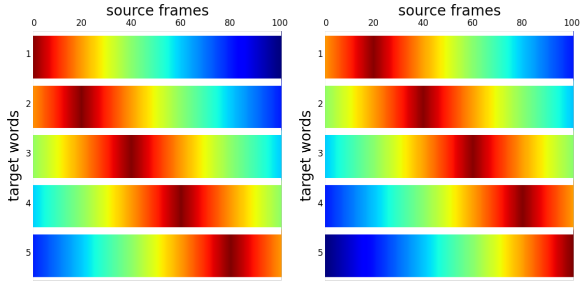

Figure 1: Sample distributions for the alignment variables a and b for $m = 100$, $l = 5$, $p_0 = 0$, $\lambda = 0.5$, and $\mu = 20$.

Distortion model The submodel δ controls the reordering of the target words relative to the source frames. It is an adaptation of fast_align to our setting, where there is not a single source word position a_i, but a span (a_i, b_i). We want the model to prefer the middle of the word to be close to the diagonal, so we need the variable a to be somewhat to the left and b to be somewhat to the right. Therefore, we introduce an additional hyperparameter μ which is intuitively the number of frames in a word. Then we define

$$\begin{aligned} h_a(i, j, l, m, \mu) &= -\left| \frac{i}{l} - \frac{j}{m - \mu} \right| \\ h_b(i, j, l, m, \mu) &= -\left| \frac{i}{l} - \frac{j - \mu}{m - \mu} \right| \\ \delta_a(a_i \mid i, l, m) &= \begin{cases} p_0 & a_i = 0 \\ (1 - p_0) \frac{\exp \lambda h_a(i, a_i, l, m)}{Z_\lambda(i, l, m)} & a_i > 0 \end{cases} \\ \delta_b(b_i \mid i, l, m) &= \begin{cases} p_0 & b_i = 0 \\ (1 - p_0) \frac{\exp \lambda h_b(i, b_i, l, m)}{Z_\lambda(i, l, m)} & b_i > 0 \end{cases} \\ \delta(a_i, b_i \mid i, l, m) &= \delta_a(a_i \mid i, l, m)\, \delta_b(b_i \mid i, l, m) \end{aligned}$$

where the $Z_\lambda(i, l, m)$ are set so that all distributions sum to one. Figure 1 shows an example visualisation of the the resulting distributions for the two variables of our model.

We set μ differently for each word. For each i, we set μ_i to be proportional to the number of *characters* in e_i, such that $\sum_i \mu_i = m$.

Translation model The translation model $t(e \mid f)$ is just a categorical distribution, in principle allowing a many-to-many relation between source clusters and target words. To speed up training (with nearly no change in accuracy, in our experiments), we restrict this relation so that there are k source clusters for each target word, and a source cluster uniquely determines its target word. Thus, $t(e \mid f)$ is fixed to either zero or one, and does not need to be reestimated. In our experiments, we set $k = 2$, allowing each target word to have up to two source-language translations/pronunciations. (If a source word has more than one target translation, they are treated as distinct clusters with distinct prototypes.)

4 Training

We use the hard (Viterbi) version of the Expectation-Maximization (EM) algorithm to estimate the parameters of our model, because calculating expected counts in full EM would be prohibitively expensive, requiring summations over all possible alignments.

Recall that the hidden variables of the model are the alignments (a_i, b_i) and the source words (f_i). The parameters are the translation probabilities $t(e_i \mid f)$ and the prototypes $(\boldsymbol{\phi}^f)$. The (hard) E step uses the current model and prototypes to find, for each target word, the best source segment to align it to and the best source word. The M step reestimates the probabilities $t(e \mid f)$ and the prototypes $\boldsymbol{\phi}^f$. We describe each of these steps in more detail below.

Initialization Initialization is especially important since we are using hard EM.

To initialize the parameters, we initialize the hidden variables and then perform an M step. We associate each target word type e with $k = 2$ source clusters, and for each occurrence of e, we randomly assign it one of the k source clusters.

The alignment variables a_i, b_i are initialized to

$$a_i, b_i = \arg\max_{a,b} \delta(a, b \mid i, l, m).$$

M step The M step reestimates the probabilities $t(e \mid f)$ using relative-frequency estimation.

The prototypes $\boldsymbol{\phi}^f$ are more complicated. Theoretically, the M step should recompute each $\boldsymbol{\phi}^f$ so as to maximize that part of the log-likelihood that depends on $\boldsymbol{\phi}^f$:

$$\begin{aligned} L_{\boldsymbol{\phi}^f} &= \sum_{\boldsymbol{\phi}} \sum_{i \mid f_i = f} \log s(a_i, b_i \mid f, \boldsymbol{\phi}) \\ &= \sum_{\boldsymbol{\phi}} \sum_{i \mid f_i = f} \log \frac{\exp(-\mathrm{DTW}(\phi^f, \phi_{a_i} \cdots \phi_{b_i})^2)}{Z(f, \boldsymbol{\phi})} \\ &= \sum_{\boldsymbol{\phi}} \sum_{i \mid f_i = f} -\mathrm{DTW}(\phi^f, \phi_{a_i} \cdots \phi_{b_i})^2 - \log Z(f, \boldsymbol{\phi}) \end{aligned}$$

where the summation over $\boldsymbol{\phi}$ is over all source signals in the training data. This is a hard problem, but note that the first term is just the sum-of-squares of the DTW distance between $\boldsymbol{\phi}^f$ and all source segments that are classified as f. This is what DBA is supposed to approximately minimize, so we simply set $\boldsymbol{\phi}^f$ using DBA, ignoring the denominator.

E step The (hard) E step uses the current model and prototypes to find, for each target word, the best source segment to align it to and the best source cluster.

In order to reduce the search space for **a** and **b**, we use the unsupervised phonetic boundary detection method of Khanagha et al. (2014). This method operates directly on the speech signal and provides us with candidate phone boundaries, on which we restrict the possible values for **a** and **b**, creating a list of candidate utterance spans.

Furthermore, we use a simple silence detection method. We pass the envelope of the signal through a low-pass filter, and then mark as "silence" time spans of 50ms or longer in which the magnitude is below a threshold of 5% relative to the maximum of the whole signal. This method is able to detect about 80% of the total pauses, with a 90% precision in a 50ms window around the correct silence boundary. We can then remove from the candidate list the utterance spans that include silence, on the assumption that a word should not include silences. Finally, in case one of the span's boundaries happens to be within a silence span, we also move it so as to not include the silence.

Hyperparameter tuning The hyperparameters p_0, λ, and μ are not learned. We simply set p_0 to zero (disallowing unaligned target words) and set μ as described above.

For λ we perform a grid search over candidate values to maximize the alignment F-score on the development set. We obtain the best scores with $\lambda = 0.5$.

5 Related Work

A first step towards modelling parallel speech can be performed by modelling phone-to-word alignment, instead of directly working on continuous speech. For example, Stahlberg et al. (2012) extend IBM Model 3 to align phones to words in order to build cross-lingual pronunciation lexicons. Pialign (Neubig et al., 2012) aligns characters and can be applied equally well to phones. Duong et al. (2016) use an extension of the neural attentional model of Bahdanau et al. (2015) for aligning phones to words and speech to words; we discuss this model below in Section 6.2.

There exist several supervised approaches that attempt to integrate speech recognition and machine translation. However, they rely heavily on the abundance of training data, pronunciation lexicons, or language models, and therefore cannot be applied in a low- or zero-resource setting.

A task somewhat similar to ours, which operates at a monolingual level, is the task of zero-resource spoken term discovery, which aims to discover repeated words or phrases in continuous speech. Various approaches (Ten Bosch and Cranen, 2007; Park and Glass, 2008; Muscariello et al., 2009; Zhang and Glass, 2010; Jansen et al., 2010) have been tried, in order to spot keywords, using segmental DTW to identify repeated trajectories in the speech signal.

Kamper et al. (2016) try to discover word segmentation and a pronunciation lexicon in a zero-resource setting, combining DTW with acoustic embeddings; their methods operate in a very low-vocabulary setting. Bansal (2015) attempts to build a speech translation system in a low-resource setting, by using as source input the simulated output of an unsupervised term discovery system.

6 Experiments

We evaluate our method on two language pairs, Spanish-English and Griko-Italian, against two baseline methods, a naive baseline, and the model of Duong et al. (2016).

6.1 Data

For each language pair, we require a sentence-aligned parallel corpus of source-language speech and target-language text. A subset of these sentences should be annotated with span-to-word alignments for use as a gold standard.

6.1.1 Spanish-English

For Spanish-English, we use the Spanish CALLHOME corpus (LDC96S35) and the Fisher corpus (LDC2010T04), which consist of telephone conversations between Spanish native speakers based in the US and their relatives abroad, together with English translations produced by Post et al. (2013). Spanish is obviously not a low-resource language, but we pretend that it is low-resource by not making use of any Spanish ASR or resources like transcribed speech or pronunciation lexicons.

Since there do not exist gold standard alignments between the Spanish speech and English words, we use the "silver" standard alignments produced by Duong et al. (2016) for the CALLHOME corpus, and followed the same procedure for the Fisher corpus as well. In order to obtain them, they first used a forced aligner to align the speech to its transcription, and GIZA++ with the `gdfa` symmetrization heuristic to align the Spanish transcription to the English translation. They then combined the two alignments to produce "silver" standard alignments between the Spanish speech and the English words.

The CALLHOME dataset consists of 17532 Spanish utterances, based on the dialogue turns. We first use a sample of 2000 sentences, out of which we use 200 as a development set and the rest as a test set. We also run our experiments on the whole dataset, selecting 500 utterances for a development set, using the rest as a test set. The Fisher dataset consists of 143355 Spanish utterances. We use 1000 of them as a development set and the rest as a test set.

6.1.2 Griko-Italian

We also run our model on a corpus that consists of about 20 minutes of speech in Griko, an endangered minority dialect of Greek spoken in south Italy, along with text translations into Italian (Lekakou et al., 2013).[2] The corpus consists of 330 mostly prompted utterances by nine native speakers. Although the corpus is very small, we use it to showcase the effectiveness of our method in a hard setting with extremely low resources.

All utterances were manually annotated and transcribed by a trained linguist and bilingual speaker of both languages, who produced the Griko transcriptions and Italian glosses. We created full translations into Italian and manually aligned the translations with the Griko transcriptions. We then com-

[2] http://griko.project.uoi.gr

bined the two alignments (speech-to-transcription and transcription-to-translation) to produce speech-to-translation alignments. Therefore, our comparison is done against an accurate "gold" standard alignment. We split the data into a development set of just 30 instances, and a test set of the remaining 300 instances.

6.1.3 Preprocessing

In both data settings, we treat the speech data as a sequence of 39-dimensional Perceptual Linear Prediction (PLP) vectors encoding the power spectrum of the speech signal (Hermansky, 1990), computed at 10ms intervals. We also normalize the features at the utterance level, shifting and scaling them to have zero mean and unit variance.

6.2 Baselines

Our naive baseline assumes that there is no reordering between the source and target language, and aligns each target word e_i to a source span whose length in frames is proportional to the length of e_i in characters. This actually performs very well on language pairs that show minimal or no reordering, and language pairs that have shared or related vocabularies.

The other baseline that we compare against is the neural network attentional model of Duong et al. (2016), which extends the attentional model of Bahdanau et al. (2015) to be used for aligning and translating speech, and, along with several modifications, achieve good results on the phone-to-word alignment task, and almost match the baseline performance on the speech-to-word alignment task.

7 Results

To evaluate an automatic alignment between the speech and its translation against the gold/silver standard alignment, we compute alignment precision, recall, and F-score as usual, but on links between source-language frames and target-language words.

7.1 Overview

Table 1 shows the precision, recall, and balanced F-score of the three models on the Spanish-English CALLHOME corpus (both the 2000-sentence subset and the full set), the Spanish-English Fisher corpus, and the Griko-Italian corpus.

			method	precision	recall	F-score
CALLHOME	spa-eng	2k sents	ours	**38.8**	38.9	**38.8**
			naive	31.9	**40.8**	35.8
			neural	23.8	29.8	26.4
		17k sents	ours	**38.4**	38.8	**38.6**
			naive	31.8	**40.7**	35.7
			neural	26.1	32.9	29.1
Fisher	spa-eng	143k sents	ours	**33.3**	28.7	**30.8**
			naive	24.0	**33.2**	27.8
gri-ita		300 sents	ours	**56.6**	51.2	**53.8**
			naive	42.2	**52.2**	46.7
			neural	24.6	30.0	27.0

Table 1: Our model achieves higher precision and F-score than both the naive baseline and the neural model on all datasets.

In all cases, our model outperforms both the naive baseline and the neural attentional model. Our model, when compared to the baselines, improves greatly on precision, while slightly underperforming the naive baseline on recall. In certain applications, higher precision may be desirable: for example, in language documentation, it's probably better to err on the side of precision; in phrase-based translation, higher-precision alignments lead to more extracted phrases.

The rest of the section provides a further analysis of the results, focusing on the extremely low-resource Griko-Italian dataset.

7.2 Speaker robustness

Figure 2 shows the alignments produced by our model for three utterances of the same sentence from the Griko-Italian dataset by three different speakers. Our model's performance is roughly consistent across these utterances. In general, the model does not seem significantly affected by speaker-specific variations, as shown in Table 2.

We do find, however, that the performance on male speakers is slightly higher compared to the female speakers. This might be because the female speakers' utterances are, on average, longer by about 2 words than the ones uttered by males.

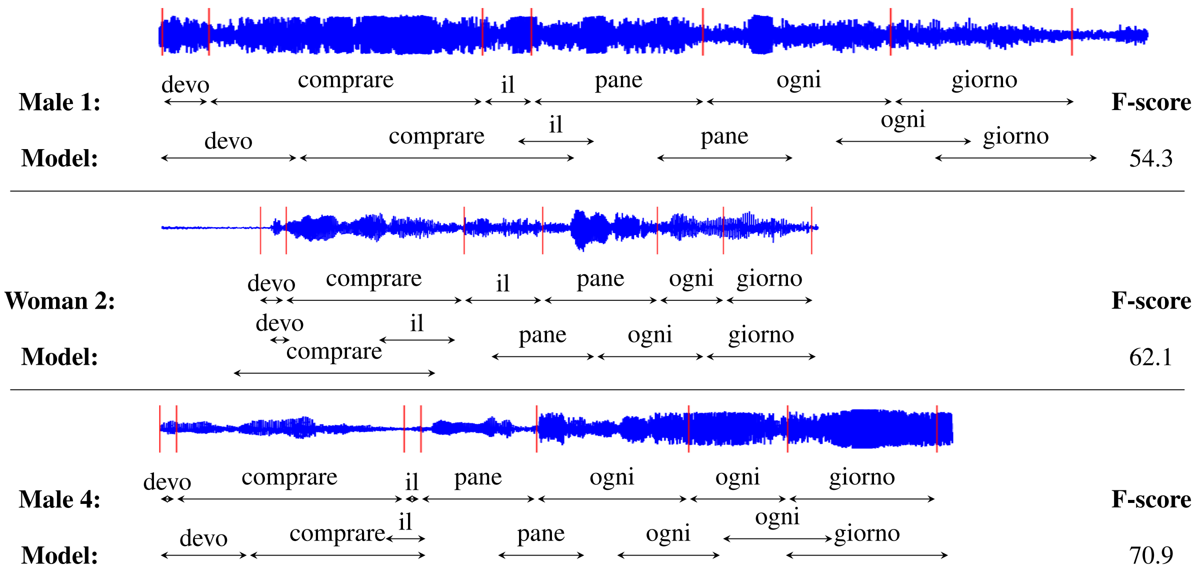

Figure 2: Alignments produced for the Italian sentence `devo comprare il pane ogni giorno` as uttered by three different Griko speakers.

speaker	utt	len	F-score
female 1	55	9.0	49.4
female 2	61	8.1	55.0
female 3	41	9.6	51.0
female 4	23	7.3	54.4
female 5	21	6.1	56.6
male 1	35	5.9	59.5
male 2	32	6.0	61.9
male 3	34	6.7	60.2
male 4	23	6.4	64.0

Table 2: Model performance (F-score) is generally consistent across speakers. The second column (utt) shows the number of utterances per speaker; the third (len), their average length in words.

7.3 Word level analysis

We also compute F-scores for each Italian word type. As shown in Figure 3, the longer the word's utterance, the easier it is for our model to correctly align it. Longer utterances seem to carry enough information for our DTW-based measure to function properly. On the other hand, shorter utterances are harder to align. The vast majority of Griko utterances that have less than 20 frames and are less accurately aligned correspond to monosyllabic determiners (`o`, `i`,`a`, `to`, `ta`) or conjunctions and prepositions (`ka`, `ce`, `en`, `na`, `an`). For such short utterances, there could be several parts of the signal that possibly match the prototype, leading the clustering component to prefer to align to wrong spans.

Furthermore, we note that rare word types tend to be correctly aligned. The average F-score for hapax legomena (on the Italian side) is 63.2, with 53% of them being aligned with an F-score higher than 70.0.

7.4 Comparison with proper model

As mentioned in Section 3, our model is deficient, but it performs much better than the model that sums to one (henceforth, the "proper" model): In the Spanish-English dataset (2000 sentences sample) the proper model yields an F-score of 32.1, performing worse than the naive baseline; in the Griko-

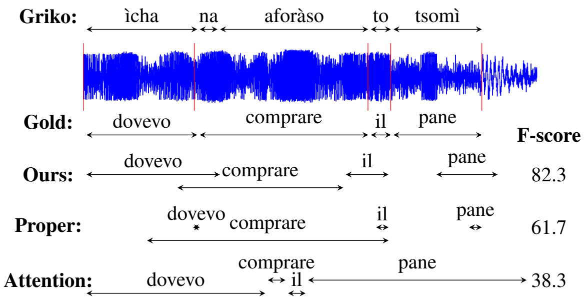

Figure 4: The deficient model performs very well, whereas the proper and the attentional model prefer extreme alignment spans. For example, the proper model's alignment for the words `dovevo` and `pane` are much too short.

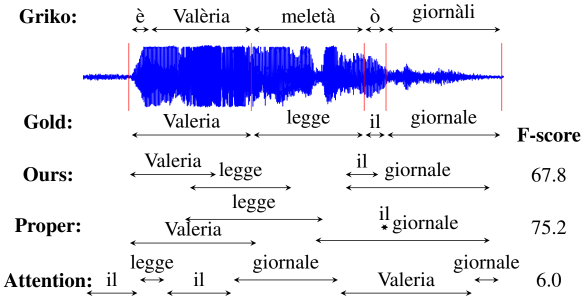

Figure 5: One of the rare examples where the proper model performs better than the deficient one. The hapax legomena `Valeria` and `giornali` are not properly handled by the attentional model.

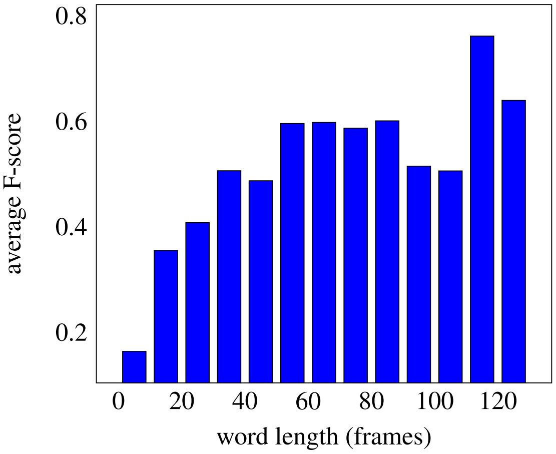

Figure 3: There is a positive correlation between average word-level F-score and average word utterance length (in frames).

Italian dataset, it achieves an F-score of 44.3, which is better than the baselines, but still worse than our model.

In order to further examine why this happens, we performed three EM iterations on the Griko-Italian dataset with our model (in our experience, three iterations are usually enough for convergence), and then computed one more E step with both our model and the proper model, so as to ensure that the two models would align the dataset using the exact same prototypes and that their outputs will be comparable.

In this case, the proper model achieved an overall F-score of 44.0, whereas our model achieved an F-score of 53.6. Figures 4 and 5 show the resulting alignments for two sentences. In both of these examples, it is clear that the proper model prefers extreme spans: the selected spans are either much too short or (less frequently) much too long. This is further verified by examining the statistics of the alignments: the average span selected by the proper model has a length of about 30 ± 39 frames whereas the average span of the alignments produced by our deficient model is 37 ± 24 frames. This means that the alignments of the deficient model are much closer to the gold ones, whose average span is 42 ± 26 frames.

We think that this is analogous to the "garbage collection" problem in word alignment. In the IBM word alignment models, if a source word f occurs in only one sentence, then EM can align many target words to f and learn a very peaked distribution $t(e \mid f)$. This can happen in our model and the proper model as well, of course, since IBM Model 2 is embedded in them. But in the proper model, something similar can also happen with $s(f \mid a, b)$: EM can make the span (a, b) large or small, and evidently making the span small allows it to learn a very peaked distribution $s(f \mid a, b)$. By contrast, our model has $s(a, b \mid f)$, which seems less susceptible to this kind of effect.

8 Conclusion

Alignment of speech to text translations is a relatively new task, one with particular relevance for low-resource or endangered languages. The model we propose here, which combines `fast_align` and k-means clustering using DTW and DBA, outperforms both a very strong naive baseline and a neural attentional model, on three tasks of various sizes.

The language pairs used here do not have very much word reordering, and more divergent language pairs should prove more challenging. In that case, the naive baseline should be much less competitive. Similarly, the `fast_align`-based distortion model may become less appopriate; we plan to try incorporating IBM Model 3 or the HMM alignment model (Vogel et al., 1996) instead. Finally, we will investigate downstream applications of our alignment methods, in the areas of both language documentation and speech translation.

Acknowledgements

We would like to thank Steven Bird, Eamonn Keogh, and the anonymous reviewers for their helpful feedback. This research was supported in part by NSF Award 1464553.

References

Gilles Adda, Sebastian Stüker, Martine Adda-Decker, Odette Ambouroue, Laurent Besacier, David Blachon, Hélène Bonneau-Maynard, Pierre Godard, Fatima Hamlaoui, Dmitry Idiatov, et al. 2016. Breaking the unwritten language barrier: The BULB project. *Procedia Computer Science*, 81:8–14.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Sameer Bansal. 2015. Speech translation without speech recognition. Master's thesis, University of Edinburgh.

Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proc. KDD*, pages 359–370.

Steven Bird, Lauren Gawne, Katie Gelbart, and Isaac McAlister. 2014. Collecting bilingual audio in remote indigenous communities. In *Proc. COLING*.

David Blachon, Elodie Gauthier, Laurent Besacier, Guy-Noël Kouarata, Martine Adda-Decker, and Annie Rialland. 2016. Parallel speech collection for under-resourced language studies using the Lig-Aikuma mobile device app. *Procedia Computer Science*, 81:61–66.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *Proc. NAACL HLT*, pages 949–959, June.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. NAACL HLT*.

Hynek Hermansky. 1990. Perceptual linear predictive (PLP) analysis of speech. *J. Acoustical Society of America*, 87(4):1738–1752.

Aren Jansen, Kenneth Church, and Hynek Hermansky. 2010. Towards spoken term discovery at scale with zero resources. In *Proc. INTERSPEECH*, pages 1676–1679.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE Trans. Audio, Speech, and Language Processing*.

Vahid Khanagha, Khalid Daoudi, Oriol Pont, and Hussein Yahia. 2014. Phonetic segmentation of speech signal using local singularity analysis. *Digital Signal Processing*.

Marika Lekakou, Valeria Baldiserra, and Antonis Anastasopoulos. 2013. Documentation and analysis of an endangered language: aspects of the grammar of Griko.

Armando Muscariello, Guillaume Gravier, and Frédéric Bimbot. 2009. Audio keyword extraction by unsupervised word discovery. In *Proc. INTERSPEECH*.

Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara. 2012. Machine translation without words through substring alignment. In *Proc. ACL*.

Alex S. Park and James R. Glass. 2008. Unsupervised pattern discovery in speech. *IEEE Trans. Audio, Speech, and Language Processing*, 16(1):186–197.

François Petitjean, Alain Ketterlin, and Pierre Gançarski. 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In *Proc. IWSLT*.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Felix Stahlberg, Tim Schlippe, Sue Vogel, and Tanja Schultz. 2012. Word segmentation through cross-lingual word-to-phoneme alignment. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*.

Louis Ten Bosch and Bert Cranen. 2007. A computational model for unsupervised word discovery. In *Proc. INTERSPEECH*, pages 1481–1484.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proc. COLING*, pages 836–841.

Yaodong Zhang and James R Glass. 2010. Towards multi-speaker unsupervised speech pattern discovery. In *Proc. ICASSP*, pages 4366–4369.