THE UNIVERSITY
of EDINBURGH

Low-resource speech translation

Sameer Bansal

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2019

Abstract

We explore the task of **s**peech-to-text **t**ranslation (ST), where speech in one language (source) is converted to text in a different one (target). Traditional ST systems go through an intermediate step where the source language speech is first converted to source language text using an automatic speech recognition (ASR) system, which is then converted to target language text using a machine translation (MT) system. However, this pipeline based approach is impractical for unwritten languages spoken by millions of people around the world, leaving them without access to free and automated translation services such as Google Translate. The lack of such translation services can have important real-world consequences. For example, in the aftermath of a disaster scenario, easily available translation services can help better co-ordinate relief efforts. How can we expand the coverage of automated ST systems to include scenarios which lack source language text? In this thesis we investigate one possible solution: we build ST systems to directly translate source language speech into target language text, thereby forgoing the dependency on source language text. To build such a system, we use only speech data paired with text translations as training data. We also specifically focus on low-resource settings, where we expect at most tens of hours of training data to be available for unwritten or endangered languages.

Our work can be broadly divided into three parts. First we explore how we can leverage prior work to build ST systems. We find that neural sequence-to-sequence models are an effective and convenient method for ST, but produce poor quality translations when trained in low-resource settings.

In the second part of this thesis, we explore methods to improve the translation performance of our neural ST systems which do not require labeling additional speech data in the low-resource language, a potentially tedious and expensive process. Instead we exploit labeled speech data for high-resource languages which is widely available and relatively easier to obtain. We show that pretraining a neural model with ASR data from a high-resource language, different from both the source and target ST languages, improves ST performance.

In the final part of our thesis, we study whether ST systems can be used to build applications which have traditionally relied on the availability of ASR systems, such as information retrieval, clustering audio documents, or question/answering. We build proof-of-concept systems for two downstream applications: topic prediction for speech

and cross-lingual keyword spotting. Our results indicate that low-resource ST systems can still outperform simple baselines for these tasks, leaving the door open for further exploratory work.

This thesis provides, for the first time, an in-depth study of neural models for the task of direct ST across a range of training data settings on a realistic multi-speaker speech corpus. Our contributions include a set of open-source tools to encourage further research.

Lay Summary

There is a broad demand for translation services in our daily lives as we travel more, study abroad, and conduct business globally. For general use, we do not always require the quality-of-service a human translator provides and instead are content with using automated translation tools such as Google Translate. But the complexity of building automated translation systems is high and as a result only around 100 languages, out of an estimated total of 7000, are currently supported by publicly available systems. Further exacerbating the challenge, the most commonly used method to build translation systems for written languages, does not work for unwritten ones.

An estimated 3000 languages, or 40% of total number of spoken languages, do not have a standard written form and are also considered endangered, with fewer than 1000 active speakers. In this work we explore building speech translation (ST) systems for unwritten languages. Such systems can potentially be of use during crisis-relief scenarios and aid language preservation efforts.

Our work demonstrates that useful ST systems can be built for unwritten languages using neural models. We also demonstrate a simple and effective method to improve translation performance for unwritten languages using widely available data from written ones.

Acknowledgements

"It takes a village" ... It does and in fact it took several! I have many many people to thank for helping me shape this thesis, and to an extent my life. I fear I will inadvertently fail to mention all the people who played a big role in the completion of this thesis. I apologize in advance.

Adam, walking into that MT lecture in January of 2015 turned out to be one of the best decisions of my life. Your inspirational teaching style led me to sign up for your class; the MSc dissertation project you proposed got me interested in what eventually became my PhD topic. You always made time for your students, no matter how gruelling your schedule was, and offered honest advice in a friendly and approachable manner, a trait many of your research students admire and depend upon. Beyond research, I was deeply inspired by your incredible kindness and passion towards creating a more equitable and diverse research group. You taught us that we are not in a race and we should pursue the tough questions. You encouraged us to collaborate and be willing to both ask for and provide feedback. Thank you for championing your students. And, you also introduced me to Sharon and Herman, which rounded up what turned out to be an extraordinary supervisory team.

Sharon, I am not sure that I can enumerate all the lessons I have learned under your supervision and how grateful I am for you having had the opportunity to work with you. Most importantly, you taught me the importance of communicating ideas clearly and being stoic during the phases where ideas were scarce and would not produce expected results. You would often ask me "What is your research question?" which would pull me out of a myopic view and help me organise my thoughts. During my internships or when working with other people, I would often ask myself "What would Sharon and Adam say or ask about this?" and this virtual conversation helped me immensely. It is now a part of my standard operating procedure. For all the papers we have written together, I can honestly say that it meant a lot to me if you and Adam were pleased with the work, as I value and respect your opinion the most.

Herman, thanks for all your guidance and detailed feedback on the drafts, experiments and ideas over the course of my MSc and PhD. Your PhD work set a strong foundation for researchers like me to build upon and your spirit of collaboration was extremely helpful and served as an example. Thank you for infusing many of our discussions with your enthusiasm and openness to new ideas. You make a great supervisor and I am excited about how you will continue to grow your research group taking on many new, interesting, and important challenges in this field.

Karen, thank you for inviting me to visit and spend a wonderful 3 months at TTI Chicago. This period coincided with a change in direction in my PhD and I deeply appreciate your mentorship at this point. I thoroughly enjoyed spending a winter in Chicago, both working with an amazing set of colleagues at TTIC and the city itself. The visit gave me refreshed sense of purpose and excitement about my research. Antonios and David, thank you for being wonderful collaborators. Your early work in this research area was inspiring and you were always open to sharing details from your experimental setup. Thank you for a productive partnership and for the opportunity to visit the University of Notre Dame and meet with your group.

Thank you to my thesis examiners Graham Neubig and Steve Renals for an engaging and thought-provoking discussion. I would also like to thank Simon King for being on my annual review committee.

To all of you, I have followed and respected your work for a long time and it was a pleasure to interact in this setting. Thank you for sharing your experience and knowledge.

Thank you to ILCC for providing access to computing infrastructure, eclectic mix of research talks and seminars, and a highly supportive and collaborative workplace. I will miss being part of the Forum. Thank you to level2/3/4admin for all your patience and support over the years and to John for maintaining the upkeep of our workspace, watering plants in our offices, refilling the coffee beans in the machines, and in general for going above and beyond to make our life easier.

Thank you to all my friends who supported me in countless ways over the years, listening to my research thoughts in the Forum hallways; reading crappy first, second, third . . . umpteenth drafts and patiently providing feedback; organizing and participating in movie nights and dinner outings, especially Korean food (Bibimbap group!) and Saturday morning pancakes; and most importantly the chit chat sessions (timepass) on the 3rd floor sofas about everything and nothing. My family here is large and includes ILCC and the dynamic AGORA group members but I would like to mention by name Sorcha, Nick, Federico, Naomi, Andreas, Ramon, Pippa, Joachim, Ondrej, Joanna, Joana, Jonathan, Maria, Laura, Kasia, Aibek, Seraphina, Kate, Julie-Anne, Sabine, Esma, Sander, and Yumnah.

Thank you to my office mates Mona, Irene, Spandana, Lucia, Avashna, and Gozde who ensured a comfortable and fun work environment with plenty of snacks and a friendly ear in reach.

Outside of Informatics, I would like to thank my former boss Vinayak and colleagues Gourav, Sindhu, and Deepa for all their guidance and support. My friends Avantika, Himani, and Lucas who welcomed me into their world and were an outlet for me to discuss life in Delhi and India in general among many other topics.

Thank you to Lucia, Marco, Clara, Yevgen, Ida, and Elizabeth for all your support and encouragement, especially during the final stages of my thesis including the defense and the corrections. I am very fortunate to have stumbled into this amazing set of friends who have positively influenced many aspects of my life and made me a better person. You have all made my life really tough by showing me immense love and making sure that I will thoroughly miss my time here in Edinburgh.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Sameer Bansal*)

To Mom, Pops, and Mamu.

Table of Contents

Chapter 1

Introduction

There is a broad demand for translation services in our daily lives as we travel more, study abroad and conduct business globally. For use-cases such as negotiating complex treaties between world governments, diplomats rely on human interpreters to provide high-quality near real-time translation; Figure 1.1 shows the earpiece device through which UN members can listen to speech translated into their preferred language.[1] For more general use, we do not always require the quality-of-service a human translator provides, and instead are content with using automated translation tools such as Google Translate — which is used to translate around 100 billion words a day, even though their output might contain errors.[2] For many of us, this implies typing out a query into a browser (or an app) to obtain the translated text. This interface poses two main problems for certain languages: using text as input is impossible for languages without a written form; and illiteracy rates remain high in the developing world. In such scenarios, applications which can accept speech as input offer clear benefits over those which just accept text.

The complexity of building automated translation systems is high, and as a result only around a hundred languages are currently supported by publicly available systems.[3] In fact there are many languages which are spoken by millions of people in the developing world, but for which there is no speech translation support. These include: Punjabi, spoken in India by around 31 million speakers; Javanese, spoken in Indonesia by around 70 million speakers; and Mboshi, a Bantu language spoken (without a written form) in

[1] UN Interpretation Service: un.org/Depts/DGACM/interpretation.shtml

[2] Google Q2, 2018 earnings call: abc.xyz/investor/static/pdf/2018_Q2_Earnings_Transcript.pdf

[3] translate.google.com/intl/en/about/languages

Figure 1.1: Earpiece which provides real-time translation to United Nations members. Photo credit: (left) BBC news, (right) twitter user @ zeldman.

the Republic of Congo, with around 160,000 speakers.[4]

Lack of automated translation services can have important real-world consequences. Haiti was hit by an earthquake in 2010, following which a massive international rescue effort was organized. To help victims reach out for help, a text-message service was set up. Unfortunately this quickly created a communication bottleneck, as many rescue workers who were from the US military could not understand these messages which were written in Haitian Creole. At the time, Haitian Creole was not supported by Google Translate, and therefore the burden of translation fell onto human experts. Munro (2010) describe how an innovative solution was quickly and successfully implemented, where a global network of volunteers from the Haitian diaspora was put into action to translate these text messages and channel them back to the rescue workers, with a turn-around time of less than 10 minutes.

From a technology perspective, are we better prepared to handle similar situations today? Arguably yes for processing text messages, but this leaves out people who may be unable to read and write, and would instead prefer to communicate via speech. Modern messaging tools such as WhatsApp allow users to share voice snippets and images (Figure 1.2); and we expect many people to use these during a crisis. Though Google Translate now supports text-to-text translation from Haitian Creole to English, it cannot currently translate speech. Therefore, we find ourselves in a position where technology may again fail to help. This will also be true for many other countries such as India and Indonesia.

[4] ethnologue.com, language codes: *pan (Punjabi), jav (Javanese), mdw (Mboshi)*

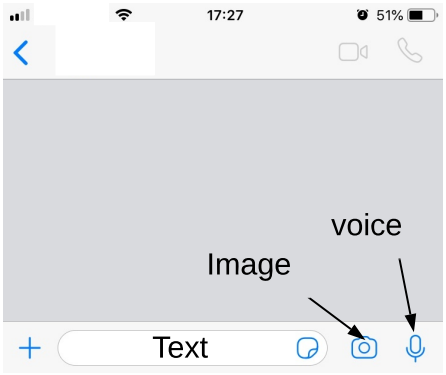


Figure 1.2: WhatsApp messaging interface. Supports image and text messages in addition to text.

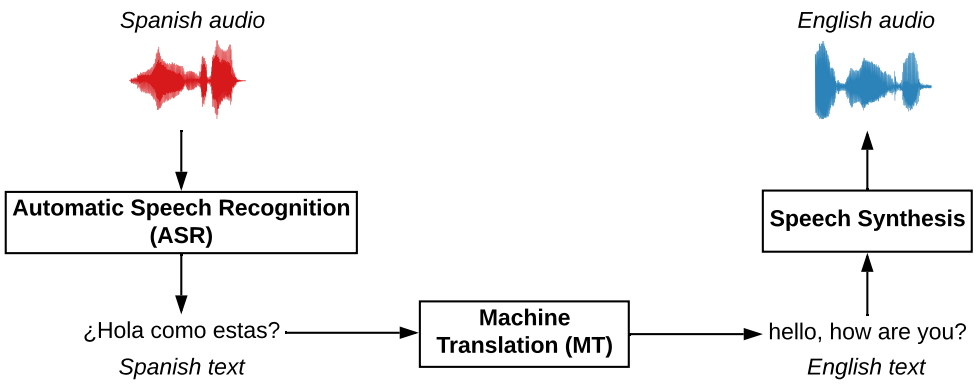


Figure 1.3: Speech-to-speech translation using a cascade of ASR, MT, and Speech Synthesis systems. In this example Spanish and English are the source and target languages respectively.

1.1 Research setting

Figure 1.3 shows a conventional pipelined system for speech-to-speech translation. Source language speech is converted into source language text using automatic speech recognition (ASR); followed by machine translation (MT) to convert source language text to target language text; and finally speech synthesis to generate target language speech. These require extensive training resources: thousands of hours of paired source language speech and text to train ASR; paired target language speech and text for speech synthesis; and millions of lines of bilingual text to train MT.

Such large training sets are available for only a tiny fraction of the world's highest-resource languages: around 100 out of the estimated 7000 languages currently spoken

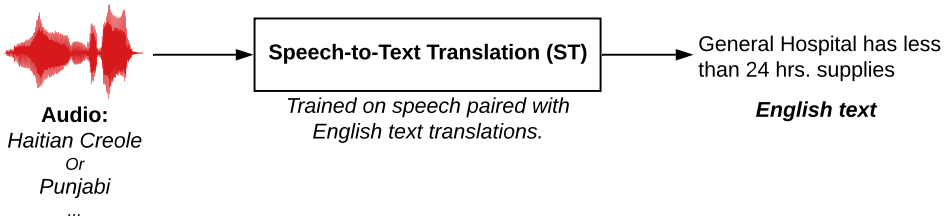


Figure 1.4: Speech-to-text translation (ST) hypothetical example. Speech data is directly translated to English text. Example text shown here is from Munro (2010) and is an actual text message exchanged during the Haiti earthquake relief operations.

around the world (Ethnologue, 2019b).[5] The remaining languages, around 6900, can be broadly classified as *low-resource* based on the number of hours of labeled speech data available, typically in the tens of hours compared to hundreds or thousands for high-resource languages. An estimated 3000 languages, or 40%, out of these do not have a standard written form and are also considered endangered, with fewer than 1000 active speakers remaining (Ethnologue, 2019c,a). For these unwritten languages, the pipelined approach for speech translation is impractical as we cannot obtain source language text to train ASR systems.

In this thesis, we investigate whether we can directly translate source language speech into target language text — referred to as speech-to-text translation (ST), as shown in Figure 1.4. Revisiting the crisis-relief scenario, such a system would allow rescue personnel to quickly parse incoming audio. To build such a system, we wish to use only speech data paired with text translations as training data.[6] Our work specifically focuses on unwritten languages, as we do not use source language text to build ST. Henceforth, we use the term *low-resource* to imply that limited amounts (tens of hours) of training data is available and the source language has no written form or source text is not available.

Pairing speech with translations is a convenient option for languages without a written form (Bird et al., 2014; Blachon et al., 2016; Adda et al., 2016; Besacier et al., 2006), and such datasets are increasingly being made publicly available. Godard et al. (2018)

[5]There is no official resource which provides a categorization of all languages as either high-resource or low-resource. We assume that languages supported by tools such as Google Translate are high-resource. Estimates for the total number of languages and how many are written, unwritten and endangered can vary.

[6]If source language text is available, we may use it for analysis only.

released a corpus of Mboshi speech paired with French text translations. For endangered languages such as Ainu (spoken in Japan with 10 native speakers alive as of 2007), and Arapaho (spoken in Wyoming, United States, with about 1000 native speakers), audio data has been collected with accompanying English text translations to aid conservation efforts.[7]

Although working on a true low-resource language corpus would be ideal, for the majority of our experiments, we use the Fisher Spanish speech corpus (Graff et al., 2010), and its accompanying English translations (Post et al., 2014) to build Spanish-English ST models in simulated low-resource conditions. Spanish is not an endangered or a low-resource language, but we chose to work with this dataset as it provides us with over a hundred hours of translated speech data, giving us flexibility in experimental design. We can therefore simulate a wider space of low-resource settings; for example, using between 5 and 50 hours of labeled data during training. The speech data consists of unscripted conversations recorded in realistic noise conditions, with multiple speakers (with 80 female, and 50 male) and dialects; and the English text translations were collected through crowdsourcing. The dataset is also closer to our settings of interest compared to "clean" speech — read (audiobooks) or synthesized audio — which is typically bereft of properties of natural speech such as disfluencies and speaker-to-speaker variations.

A speech-to-speech translation system is an attractive end goal with several important applications. These include: facilitating conversation between humans; translating recorded audio (or audio-visual) content and disseminating it to a wider audience; and translating automated voice announcements commonly used in airports/train stations. In these examples, the source audio may be produced by a human or a machine, but the target speech is primarily intended for human consumption. But there are several useful applications for speech-to-text systems, especially in scenarios where users interact with an electronic device — such as an Amazon Echo smart speaker — using speech as input.[8] Here, speech is typically converted to text which is then further processed to provide services such as information retrieval, clustering audio documents, or question/answering. We explore whether ST can be used to build similar services for low-resource languages, where ASR is not feasible.

[7] Ainu: ainucorpus.ninjal.ac.jp/en

Arapaho: colorado.edu/center/csilw/language-archives/arapaho-narratives

[8] Product details can be found on amazon.com

1.2 Contributions and outline

The central goal of this thesis is to shed light on the following question:

> Given only a few hours of speech paired with text translations, can we build a "useful" ST system?

We answer this question affirmatively and show that neural sequence-to-sequence models are an effective and convenient method for building useful speech technology for low-resource languages.

Our work can be broadly divided into three parts: building baseline low-resource ST systems (Chapters 3 and 4); exploring methods to improve baseline translation performance (Chapter 5); and demonstrating of the utility of the translations produced (Chapter 6).

The outline of the thesis is as follows:

Chapter 2. In this chapter, we present the background materials for the rest of thesis. We provide an overview of the traditional speech features used as input to build ST systems. Next, we review studies from the zero-resource speech processing community which attempts to learn from unlabeled audio data alone. In particular, we review the task of unsupervised term discovery (UTD) which involves detecting and clustering acoustically similar patterns in speech data and is one of the most well-developed areas. We then review neural architectures for sequence-to-sequence modeling which have been applied for high-resource text-to-text translation and speech-to-text transcription/translation.

Chapter 3. An appealing approach for building a low-resource ST system would be to take an end-to-end neural ASR model architecture, which has been demonstrated to work well in high-resource monolingual settings (Chan et al., 2016), and train it in low-resource cross-lingual settings.[9] However, previous attempts achieved poor results.

In this chapter we explore an alternative: can we build off the work of zero-resource speech research and expand it to low-resource settings? We test whether a recently released state-of-the-art UTD algorithm can be utilized to build an ST system. We conduct experiments on around 10 hours of Spanish-English data and find that the UTD

[9] Here, end-to-end implies that we directly translate speech input to target language text, without going through the intermediate ASR step, as done in a pipeline approach.

algorithm struggles to discover patterns across speakers in our conversational speech corpus recorded in realistic noise conditions. The translations produced by our method achieve poor precision/recall scores and our takeaway is that significant improvements will be required to make this method practically useful. We next switch to a neural approach for ST based on the promising work by Weiss et al. (2017).

Chapter 4. We conduct experiments using a neural model for ST. We build our own software pipeline, and use it to study the impact of training data size on translation quality. We try and answer the question: how many hours of labeled speech data are required to train neural models to make "useful" predictions? We show that using just 20 hours of Spanish-English ST data, our model achieves a BLEU score of 10.8. For comparison, the state-of-the-art model trained on 160 hours of data achieves a BLEU score of 47.3 on the same dataset. We compute additional evaluation metrics, and discover that although the BLEU score of our model is low, the predicted translations achieve a word-level unigram precision/recall of around 40%, compared to 70% for the state-of-the-art.[10] This implies that the predicted text contains 40% of the tokens in the reference human text, many of which we expect to carry meaning and are not just stopwords, and can therefore still be useful in low-resource scenarios.

We also show that models struggle to learn when less than 20 hours of data is used, and are outperformed by a naive baseline model which just predicts the top 10 most frequent words in the training set for each test set utterance.

Chapter 5. In this chapter we explore methods to improve the translation performance of our ST models, which do not require labeling additional speech data in the low-resource language. We show that pre-training the ST model on ASR data from a high-resource language helps improve translation performance.

For our Spanish-English experiments, we pre-train the neural model on 300 hours of English ASR data, and then fine-tune the parameters on 20 hours of Spanish-English ST data, and observe that the BLEU score improves from 10.8 to 19.9. We also find that pre-trained model trains faster on ST, surpassing the BLEU score of 10.8 in around 2 hours of training (time), which takes the baseline model around a day of training to achieve. This can be useful in disaster-recovery scenarios where an ST system has to be bootstrapped in quick time.

[10] The precision and recall scores are very similar, and therefore we report a single value.

Pre-training also helps when the high-resource language is different from both the source and target ST languages. For example, we show that pre-training on French ASR improves Spanish-English ST.

Chapter 6. Using our pre-training method, we were able to improve the translation performance of our ST models. However, the BLEU scores still remain low, and a review of the predicted translations shows that they are mediocre. Can these translations still be "useful"? It has long been recognised that there are good applications for bad translations (Church and Hovy, 1993). In this chapter we consider two such applications:

1. **Classifying speech utterances by topics.** We show that the noisy translations produced by our ST models, are still good enough to correctly predict the topic of discussion in 1-minute long speech utterances. This can be useful to triage large volumes of incoming speech data by topics of interest.
2. **Cross-lingual keyword spotting.** Using our ST output, we build a system to retrieve speech utterances using cross-lingual keyword queries. This can be useful in scenarios where a human operator is searching for specific/urgent keywords, such as *medical* or *help*, in large volumes of audio.

Chapter 7. We summarize our main findings from the previous chapters and suggest future research directions.

Chapter 2

Background

In this chapter, we present relevant background for the rest of the thesis. We discuss how speech data is typically represented for use in computational models. We then review previous studies which attempt to learn from audio alone, referred to as *zero-resource* or *unsupervised* speech processing. Next, we introduce the deep learning model widely used for the task of text-to-text translation (MT) and serves as the foundation for the end-to-end neural ST models we develop in this work.

2.1 Speech features

This section describes how speech data is typically preprocessed before being used in computational models. For a speech utterance, the recording process involves sampling the natural speech signal at regular time intervals (sampling rate). For example, in our Spanish telephone speech dataset (*fisher-spanish*), audio is recorded with a sampling rate of 8 KHz, implying that a continuous waveform of 1 second would be encoded as 8000 equally spaced real valued measurements. Figure 2.1 (a) shows the raw audio (with a duration of 1.57 seconds) recorded for the Spanish speech utterance *oh mi nombre es ricardo*, translated to English text as *oh my name is ricardo*.

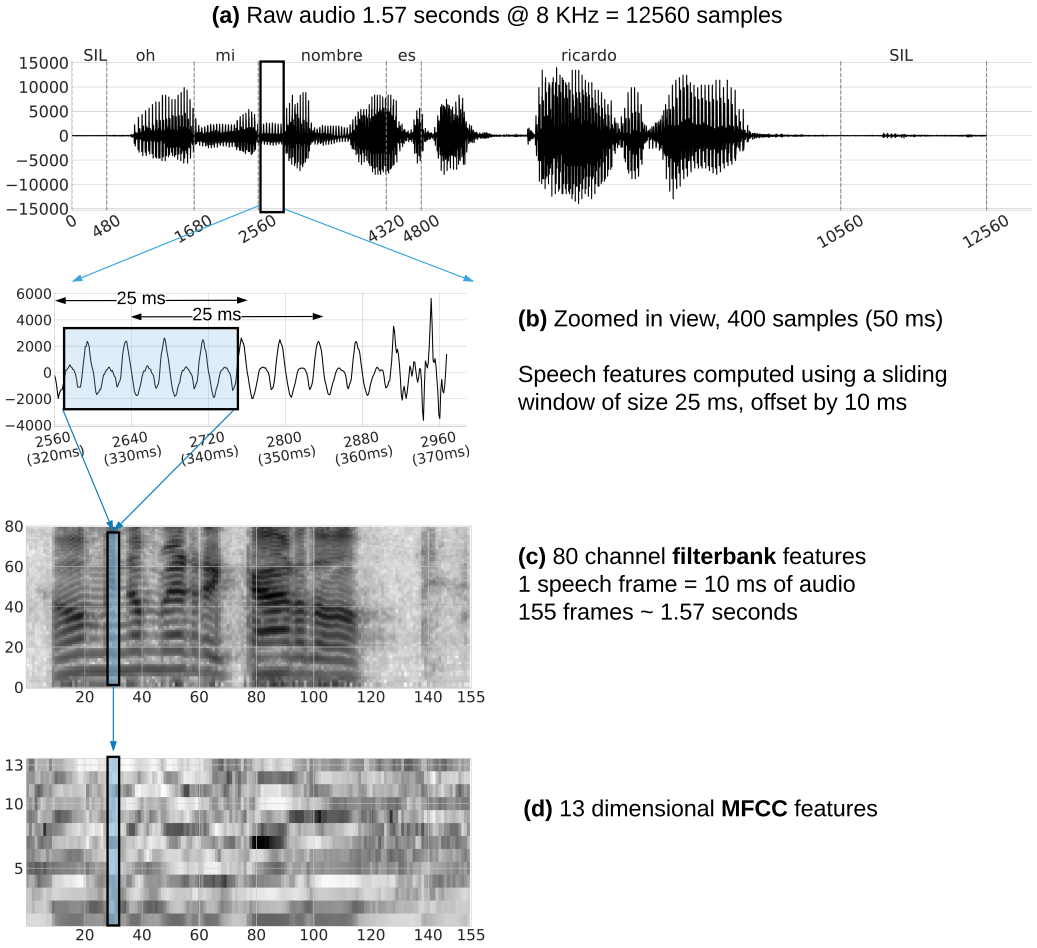


Figure 2.1: Speech features. x-axis is time; y-axis denotes amplitude for raw audio, and frequency (or channels) for speech features. **Raw audio** (a converted into **Fbanks** (c), which are converted into **MFCCs** (d).**SIL** denotes silence phase in the speech utterance.

The standard approach (Davis and Mermelstein, 1980) in building speech-to-text systems involves converting the raw audio samples into Mel-scale filter bank feature vectors (*Fbanks*) which are then converted to Mel-frequency cepstral coefficient features (*MFCCs*) as shown in Figure 2.1 (c) and (d). Each Fbank or MFCC vector, referred to as a speech frame, is computed over a 25 millisecond (ms) sliding window of audio with a hop of 10 ms, as shown in Figure 2.1 (b).[1] The window size is selected on the basis that human speech does not show much variations at this scale. Each Fbank vector has 80 dimensions; and each MFCC vector has 13. These dimensions are referred to as *channels*. [2] The speech frames are then concatenated in order to form the speech features for the audio input. For our example speech utterance, we have 1.57 seconds of audio, captured as 12560 raw audio samples (R^{1}), converted to Fbanks ($R^{155 \times 80}$); which are then converted to MFCCs ($R^{155 \times 13}$).[3]

We use both Fbanks and MFCCs as speech features. The details on how these are computed are not discussed in this work, but it's important to note that Fbanks are closer to raw audio than MFCCs as can be seen in Figure 2.1. To compute speech features, we use the Kaldi toolkit (Povey et al., 2011) with default settings. For a more detailed description and comparison of speech features we recommend Mohamed (2014) and Renshaw (2016).

Although there has been recent promising work on using raw audio (as shown at the top of Figure 2.1) directly for speech-to-text (Sainath et al., 2015; Palaz et al., 2015; Golik et al., 2015; Bhargava and Rose, 2015), we do not explore it in this work. Given vast quantities of training data — hundreds of hours of speech — it should be possible to learn salient features from raw audio directly using deep learning methods. However, in low-resource settings of our interest, learning the complex series of transformations required to produce features as effective as MFCCs (or Fbanks) may be difficult. Therefore, we chose to rely on speech features which have been widely used over the past several years to produce state-of-the-art speech-to-text systems.

[1] Other window sizes can also be used, but the default is 25 ms width with a 10 ms hop.

[2] These dimensions are configurable, but typically 40 to 80 are used for Fbanks, and 13 for MFCCs.

[3] For 1.57 secs, we only compute 155 speech frames, and not 157. This is because for the 155th speech frame, the window size of 25 ms extends to the end of the audio input.

2.2 Unsupervised methods

We use the term unsupervised learning to refer to methods which attempt to learn given audio data only. This includes methods that discover acoustic patterns represented in symbolic form, but can also refer to methods for finding continuous features capturing linguistically meaningful information.

To build ST systems, at a minimum we require audio data from the source language and text data from the target language. However, collecting audio data for spoken languages is considerably easier than transcribing and/or translating it, and it is reasonable to expect a scenario where there are several hours of recorded audio data available in an endangered language, waiting to be labeled or to be utilized in some other way. Can we build useful systems from audio only? This question has been the focus of work by the *zero-resource* speech research community.[4]

A core focus area of the zero-resource community has been *unsupervised subword modeling*, where the aim is to learn frame-level feature representations which capture linguistic properties better than features like MFCCs and Fbanks (Badino et al., 2015; Thiollière et al., 2015; Kamper et al., 2015; Renshaw et al., 2015). Several other tasks have also been considered, including automatic discovery of subword units (Lee and Glass, 2012; Siu et al., 2014); query-by-example (Zhang et al., 2012; Metze et al., 2013; Levin et al., 2015); topic based clustering of audio documents (Dredze et al., 2010); full segmentation and clustering of the audio into word-like units (Walter et al., 2013; Lee et al., 2015; Räsänen et al., 2015; Kamper et al., 2016); and unsupervised term discovery (Park and Glass, 2008; Zhang and Glass, 2010; Jansen and Van Durme, 2011).

Versteegh et al. noted that it was difficult to compare the systems being developed by various research groups as there were no common training datasets and evaluation methods being used. To address this, they introduced the Zero Resource Speech Challenge at Interspeech 2015 (Versteegh et al., 2015). The challenge focused on two zero-resource tasks: unsupervised subword modeling and spoken term discovery. The evaluation code and datasets were made available to the participating teams and a summary of the submitted systems was published in Versteegh et al. (2016). There have been two further iterations of the Zero Resource Speech Challenge in 2017 and 2019 (Dunbar et al., 2017, 2019), and project details and resources are available

[4]The term *zero* implies that there are no other resources (such as transcripts) available apart from audio.

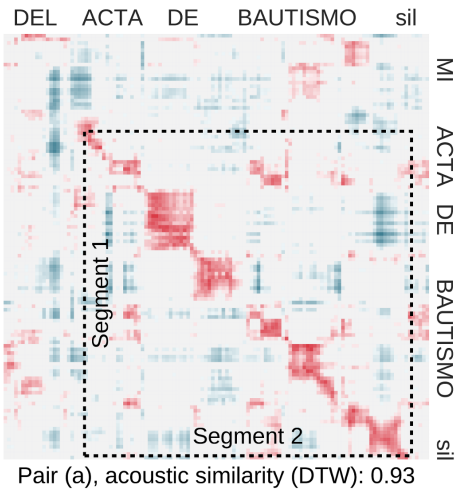


Figure 2.2: Acoustic similarity for utterance pairs. Dark/Red regions indicate strong match, Light/Blue indicate weak match. Dotted box marks the matching segments returned by UTD.

at `http://zerospeech.com/`.

In this work, we review the task of *unsupervised term discovery* (UTD), which aims to identify and cluster repeated word-like units from audio and is one of the most well-developed areas. It has also been used to build ST systems (Bansal et al., 2017; Anastasopoulos et al., 2017), and we discuss these in Chapter 3.

UTD systems, also referred to as *spoken term discovery* systems, search for pairs of audio segments that are similar, typically measured according to their dynamic time warping (DTW; Sakoe and Chiba, 1978) distance. Functionally, DTW can be considered as the continuous counterpart to Levenshtein (or edit) distance (Levenshtein, 1966) over discrete symbols. In other words, the DTW distance between two continuous vector sequences which may vary in length, is the optimal alignment between them which respects the temporal ordering. Figure 2.2 shows an acoustic pattern detected by a UTD system given two Spanish utterances which share the phrase *acta de bautismo*. This task is inherently quadratic in the input size, and early systems (Park and Glass, 2008; Zhang and Glass, 2009) were prohibitively slow. Jansen and Van Durme (2011) introduced a quasilinear time algorithm, implemented as part of the open-source Zero Resource Toolkit (ZRTools).[5,6] In addition to being computationally efficient, it is also the only

[5]Quasilinear complexity implies $O(n \log n)$.

[6]ZRTools are available at https://github.com/arenjansen/ZRTools

freely available UTD system we know of.

In its first pass, ZRTools uses an approximate randomized algorithm and image processing techniques to extract potential matching segments. Image processing is used based on the intuition that if we plot the cosine similarity between every frame of the input feature vector representation (e.g. MFCCs), any repeated segments in the pair of utterances will show up as diagonal line patterns. Figure 2.2 illustrates this, showing a clear diagonal pattern corresponding to similar words in two utterances. In its second pass, ZRTools computes a normalized DTW score over potential matches to extract the final output. It returns segment pairs longer than a minimum duration (we used the recommended value of 500ms) along with their DTW alignment score (normalized to be between 0 and 1, with higher scores indicating greater similarity). These word-like or phrase-like segments can then be used for downstream tasks like keyword search and topic modeling (Park and Glass, 2008; Zhang and Glass, 2009; Dredze et al., 2010). Full details of the system can be found in Jansen and Van Durme (2011).

2.3 Neural sequence to sequence models

We predominantly use a neural sequence-to-sequence with attention model (*seq2seq*) for ST (Chapters 4, 5, 6). This type of model was originally introduced by Bahdanau et al. (2015) for the task of machine translation (MT), and was an extension of previous work on encoder-decoder models (Sutskever et al., 2014; Cho et al., 2014), which did not include attention. As the name implies, the model was designed for tasks where the input and the output are sequential. For MT, the input and output is text, represented as a sequence of discrete symbols (words, or characters or subword units). The key aspects of this model are the use of recurrent neural network (RNN) units such as long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997) to process sequential input and output, and the attention mechanism to learn complex alignments between the source language and the target language text. *Seq2seq* models have since found widespread use in other tasks such as image (Xu et al., 2015) and video captioning (Venugopalan et al., 2015), where the input is a sequence of pixel values and the output is text. Closer to our task of ST, they have also been applied for end-to-end ASR (Chan et al., 2016), where the input is an audio sequence. Next, we discuss changes required to adapt the encoder component of a *seq2seq* model to accept speech as input in place of text.

For text input, a *seq2seq* RNN encoder typically consists of an embedding layer which feeds into a stack of bi-directional LSTM layers (bi-LSTM). Figure 2.3 shows the encoding process for the example Spanish text: *oh mi nombre es ricardo*.[7] The text is composed of 5 word units, which yields 5 hidden states as the output of the RNN. The RNN encoding procedure has a time complexity of $O(N)$, where N is the number of input vectors, in this case the word embeddings. If the text were to be processed as character level input, we would get 23 character level embeddings as input to the RNN, increasing the encoding compute time. But these remain within reasonable compute expectations for a modern GPU.[8] However, the speech features for this utterance consists of 155 MFCC vectors, vastly increasing the number of input vectors to the RNN, and subsequently increasing the time taken to encode. And it is common to have longer utterances which vary in duration between 5 to 20 seconds, or 500 to 2000 speech frames, in most datasets. The RNN encoding process for such large sequences is computationally expensive. For example, Chan et al. (2016) conducted experiments on a large speech dataset ($>$2000 hours) with speech features computed over a 10ms window. In a set of experiments, they encoded these features directly without compressing or controlling the sequence length for long audio utterances. Therefore, for a 5 second long speech utterance, the RNN encoder would receive as input around 500 speech frames, and correspondingly produce 500 hidden states. Under these conditions, they found that models converge slowly, taking over a month to converge. Longer training times may be justified if the evaluation results are improved, but they found that learning from such a large number of speech frames actually hurts end-task performance with models performing poorly on evaluation sets.

To address the issue of long input sequences for speech data, Chan et al. (2016) proposed a pyramidal bi-LSTM encoder architecture for ASR. In this architecture, each successive RNN layer after the bottom one which accepts speech features, reduces the number of time steps by a factor of 2. As shown in Figure 2.4, by stacking 2 additional bi-LSTM layers above the bottom layer, we can reduce the total number of RNN output states by a factor of 4, (from 155 to 38 in our example speech utterance), making it easier for the model to learn the desired task. Note that as the data is already continuous, an embedding layer is not required. Chan et al. (2016) used a stack of pyramidal layers to reduce the input sequence length by a factor of 8. They found that reducing the number of input frames resulted in a faster training time, with models converging in

[7] All layers have 256 hidden units or dimensions in this example. We show a 3 layer RNN.

[8] Titan X and equivalent.

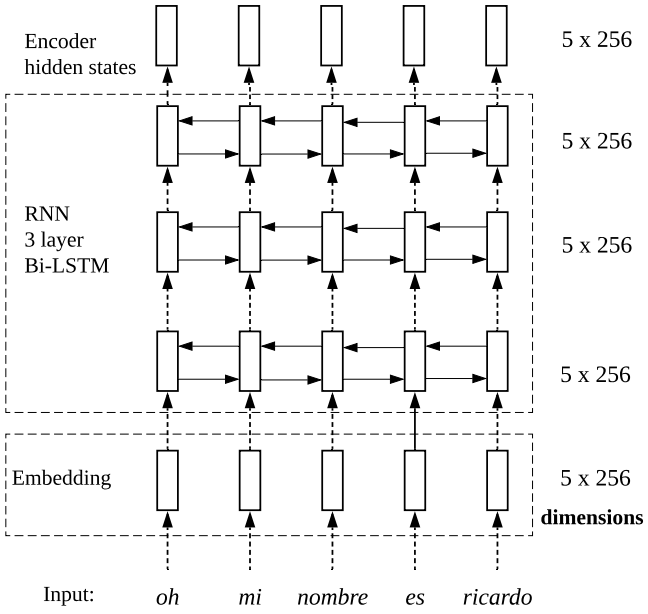


Figure 2.3: Typical Machine Translation (MT) encoder architecture used in a *seq2seq* model. Input is Spanish text.

around two weeks compared to over a month when using all the frames, and importantly, this also helped the model achieve a better word-error-rate (WER) on the evaluation set. Zhang et al. (2017) further improved upon the pyramidal bi-LSTM encoder, by including a convolutional neural network (CNN) layer component. The use of CNNs was inspired by their successful application in computer vision tasks, and previous work demonstrated their utility for speech recognition as well (Abdel-Hamid et al., 2013; Sainath et al., 2013; Chan and Lane, 2015; Sercu and Goel, 2016; Sercu et al., 2016). In this new architecture, speech features (Fbanks) are first fed into a stack of CNN layers. The first and second CNN layers use a stride of 2, thereby reducing the overall number of time steps by 4; therefore the pyramidal method to reduce time steps is no longer required, and a vanilla bi-LSTM encoder is used. Zhang et al. (2017) state that reducing the number of time steps, using CNNs with a stride of 2, is important for computational reasons. We do not explore this in detail in our own work, but refer the reader to the work of Sercu and Goel (2016), who use CNN-based models for ASR and test configurations with and without time pooling.

Whereas Chan et al. (2016) focused only on ASR, since then their pyramidal bi-LSTM encoder has also been applied for ST by Duong et al. (2016) and Berard et al. (2016). However, Duong et al. (2016) used the model only to align speech and text, and did not

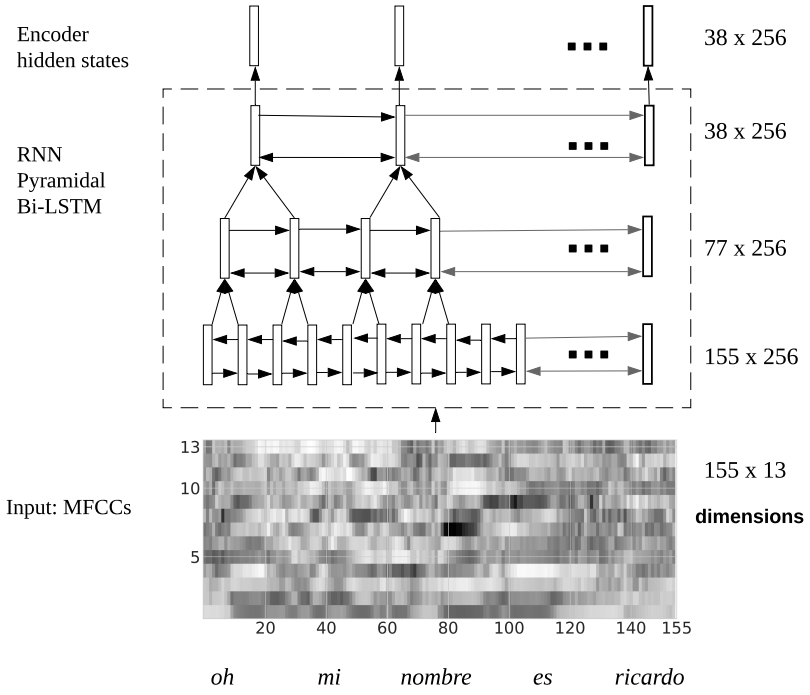


Figure 2.4: Pyramidal bi-LSTM encoder architecture. Input is Spanish speech.

make any translation predictions on test data. Berard et al. (2016) did make predictions, but their experiments used synthesized speech data, which although promising, may be a poor indicator for performance on real speech data.

The work by Weiss et al. (2017) was the first to show the effectiveness of *seq2seq* models for ST on a real speech dataset. They used a hybrid CNN and RNN model, adapted from the work of Zhang et al. (2017), for Spanish-English ST and achieved state-of-the-art performance on the Fisher Spanish telephone speech dataset (Graff et al., 2010; Post et al., 2014). We chose the Weiss et al. model to serve as the foundation of our neural model based ST approach (explored in Chapter 4). As we are using the same dataset, this provides us with a reference set of results to compare the performance of our own ST models trained under simulated low-resource settings. The authors also shared their model predictions on the evaluation sets, allowing us to use them for a detailed quantitative and qualitative analysis.

Since we began our work, several newer architectures have been proposed. In contrast to CNN and/or RNN based *seq2seq* models, Vaswani et al. (2017) proposed the *Transformer* model for MT, which relies primarily on the attention mechanism, doing away with both convolutional and recurrent model components. The model is parallelizable to a higher degree, especially as a result of eliminating the RNN procedure, where the

input needs to be processed in sequence. Vaswani et al. (2017) show that the model not only trains faster, but also achieves state-of-the-art performance on an MT task. The model architecture is rapidly being adopted for a wide variety of tasks, and was also used to train Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019). Although the encoding time for this model architecture is faster than the RNN-based model architectures, as there are $O(1)$ sequential operations compared to $O(N)$ for RNN-based, the number of trainable parameters is also much higher. For example, BERT has greater than 100 million parameters, compared to around 10 million for the Weiss et al. (2017) model. In practice, this large number of parameters means that models may not fit in a regular Titan X (or equivalent) GPU unless the mini-batch size is reduced. And, for long speech utterances, the mini-batch size would have to be reduced further. For example, the BERT model architecture, currently used to encode text input, limits the longest sequence length to 512 tokens, which if used directly can only be used to encode short speech utterances (around 5 seconds). According to the official implementation for the basic BERT model, the recommended mini-batch size is 6 at the maximum sequence length. This would result in much slower training, which could be addressed by training on several GPUs.[9] Therefore, for now, some kind of length pruning for speech utterances might still be required if models have to be trained within reasonable compute.

Finally, Chen et al. (2018) propose a hybrid approach combining aspects of various neural architectures: CNNs, RNNs and Transformer, and this is a promising direction for future work. New (and improved) neural models will continue to be released in the near future, but for the purposes of our research we use the CNN/RNN neural architecture only. By keeping the model architecture consistent, we can compare whether a method, such as transfer-learning (explored in Chapter 5), improves translation performance over a baseline.

2.4 Speech recognition in low-resource scenarios

While our work focuses on low-resource ST, improving the performance of ASR systems in low-resource scenarios has long been an active area of research. For example, a common scenario in ASR is the *lightly-supervised* setting, where there exists mono-

[9]GPU memory guidelines for BERT model: `https://github.com/google-research/bert/blob/master/README.md#out-of-memory-issues`

lingual transcribed speech data along with unlabeled speech data. The quantity of labeled data is typically smaller than the unlabeled data. In this setting, *semi-supervised learning* is often used, where an ASR system is first trained on the labeled data, which is then used to predict text for the unlabeled speech. The speech data paired with predicted text is then used to further train or refine the ASR system (Kemp and Waibel, 1999; Lamel et al., 2002). Lamel et al. (2002) further expand this setting to include scenarios where instead of additional unannotated speech, a corpus of speech paired with noisy transcriptions is available. They provide the example of broadcast news data, where the closed captions or subtitles may not directly correspond to the audio. In this setting, the noisy annotated transcripts are used to improve the quality of the predicted transcripts from the ASR system, which are then used for further training (Chan and Woodland, 2004; Laurent et al., 2016; Fainberg et al., 2019). This semi-supervised learning method, which combines learning from annotated and unannotated speech, is a promising avenue for future work in low-resource ST, but one which we do not explore in our own work.

A different method which has been widely explored to improve low-resource ASR is multi-lingual training (Schultz, 2002; Niesler, 2007; Ghoshal et al., 2013; Huang et al., 2013; Heigold et al., 2013; Deng et al., 2013; Vu et al., 2012; Thomas et al., 2012; Cui et al., 2015; Alumäe et al., 2016; Yuan et al., 2016). Here, transcribed speech data from different languages is exploited to improve the ASR performance of a target language. We explore this method in our own work to improve ST and discuss it in detail in Chapter 5.

2.5 Review and next steps

In this chapter we provided an overview of traditional speech features, which we use as input to build ST systems. We also briefly reviewed unsupervised or zero-resource learning using speech data only and focus on the UTD task. In Chapter 3 we build a speech translation system using a publicly available, state-of-the-art UTD software library. Finally, we reviewed contemporary neural sequence-to-sequence models for directly translating speech into target language text, and use them as a basis for our models in Chapters 4 and 5.

Chapter 3

UTD based Speech-to-Text Translation

3.1 Introduction

In this chapter, we describe our first attempt at building an ST system trained and tested on a realistic speech corpus. As we are interested in low-resource scenarios, it is natural to ask whether methods developed by the zero-resource community (described in Chapter 2) can be exploited. To test this, we use a recently released state-of-the-art algorithm for UTD (Jansen and Van Durme, 2011) to build a traditional pipeline system for Spanish-English ST trained on around 10 hours of data.

As an alternative to using UTD to construct an intermediate representation for low-resource speech, we could have chosen to use a phone recognizer trained on a high-resource language (Stahlberg et al., 2014). For example, Wilkinson et al. (2016) used an English phoneme recognizer to convert Mandarin speech into discrete units. However, they used synthesized speech generated from Mandarin text in their experiments. Recently, Salesky et al. (2019a) did conduct experiments on real speech data. They built a Mboshi-French ST system and processed Mboshi speech using an English phoneme classifier. We do not explore this method in our own work and refer readers to the work carried out by Salesky et al. (2019a) instead.

3.2 Paper: Towards speech-to-text translation without speech recognition

Publication status. This work was published in EACL 2017.

Contributions. The main idea for this paper was originally proposed by Adam Lopez and was developed further by all the co-authors. The co-authors also provided regular feedback on all results and helped identify areas for improvement. Each co-author also played a key role in the publication writing process.

My individual contributions in this work were the experimental setup; running ZRTools on our speech corpus; and developing the prediction and evaluation framework.

Towards speech-to-text translation without speech recognition

Sameer Bansal[1], *Herman Kamper*[2], *Adam Lopez*[1], *Sharon Goldwater*[1]
[1]School of Informatics, University of Edinburgh
[2]Toyota Technological Institute at Chicago, USA
{sameer.bansal, sgwater, alopez}@inf.ed.ac.uk, kamperh@gmail.com

Abstract

We explore the problem of translating speech to text in low-resource scenarios where neither automatic speech recognition (ASR) nor machine translation (MT) are available, but we have training data in the form of audio paired with text translations. We present the first system for this problem applied to a realistic multi-speaker dataset, the CALLHOME Spanish-English speech translation corpus. Our approach uses unsupervised term discovery (UTD) to cluster repeated patterns in the audio, creating a *pseudotext*, which we pair with translations to create a parallel text and train a simple bag-of-words MT model. We identify the challenges faced by the system, finding that the difficulty of cross-speaker UTD results in low recall, but that our system is still able to correctly translate some content words in test data.

1 Introduction

Typical speech-to-text translation systems pipeline automatic speech recognition (ASR) and machine translation (MT) (Waibel and Fugen, 2008). But high-quality ASR requires hundreds of hours of transcribed audio, while high-quality MT requires millions of words of parallel text—resources available for only a tiny fraction of the world's estimated 7,000 languages (Besacier et al., 2014). Nevertheless, there are important low-resource settings in which even limited speech translation would be of immense value: documentation of endangered languages, which often have no writing system (Besacier et al., 2006; Martin et al., 2015); and crisis response, for which text applications have proven useful (Munro, 2010), but only help literate populations. In these settings, target translations may be available. For example, ad hoc translations may be collected in support of relief operations. Can we do anything at all with this data?

In this exploratory study, we present a speech-to-text translation system that learns directly from source audio and target text pairs, and does not require intermediate ASR or MT. Our work complements several lines of related recent work. For example, Duong et al. (2016) and Anastasopoulos et al. (2016) presented models that align audio to translated text, but neither used these models to try to translate new utterances (in fact, the latter model cannot make such predictions). Berard et al. (2016) did develop a direct speech to translation system, but presented results only on a corpus of synthetic audio with a small number of speakers. Finally, Adams et al. (2016a; 2016b) targeted the same low-resource speech-to-translation task, but instead of working with audio, they started from word or phoneme lattices. In principle these could be produced in an unsupervised or minimally-supervised way, but in practice they used supervised ASR/phone recognition. Additionally, their evaluation focused on phone error rate rather than translation. In contrast to these approaches, our method can make translation predictions for audio input not seen during training, and we evaluate it on real multi-speaker speech data.

Our simple system (§2) builds on unsupervised speech processing (Versteegh et al., 2015; Lee et al., 2015; Kamper et al., 2016b), and in particular on *unsupervised term discovery* (UTD), which creates hard clusters of repeated word-like units in raw speech (Park and Glass, 2008; Jansen and Van Durme, 2011). The clusters do not account for all of the audio, but we can use them to simulate a partial, noisy transcription, or *pseudotext*, which we pair with translations to learn a bag-of-words translation model. We test our system on the CALLHOME Spanish-English speech translation corpus (Post et al., 2013), a noisy multi-speaker corpus of telephone calls in a variety of Spanish di-

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 474–479,
Valencia, Spain, April 3-7, 2017.

alects (§3). Using the Spanish speech as the source and English text translations as the target, we identify several challenges in the use of UTD, including low coverage of audio and difficulty in cross-speaker clustering (§4). Despite these difficulties, we demonstrate that the system learns to translate some content words (§5).

2 From unsupervised term discovery to direct speech-to-text translation

For UTD we use the Zero Resource Toolkit (ZRTools; Jansen and Van Durme, 2011).[1] ZRTools uses dynamic time warping (DTW) to discover pairs of acoustically similar audio segments, and then uses graph clustering on overlapping pairs to form a hard clustering of the discovered segments. Replacing each discovered segment with its unique cluster label, or *pseudoterm*, gives us a partial, noisy transcription, or pseudotext (Fig. 1).

In creating a translation model from this data, we face a difficulty that does not arise in the parallel texts that are normally used to train translation models: the pseudotext does not represent all of the source words, since the discovered segments do not cover the full audio (Fig. 1). Hence we must not assume that our MT model can completely recover the translation of a test sentence. In these conditions, the language modeling and ordering assumptions of most MT models are unwarranted, so we instead use a simple bag-of-words translation model based only on co-occurrence: IBM Model 1 (Brown et al., 1993) with a Dirichlet prior over translation distributions, as learned by `fast_align` (Dyer et al., 2013).[2] In particular, for each pseudoterm, we learn a translation distribution over possible target words. To translate a pseudoterm in test data, we simply return its highest-probability translation (or translations, as discussed in §5).

This setup implies that in order to translate, we must apply UTD on both the training and test audio. Using additional (not only training) audio in UTD increases the likelihood of discovering more clusters. We therefore generate pseudotext for the combined audio, train the MT model on the pseudotext of the training audio, and apply it to the pseudotext of the test data. This is fair since the UTD has access to only the audio.[3]

[1] `https://github.com/arenjansen/ZRTools`

[2] We disable diagonal preference to simulate Model 1.

[3] This is the simplest approach for our proof-of-concept system. In a more realistic setup, we could use the training audio to construct a consensus representation of each pseudoterm (Petitjean et al., 2011; Anastasopoulos et al., 2016), then use DTW to identify its occurrences in test data to translate.

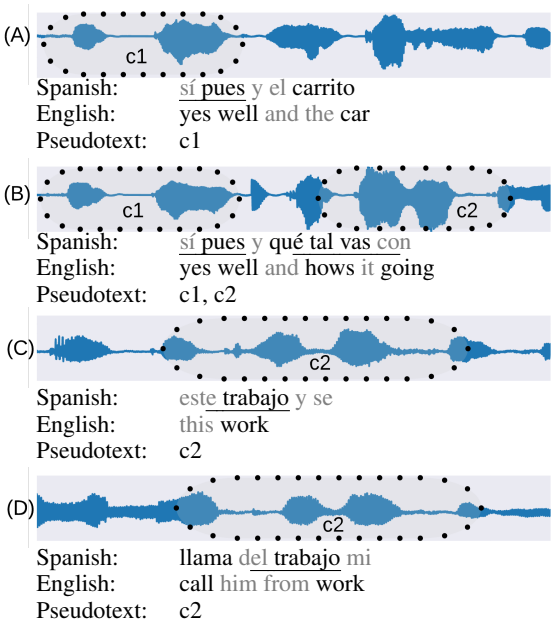


Figure 1: Example utterances from our data, showing UTD matches, corresponding pseudotext, and English translation. For clarity, we also show Spanish transcripts with the approximate alignment of each pseudoterm underlined, though these transcripts are unavailable to our system. Stopwords (in gray) are ignored in our evaluations. These examples illustrate the difficulties of UTD: it does not match the full audio, and it incorrectly clusters part of utterance B with a good pair in C and D.

3 Dataset

Although we did not have access to a low-resource dataset, there is a corpus of noisy multi-speaker speech that simulates many of the conditions we expect to find in our motivating applications: the CALLHOME Spanish–English speech translation dataset (LDC2014T23; Post el al., 2013).[4] We ran UTD over all 104 telephone calls, which pair 11 hours of audio with Spanish transcripts and their crowdsourced English translations. The transcripts contain 168,195 Spanish word tokens (10,674 types), and the translations contain 159,777 English word tokens (6,723 types). Though our system does not require Spanish transcripts, we use them to evaluate UTD and to simulate a perfect UTD system, called the *oracle*.

For MT training, we use the pseudotext and translations of 50 calls, and we filter out stopwords in the

[4] We did not use the Fisher portion of the corpus.

translations with NLTK (Bird et al., 2009).[5] Since UTD is better at matching patterns from the same speaker (§4.2), we created two types of 90/10% train/test split: at the *call level* and at the *utterance level*. For the latter, 90% of the utterances are randomly chosen for the training set (independent of which call they occur in), and the rest go in the test set. Hence at the utterance level, but not the call level, some speakers are included in both training and test data. Although the utterance-level split is optimistic, it allows us to investigate how multiple speakers affect system performance. In either case, the oracle has about 38k Spanish tokens to train on.

4 Analysis of challenges from UTD

Our system relies on the pseudotext produced by ZRTools (the only freely available UTD system we are aware of), which presents several challenges for MT. We used the default ZRTools parameters, and it might be possible to tune them to our task, but we leave this to future work.

4.1 Assigning wrong words to a cluster

Since UTD is unsupervised, the discovered clusters are noisy. Fig. 1 shows an example of an incorrect match between the acoustically similar "qué tal vas con" and "te trabajo y" in utterances B and C, leading to a common assignment to c2. Such inconsistencies in turn affect the translation distribution conditioned on c2.

Many of these errors are due to cross-speaker matches, which are known to be more challenging for UTD (Carlin et al., 2011; Kamper et al., 2015; Bansal et al., 2017). Most matches in our corpus are across calls, yet these are also the least accurate (Table 1). Within-utterance matches, which are always from the same speaker, are the most reliable, but make up the smallest proportion of the discovered pairs. Within-call matches fall in between. Overall, average cluster purity is only 34%, meaning that 66% of discovered patterns do not match the most frequent type in their cluster.

4.2 Splitting words across different clusters

Although most UTD matches are across speakers, recall of cross-speaker matches is lower than for same-speaker matches. As a result, the same word from different speakers often appears in multiple clusters, preventing the model from learning good translations. ZRTools discovers 15,089 clusters in our data, though there are only 10,674 word types. Only 1,614 of the clusters map one-to-one to a unique word type, while a many-to-one mapping of the rest covers only 1,819 gold types (leaving 7,241 gold types with no corresponding cluster).

[5] http://www.nltk.org/

	utterance	call	corpus
Matches	2%	17%	81%
Accuracy	78%	53%	8%

Table 1: UTD matches within utterances, within calls and within the corpus. Matches within an utterance or call are usually from the same speaker.

	utterance split	call split
Oracle	420 (10%)	719 (17%)
Pseudotext	601 (29%)	892 (44%)

Table 2: Number (percent) of out-of-vocabulary (OOV) word tokens or pseudoterms in the test data for different experimental conditions.

Fragmentation of words across clusters renders pseudoterms impossible to translate when they appear only in test and not in training. Table 2 shows that these *pseudotext out-of-vocabulary (OOV)* words are frequent, especially in the call-level split. This reflects differences in acoustic patterns of different speakers, but also in their vocabulary — even the oracle OOV rate is higher in the call-level split.

4.3 UTD is sparse, giving low coverage

UTD is most reliable on long and frequently-repeated patterns, so many spoken words are not represented in the pseudotext, as in Fig. 1. We found that the patterns discovered by ZRTools match only 28% of the audio. This low coverage reduces training data size, affects alignment quality, and adversely affects translation, which is only possible when pseudoterms are present. For almost half the utterances, UTD fails to produce any pseudoterm at all.

5 Speech translation experiments

We evaluate our system by comparing its output to the English translations on the test data. Since it translates only a handful of words in each sentence, BLEU, which measures accuracy of word sequences, is an inappropriate measure of accuracy.[6] Instead we compute precision and recall over

[6] BLEU scores for supervised speech translation systems trained on our data can be found in Kumar et al. (2014).

	source text	gold translation	oracle translation	utd translation
1	cómo anda el plan escolar	how is the school plan going	things whoa mean plan school	school going
2	dile que le mando saludos	tell him that i say hi	tell send best says	say hi
3	sí con dos dientes menos	yeah with two teeth less	two teeth less least	denture yeah teeth
4	o dejando o dejando dos días	or giving or giving two days	improves apart improves apart two days	two days
5	ah ya okey veintitrés de noviembre no	ah yeah okay twenty third of november no	oh ah okay another three fourth november	twenty november

Table 3: Source text (left) paired with translations by humans (gold), oracle, and UTD-based system. Underlined words appear in UTD and the corresponding human translations.

		oracle		pseudotext	
K	metric	utterance	call	utterance	call
1	Prec.	38.6	35.7	7.9	4.0
1	Rec.	33.8	28.4	1.8	0.6
5	Prec.	24.6	23.1	5.9	2.7
5	Rec.	**54.4**	46.4	**5.2**	1.5

Table 4: Precision and recall for $K = 1$ and $K = 5$ under different conditions.

the content words in the translation. We allow the system to guess K words per test pseudoterm, so for each utterance, we compute the number of correct predictions as $corr@K = |pred@K \cap gold|$, where $pred@K$ is the multiset of words predicted using K predictions per pseudoterm and $gold$ is the multiset of content words in the reference translation. For utterances where the reference translation has no content words, we use stop words. The utterance-level scores are then used to compute corpus-level Precision@K and Recall@K.

Table 4 and Fig. 2 show that even the oracle has mediocre precision and recall, indicating the difficulties of training an MT system using only bag-of-content-words on a relatively small corpus. Splitting the data by utterance works somewhat better, since training and test share more vocabulary.

Table 4 and Fig. 2 also show a large gap between the oracle and our system. This is not surprising given the problems with the UTD output discussed in Section 4. In fact, it is encouraging given the small number of discovered terms and the low cluster purity that our system can still correctly translate some words (Table 3). These results are a positive proof of concept, showing that it is possible to discover and translate keywords from audio data even with no ASR or MT system. Nevertheless, UTD quality is clearly a limitation, especially

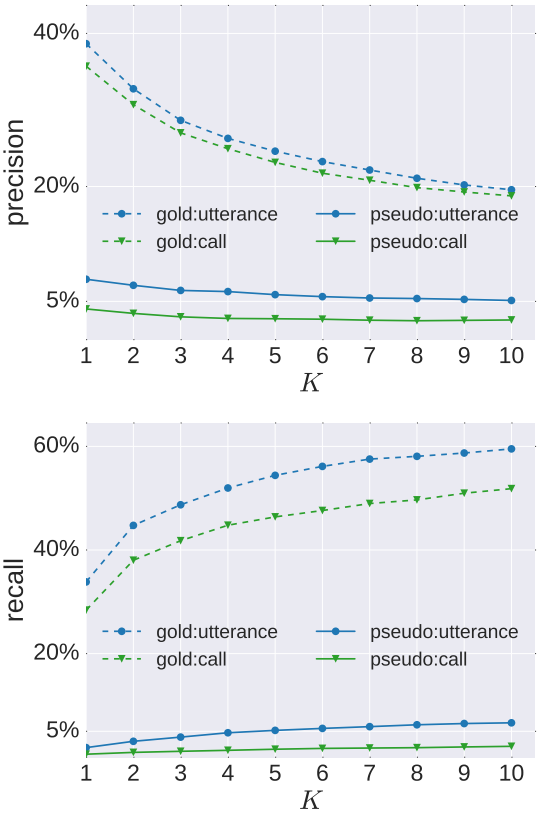


Figure 2: Precision and Recall @K for the call and utterance level test sets.

for the more realistic by-call data split.

6 Conclusions and future work

Our results show that it is possible to build a speech translation system using only source-language audio paired with target-language text, which may be useful in many situations where no other speech technology is available. Our analysis also points to several possible improvements. Poor cross-speaker matches and low audio coverage prevent our system from achieving a high recall, suggesting the of use speech features that are effective in multi-

speaker settings (Kamper et al., 2015; Kamper et al., 2016a) and speaker normalization (Zeghidour et al., 2016). Finally, Bansal et al. (2017) recently showed that UTD can be improved using the translations themselves as a source of information, which suggests joint learning as an attractive area for future work.

On the other hand, poor precision is most likely due to the simplicity of our MT model, and designing a model whose assumptions match our data conditions is an important direction for future work, which may combine our approach with insight from recent, quite different audio-to-translation models (Duong et al., 2016; Anastasopoulos et al., 2016; Adams et al., 2016a; Adams et al., 2016b; Berard et al., 2016). Parameter-sharing using word and acoustic embeddings would allow us to make predictions for OOV pseudoterms by using the nearest in-vocabulary pseudoterm instead.

Acknowledgments

We thank David Chiang and Antonios Anastasopoulos for sharing alignments of the CALLHOME speech and transcripts; Aren Jansen for assistance with ZRTools; and Marco Damonte, Federico Fancellu, Sorcha Gilroy, Ida Szubert, Nikolay Bogoychev, Naomi Saphra, Joana Ribeiro and Clara Vania for comments on previous drafts. This work was supported in part by a James S McDonnell Foundation Scholar Award and a Google faculty research award.

References

Oliver Adams, Graham Neubig, Trevor Cohn, and Steven Bird. 2016a. Learning a translation model from word lattices. In Proc. Interspeech.

Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura. 2016b. Learning a lexicon and translation model from phoneme lattices. In Proc. EMNLP.

Antonios Anastasopoulos, David Chiang, and Long Duong. 2016. An unsupervised probability model for speech-to-translation alignment of low-resource languages. In Proc. EMNLP.

Sameer Bansal, Herman Kamper, Sharon Goldwater, and Adam Lopez. 2017. Weakly supervised spoken term discovery using cross-lingual side information. In Proc. ICASSP.

Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In NIPS Workshop on End-to-end Learning for Speech and Audio Processing.

Laurent Besacier, Bowen Zhou, and Yuqing Gao. 2006. Towards speech translation of non written languages. In Proc. SLT.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. Speech Communication, 56:85–100.

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python. O'Reilly Media.

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.

Michael A. Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky. 2011. Rapid evaluation of speech representations for spoken term discovery. In Proc. Interspeech.

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

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In Proc. ACL.

Aren Jansen and Benjamin Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In Proc. ASRU.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. In Proc. ICASSP.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016a. A segmental framework for fully-unsupervised large-vocabulary speech recognition. arXiv preprint arXiv:1606.06950.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2016b. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. IEEE/ACM Trans. Audio, Speech, Language Process., 24(4):669–679.

Gaurav Kumar, Matt Post, Daniel Povey, and Sanjeev Khudanpur. 2014. Some insights from translating conversational telephone speech. In Proc. ICASSP.

Chia-ying Lee, T O'Donnell, and James Glass. 2015. Unsupervised lexicon discovery from acoustic input. Trans. ACL, 3:389–403.

Lara J. Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W. Black. 2015. Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain. In Proc. ASRU.

Robert Munro. 2010. Crowdsourced translation for emergency response in Haiti: the global collaboration of local knowledge. In AMTA Workshop on Collaborative Crowdsourcing for Translation.

Alex S. Park and James Glass. 2008. Unsupervised pattern discovery in speech. IEEE Trans. Audio, Speech, Language Process., 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.

Maarten Versteegh, Roland Thiollière, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. 2015. The Zero Resource Speech Challenge 2015. In Proc. Interspeech.

Alex Waibel and Christian Fugen. 2008. Spoken language translation. IEEE Signal Processing Magazine, 3(25):70–79.

Neil Zeghidour, Gabriel Synnaeve, Nicolas Usunier, and Emmanuel Dupoux. 2016. Joint learning of speaker and phonetic similarities with Siamese networks. In Proc. Interspeech.

3.3 Comments and further analysis

In this chapter we built an ST system using an algorithm developed by the zero-resource speech processing community. Our work highlights several challenges in using this method, primarily poor UTD performance in multispeaker conversational speech leading to low quality translations as measured by precision/recall. Further analysis revealed that these precision/recall scores were outperformed by a naive baseline where the 5 most frequent words in the training set were predicted for each dev/test set utterance, placing further doubt on the utility of the proposed UTD based ST system and our evaluation method.[1] This led us to more vigorously employing such naive baselines for comparison in all our future work.

In the next chapter we begin exploring neural models for ST, motivated in part by the promising work by Weiss et al. (2017) who trained a neural sequence-to-sequence model on around 160 hours of Spanish telephone speech data paired with English text translations. This was the first successful application of a neural ST model on a realistic speech corpus and their system produced good quality translations on the held out evaluation sets. In addition, compared to our UTD based system which produces a bag-of-words translation, the Weiss et al. system generates a complete English text sequence prediction (in order and including stop words) given a Spanish speech utterance. This allows the authors to compute more widely used evaluation metrics for translation systems such as BLEU scores.

[1] This was discovered by Antonios Anastasopoulos while we were collaborating on Anastasopoulos et al. (2017).

Chapter 4

Neural Speech-to-Text Translation

4.1 Introduction

In this chapter, we use a neural model to directly translate speech in a source language to target language text. We begin by adapting the model architecture introduced by Weiss et al. (2017), who used 160 hours of Spanish-English ST data to train a sequence-to-sequence neural model for speech-to-text. Their model produced high quality translations and also outperformed a cascaded ASR and MT model for speech translation. Though impressive, their work uses more than 10 times the training data used in our previous work (Chapter 3), and requires significant computational resources for training.

As the Weiss et al. (2017) system is not publicly available, we cannot replicate their results, nor can we run their model on data settings of our interest. Our goals for this chapter are as follows: (1) we create our own implementation of the Weiss et al. model; (2) we simulate low-resource conditions and train ST models with decreasing amounts of training data and measure translation performance; (3) we compute additional evaluation metrics to thoroughly evaluate the translations produced by our models.

4.2 Paper: Low-Resource Speech-to-Text Translation

Publication status. This work was published in Interspeech 2018.

Contributions. The ideas presented in this paper were developed jointly involving all the co-authors. They also provided regular feedback on all results and helped identify areas for improvement. Each co-author also played a key role in the publication writing process.

My individual contributions in this work were developing the code base for the neural model training and testing; experimental setup; managing the execution of all experiments on the GPUs; generating evaluation metrics and visualizations.

Interspeech 2018
2-6 September 2018, Hyderabad

Low-Resource Speech-to-Text Translation

Sameer Bansal[1], *Herman Kamper*[2], *Karen Livescu*[3], *Adam Lopez*[1], *Sharon Goldwater*[1]

[1]School of Informatics, University of Edinburgh, UK
[2]Stellenbosch University, South Africa
[3]Toyota Technological Institute at Chicago, USA

{sameer.bansal, sgwater, alopez}@inf.ed.ac.uk, kamperh@sun.ac.za, klivescu@ttic.edu

Abstract

Speech-to-text translation has many potential applications for low-resource languages, but the typical approach of cascading speech recognition with machine translation is often impossible, since the transcripts needed to train a speech recognizer are usually not available for low-resource languages. Recent work has found that neural encoder-decoder models can learn to directly translate foreign speech in high-resource scenarios, without the need for intermediate transcription. We investigate whether this approach also works in settings where both data and computation are limited. To make the approach efficient, we make several architectural changes, including a change from character-level to word-level decoding. We find that this choice yields crucial speed improvements that allow us to train with fewer computational resources, yet still performs well on frequent words. We explore models trained on between 20 and 160 hours of data, and find that although models trained on less data have considerably lower BLEU scores, they can still predict words with relatively high precision and recall—around 50% for a model trained on 50 hours of data, versus around 60% for the full 160 hour model. Thus, they may still be useful for some low-resource scenarios.
Index Terms: speech translation, low-resource speech processing, encoder-decoder models

1. Introduction

Conventional systems for speech-to-text translation [1] typically pipeline automatic speech recognition and machine translation, and since both of these applications require large training sets, these systems are available for only a tiny fraction of the world's highest-resource languages. But speech-to-text translation could be especially valuable in low-resource scenarios, for example in documentation of unwritten or endangered languages [2–6]; or in crisis relief, where emergency workers might need to respond to calls or requests in a foreign language [7]. These applications have motivated recent research on low-resource speech translation trained on a (potentially) cheaper resource: speech paired with its translation, with no intermediate transcriptions.[1] Initial work studied speech-to-text alignment without translation [11, 12], or focused on translating a few keywords using heuristic methods with just a few hours of training data [13, 14].

Recently, recurrent encoder-decoder models have been used to develop end-to-end speech-to-text translation models, which have been tested in high-resource settings on synthesized speech [15], audiobooks [10, 16], and a large dataset of conversational telephone speech [9]. So far, these neural models have been shown to produce high-quality translations with substantial resources—typically more than 100 hours of translated audio, from which models are trained for many days on multiple GPUs. But in our scenarios of interest, we expect to have less data, less time, and fewer computational resources. How will neural models perform in such low-resource settings?

[1]There is also recent work [8–10] using multitask learning to learn *both* translation and transcription models, showing improvements on the individual tasks. We focus here on the scenario without transcriptions.

	Weiss et al.	Our model
speech features	240 dim	80 dim
conv LSTM [17]	yes	no
decoder	characters	words
asynchronous SGD	yes	no
L2 penalty	10^{-6}	10^{-4}
Gaussian weight noise	yes	no
number of model replicas	10	1

Table 1: *Comparison of Weiss et al. [9] and our model.*

In this paper we perform an extensive investigation of the effects of training end-to-end speech-to-text translation models with limited resources. We implement a model inspired by the state-of-the-art architecture of Weiss et al. [9], but modify it to permit training in reasonable time on a single GPU (§2). The biggest change, compared to [9] (and also [8, 10]), is to use word-level decoding instead of character-level. We show that word-level models can be trained much faster than character-level models and still obtain comparable precision and recall; the tradeoff for this speed improvement is that they struggle to translate infrequent word types, leading to a drop in overall accuracy as measured by BLEU and METEOR.

We investigate the model's performance as we gradually reduce training data from the full 160 hours to as little as 20 hours. With only 50 hours of training data, our model still produces accurate translations for short utterances, with precision and recall around 50%. Although translation quality is much worse with 20 hours of data, precision and recall are still around 30%, which may be useful in low-resource applications. The 50-hour model trains in less than three days on a single GPU.

2. Speech-to-Text Model

Following Weiss et al. [9], we combine convolutional neural network (CNN) and recurrent neural network (RNN) components to build an encoder-decoder model with attention, but we modify the system (Table 1) so that we can train even our larger models in 3-5 days on a single GPU.

2.1. Speech encoder

Raw speech input is converted to Mel filterbank features computed every 10ms. Weiss et al.'s [9] model uses 240-dimensional input speech features, consisting of 80 filterbanks stacked with

10.21437/Interspeech.2018-1326

delta and delta-deltas. We use only 80-dimensional filterbank features. In preliminary experiments, we did not find much difference between 40, 40+deltas and 80 dimensions.

The filterbank features are fed into a stack of two CNN layers with rectified linear unit (ReLU) activations [18], each with 64 filters (compared to 32 used in Weiss et al.) of shape 3×3 along time and frequency, and a stride of 2×2. Striding reduces the sequence length by a factor of 4, which is important for reducing computation in the subsequent RNN layers.

At training time we use bucketing—80 buckets, with width increments of 25 frames—and padding of speech data. The utterances in the training set vary in length from 2 to 30 seconds; those longer than 20 seconds (80×25 frames) are truncated. We select up to 64 utterances from a bucket to create a mini-batch.

The output of the CNN layers is fed into three stacked bi-directional long short-term memory (LSTM) [19] layers, with 256-dimensional hidden states in each direction. Since the RNN operations are the main bottleneck, for initial hyper-parameter tuning we used uni-directional LSTMs with 300-dimensional hidden states. We then switch to bi-directional LSTM encoders to generate the final results.

2.2. English decoder

We use a word-level decoder, whereas Weiss et al. [9] used a character-level decoder. Since there are roughly five times as many characters as words, this greatly reduces sequence length, which speeds training for each individual utterance and allows us to use larger mini-batch sizes.

The English words are fed into an embedding layer followed by a stack of three uni-directional LSTM layers. We implement attention using the *global attentional model* with *general* score function and *input-feeding*, as described in [20]. We use beam decoding with a beam size of 8.

2.3. Optimizer

We use cross-entropy as the loss function. We regularize with dropout [21], with a ratio of 0.5 over the embedding and the LSTM layers [22], and an L2 penalty of 0.0001. We use a teacher-forcing [23] ratio of 0.8. Our code is implemented in Chainer [24].[2] Weiss et al.'s [9] model is trained using asynchronous stochastic gradient descent (ASGD) across 10 replicas; we train using a single model copy. Although Weiss et al.'s model benefited greatly from adding Gaussian noise to the weights during training (personal communication), we were unable to replicate this benefit and did not use noise injection.

3. Experiments

3.1. Experimental setup

We use the Fisher Spanish speech dataset [25]: a multispeaker corpus of telephone calls in a variety of Spanish dialects recorded in realistic noise conditions. The English translations were collected through crowdsourcing, as described in [26], and are used to train all models. There are four English references per utterance for the development and test sets, and one per utterance for the training set. We only use one of the two development sets (*dev*, not *dev2*). The training set comprises 160 hours of speech, split into 140K utterances; the development and test sets have about 4.5 hours of speech split into 4K utterances each. We lower-case and remove punctuation from the English translations and tokenize the text using NLTK [27].[3] This gives about 17K training word types and 1.5M tokens. There are about 300 out-of-vocabulary (OOV) word types (400 tokens) in the dev set, out of 40K tokens.

We first train a model using the entire 160 hours of labeled training data. To understand the impact of training data size on translation quality, we further train models using smaller subsets: 80, 50, 25, 20 hours of data, selected at random, from the entire training data. We use the same set of hyper-parameters—tuned for 160hrs—for all models. With these model parameters and training setup, we are able to train an epoch—a complete pass through the entire 160 hours of training data—in about 2 hours on a single Titan X (or equivalent) GPU.

3.2. Evaluation

In order to understand different aspects of model behavior, we evaluate with several metrics: BLEU [28], METEOR [29], and unigram precision/recall on the Fisher *dev* set, using all 4 human references.[4] BLEU measures how well a predicted translation matches a set of references based on a modified n-gram precision; it does not compute recall, and instead uses a *brevity penalty* to account for mismatch in reference and predicted lengths. METEOR computes both precision and recall and combines them via a harmonic mean, with greater weight given to recall. The final score is computed using a set of parameters, tuned for individual languages to correlate well with human judgments.

Whereas BLEU looks only for exact token matches between a prediction and set of references, METEOR also takes into account *stem*, *synonym*, and *paraphrase* matches. These four categories are weighted (by default) 1.0, 0.6, 0.8 and 0.6, respectively. For example, with these weights, a prediction of "eat" will score a recall of 1 against reference "eat" and 0.8 against reference "feed", a synonym match. METEOR can therefore be considered a more semantic measure. In low-resource settings, inexact translations that capture the semantics of an utterance are still useful. We use default settings and configuration files provided by the METEOR script for English. For comparison, we also provide human-level BLEU and METEOR scores by comparing one reference against the remaining 3.

In low-resource settings, BLEU scores may be very low and therefore difficult to interpret, but a model might still be able to predict (potentially important) keywords, which could be useful for cross-lingual applications. So, we also report word-level unigram precision using the BLEU script: if a predicted token is present in any of the 4 reference translations, it is considered a *True Positive*; otherwise it is a *False Positive*. Unigram recall is computed using the METEOR script, which includes *stem*, *synonym*, and *paraphrase* level matches. We also compute recall for *exact* matches only, by setting the METEOR weights to 1.0, 0.0, 0.0 and 0.0.

3.3. Results and discussion

Figures 1 and 2 show the BLEU, METEOR, and precision/recall scores on the dev set for each model as we change the amount of training data. Table 2 shows the BLEU scores on the *test* set.

[2]Code available at: `https://github.com/0xSameer/speech2text/tree/seq2seq`

[3]`http://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.word_tokenize`

[4]BLEU and precision are computed using `multi-bleu.pl` from the MOSES toolkit [30], which computes 4-gram BLEU. METEOR score and recall are computed using the script from `http://www.cs.cmu.edu/~alavie/METEOR/`.

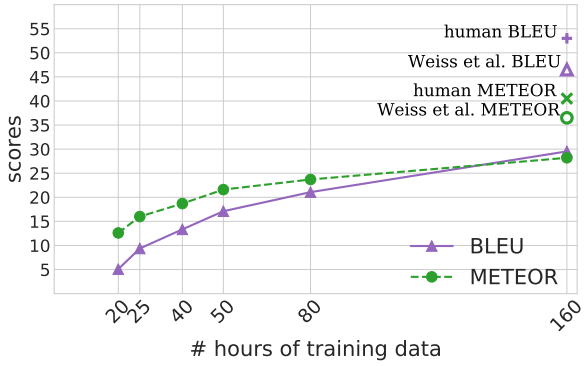


Figure 1: *Fisher dev set BLEU/METEOR results.*

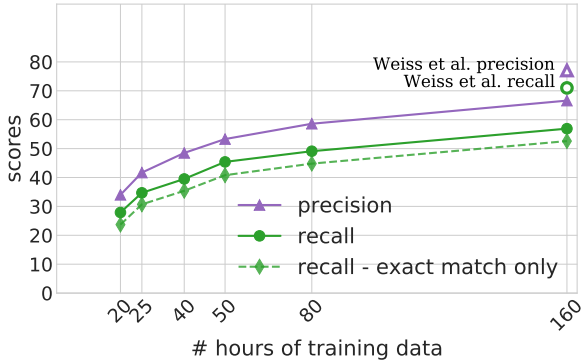


Figure 2: *Fisher dev set precision/recall results.*

Word vs. character models. Weiss et al.'s [9] character-level model achieves close to human performance.[5] Our word-level model, trained on 160 hours of data, converges to a BLEU score of 29.5 in about five days, a much lower score than Weiss et al.'s. One reason for this discrepancy is our different architecture and training setup (§2 and Table 1), which allowed us to train our models on our available computing resources.[6]

Training the character-level model takes nearly twice as long per epoch (4 hours for 160hrs of data) as the word-level model (2 hours). Our character-level model also has much smaller performance gains per epoch. To speed up character-level model training, we truncate utterances longer than 15 seconds (20 seconds for word models), reducing training time to 3 hours per epoch. Figure 3 compares character-level models trained in this way to word-level models for two training set sizes.

One of the main benefits of character models is their ability to gracefully handle OOV or infrequent words. On the dev set, the Weiss et al. model predicts about 130 word types that were not seen in training, which helps the model recall 7 OOV tokens out of 400. This is too small an effect to account for the performance differences, so we also analyze performance for a range of word frequencies. Table 3 shows the precision/recall

[5]The human scores are computed using 3 references; BLEU scores for all models are 2-4 points lower when using 3 instead of 4 references.

[6]Since our paper was submitted, [8] reported results on 20 hours of Spanish-English data for several multitask (translation/transcription) models and a baseline speech-to-text model. They used a character-level decoder and a different corpus (CALLHOME). They did not report detailed word-level BLEU scores, but said they were "between 7 and 10" for all models.

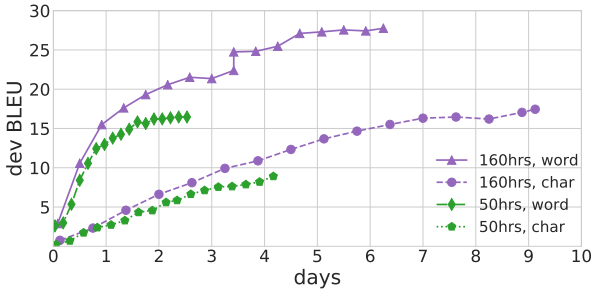


Figure 3: *Performance vs. training time for the word vs. character decoders. Each marker denotes 5 epochs.*

W	160h	80h	50h	40h	25h	20h
47.3	29.4	21.4	18.2	13.6	8.9	5.3

Table 2: *BLEU scores of (***W***)eiss et al.'s model and our models on the Fisher test set.*

for the 160hrs model and Weiss et al.'s [9] model, for words of different frequencies. We only consider content word types—words that are more than five characters long and are not in the NLTK stopword lists. The word-level model recall drops rapidly for *medium* frequency words, and for *rare* word types it has almost 0% recall. From this, we see that the main benefit of the character-level model is in handling of rare words, rather than previously unseen words.

Training considerations. Regularization parameters are critical to model performance. Figure 4 shows that increasing the L2 weight decay to a rate of 10^{-4} from 10^{-6} improved BLEU by about 2 points. Even though we use a high L2 penalty and dropout ratio, the models can overfit; the training loss continues to decrease, and we use early stopping based on dev set BLEU.

We also tried using batch normalization [31] at each CNN layer, and layer normalization [32] at each LSTM layer; but these did not have any noticeable impact on training performance.

Other design considerations. Using a bi-directional LSTM encoder does not seem to have an effect for the largest training sets, but improves BLEU by about one point in the $\leq$50 hour cases

	Rare	Medium	Frequent
training types	12K	1K	386
training tokens	30K	56K	200K
Precision (%):			
Weiss et al.	81.1	76.3	79.6
160hrs	87.5	69	65.7
Recall (%):			
Weiss et al.	24.5	65.4	78.1
160hrs	1.1	36.9	64.4

Table 3: *Content word frequency vs. dev precision/recall.* Rare *words have $\leq$10 tokens per type in the training text;* medium *have 25–100 tokens;* frequent *have $\geq$150 tokens.*

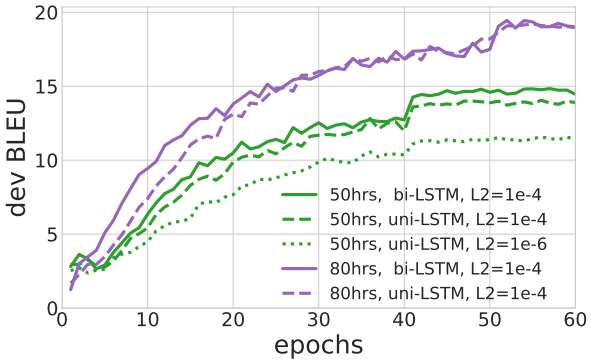


Figure 4: *Performance comparison: uni-directional vs. bi-directional encoders, L2 loss penalties.*

model	translations
Ref	so no yes but there are people who do get bothered a lot
W	so no yes there are people that do bother a lot
160h	so no if people are bothering a lot
80h	so if there are people that are bothering a lot
50h	so no yes that's why it bothers me a lot
40h	so if you think that it's like a lot
25h	so i don't know if people who are bother me much
20h	so if you have a car you can do it a lot
Ref	greetings ah my name is jenny and i'm calling from new york
W	hi ah my name is jenny i'm calling from new york
160h	hi ah my name is jenny i'm calling from new york
80h	good ah my name is jenny i'm calling from new york
50h	good ah my name is jenny calling from new york
40h	well ah i'm calling from from new york
25h	good ah my name is peruvian i'm calling from new york
20h	good ah my name is jenny

Table 4: *Example translations of (***W***)eiss et al.'s model and our models on dev set utterances, with stem-level n-gram matches to the reference sentence underlined.*

(Figure 4). This comes at a training time cost: bi-directional encoder models are almost 50% slower to train per epoch.

Figure 5 shows the improvement in BLEU scores by using beam decoding, over greedy decoding. Beam decoding always helps, but has a larger effect with more training data.

Exact vs. semantic matches. Figure 1 shows that the gap in METEOR scores between our models and Weiss et al.'s is much smaller than the gap in BLEU, and METEOR degrades more slowly as training data is reduced. This suggests that although our models are much worse at predicting the exact words in the reference translations, they often predict near-exact matches.

Table 4 shows some example predictions. As expected from the BLEU/METEOR results, the translations trained on more than 50 hours are fairly good, though they may contain different forms of the content words than are in the reference (e.g. *bothering/bothers* vs. *bothered*). The models trained on less data are clearly worse: they usually get some words right, which could be useful for keyword spotting or topic modeling in low-resource settings, but in some cases (as in the first example) the correctly

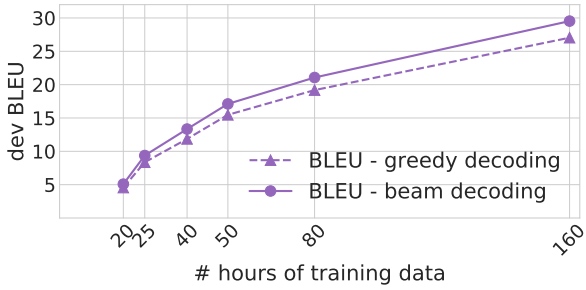


Figure 5: *Performance comparison: Greedy vs. beam decoding.*

predicted words do not carry much of the meaning.

Neural models in the extreme low-data setting. One may wonder whether, for very low-data settings, neural models still outperform older non-neural models at all. While we did not directly compare to a non-neural model, one indication is that, in the lowest-data settings, BLEU and METEOR are much worse but unigram precision and recall are still in the 25-35% range. These results compare very favorably to the 2-10% precision/recall reported by [14], who used a heuristic speech-to-text translation system trained on the CALLHOME corpus (also about 20 hours of Spanish conversational telephone speech with English text translations). So, it appears that even in a very low-resource setting with a model that is not state-of-the-art, the neural approach significantly outperforms previous non-neural models.

4. Conclusion

We performed a thorough analysis of a neural end-to-end speech-to-text translation model, with a specific focus on how performance is affected when using limited computational resources and limited amounts of data. We also showed the effects of a number of architectural design decisions using several performance metrics. While word-level models fall behind previously proposed character-level models when trained on around 160 hours of translated speech, our word-level models can be trained much faster and give reasonable performance on smaller training sets. Although translation quality drops, models trained on only 20 hours of translated speech achieve precision and recall of around 30% for content words. This could still be useful in search applications in severely low-resource scenarios. We believe that our extensive analyses in this work will contribute to better decision-making for architectural choices in computation- and data-limited settings.

In future work we aim to consider sub-word modelling [33], which could balance the trade-off between training costs and translation performance. In addition, we plan to try speech features that are targeted to low-resource multi-speaker settings [34, 35] and speaker normalization [36].

5. Acknowledgements

We thank Ron Weiss and Jan Chorowski for sharing their translation output, and Kenneth Heafield for giving access to GPUs. This work was supported in part by a James S. McDonnell Foundation Scholar Award and a Google faculty research award.

6. References

[1] A. Waibel and C. Fugen, "Spoken language translation," *IEEE Signal Proc. Mag.*, vol. 3, no. 25, pp. 70–79, 2008.

[2] L. Besacier, B. Zhou, and Y. Gao, "Towards speech translation of non written languages," in *Proc. SLT*, 2006.

[3] L. J. Martin, A. Wilkinson, S. S. Miryala, V. Robison, and A. W. Black, "Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain," in *Proc. ASRU*, 2015.

[4] O. Adams, G. Neubig, T. Cohn, and S. Bird, "Learning a translation model from word lattices." in *Proc. Interspeech*, 2016.

[5] O. Adams, G. Neubig, T. Cohn, S. Bird, Q. T. Do, and S. Nakamura, "Learning a lexicon and translation model from phoneme lattices," in *Proc. EMNLP*, 2016.

[6] A. Anastasopoulos and D. Chiang, "A case study on using speech-to-translation alignments for language documentation," in *Proc. ACL*, 2017.

[7] R. Munro, "Crowdsourced translation for emergency response in Haiti: The global collaboration of local knowledge," in *AMTA Workshop Collaborative Crowdsourcing Transl.*, 2010.

[8] A. Anastasopoulos and D. Chiang, "Tied multitask learning for neural speech translation," in *Proc. NAACL HLT*, 2018.

[9] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[10] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proc. ICASSP*, 2018.

[11] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proc. NAACL HLT*, 2016.

[12] A. Anastasopoulos, D. Chiang, and L. Duong, "An unsupervised probability model for speech-to-translation alignment of low-resource languages," *arXiv preprint arXiv:1609.08139*, 2016.

[13] A. Anastasopoulos, S. Bansal, D. Chiang, S. Goldwater, and A. Lopez, "Spoken term discovery for language documentation using translations," in *Proc. Workshop EMNLP-SCNLP*, 2017.

[14] S. Bansal, H. Kamper, A. Lopez, and S. Goldwater, "Towards speech-to-text translation without speech recognition," in *Proc. EACL*, 2017.

[15] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop End-to-End Learn. Speech Audio Process.*, 2016.

[16] A. C. Kocabiyikoglu, L. Besacier, and O. Kraif, "Augmenting Librispeech with French translations: A multimodal corpus for direct speech translation evaluation," *arXiv preprint arXiv:1802.03142*, 2018.

[17] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. NIPS*, 2015.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. international conference on machine learning*, 2010.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, 1997.

[20] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 2014.

[22] Y. Gal, "A theoretically grounded application of dropout in recurrent neural networks," *arXiv:1512.05287*, 2015.

[23] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, 1989.

[24] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: A next-generation open source framework for deep learning," in *Proc. LearningSys*, 2015.

[25] D. Graff, S. Huang, I. Cartagena, K. Walker, and C. Cieri, "Fisher Spanish Speech (LDC2010S01)," https://catalog.ldc.upenn.edu/ldc2010s01.

[26] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved speech-to-text translation with the Fisher and Callhome Spanish-English speech translation corpus," in *Proc. IWSLT*, 2013.

[27] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proc. ACL Workshop Effective Tools Methodologies Teaching Natural Language Process. Comput. Linguistics*, 2002.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002.

[29] A. Lavie and A. Agarwal, "Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proc. WMT*, 2007.

[30] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL Demo and Poster Sessions*, 2007.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[32] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[33] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016.

[34] H. Kamper, M. Elsner, A. Jansen, and S. J. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. ICASSP*, 2015.

[35] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *ArXiv e-prints*, 2016.

[36] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint learning of speaker and phonetic similarities with Siamese networks," in *Proc. Interspeech*, 2016.

4.3 Comments, updates, and further analysis

In this chapter we trained several ST models using a neural model architecture adapted from Weiss et al. (2017). Our decision to model English text as a sequence of words (instead of characters), helped us expedite the training process, but at a cost of generalization performance. Our aim with this set of experiments was to test whether neural models will learn under low-resource conditions. We observe that with as little as 20 hours of training data, our ST model achieves a BLEU score of around 5 and a unigram precision/recall of around 30%. Although these scores seem low, they are a positive sign that the model is learning something useful and motivate further exploration in this data setting. There is clearly room for improvement, as while carrying out these experiments we were not targeting the best performance possible which would have required careful hyperparameter tuning (considerably extending training time) for each individual ST model and instead used the same set of hyperparameters — optimized for the 160 hour training data setting — for all training data settings.

We now turn our attention towards improving ST performance in low-resource settings and modify the neural model architecture and training procedures. We describe the changes made and report the improved scores in Section 4.3.1. We include further analysis of our results and compare neural model based approach for ST to previous methods. We end the chapter with a discussion about some of the other methods we tried to improve translation performance, but which were ultimately unsuccessful.

4.3.1 Improved ST model architecture and training

We make the following changes to our neural model presented in Bansal et al. (2018) and training procedures:

1. **Subword modeling.** We switch to subword-level modeling at the decoder. We segment the target English text into subword units using byte pair encoding (BPE; Gage, 1994; Sennrich et al., 2016b). We use the English text from the full 160-hour Fisher training set as input to learn a target vocabulary of 1000 BPE subword units.[1] This change makes our model an open-vocabulary system (it can predict unseen words).

[1] We settled on 1000 BPE units as it provides a balance between the vocabulary size (types) and the number of tokens for training. We also experimented with a BPE vocabulary of 100 and 10,000.

Representation	Decoder input	# tokens
word	i live in the bronx	5
character	i ␣ l i v e ␣ i n ␣ t h e ␣ b r o n x	19
BPE	i live in the br⋆ on⋆ x	7

Table 4.1: Word, character and BPE representations for the English text: "i live in the bronx". ␣ indicates a space character; ⋆ symbol indicates a subword boundary.

Using BPE to preprocess the English text gives us 1.9 million tokens, compared to 1.5 million words, an increase of 25%. In practice this results in a marginal increase in the overall training time; but is still a lot faster to train than a character-level model.

As an example, Table 4.1 shows the different representations for the English text: "i live in the bronx", which contains 5 word/19 character tokens. The BPE encoding splits the infrequently occuring *bronx* token into a sequence of subword units: *br⋆*, *on⋆*, and *x*, where the ⋆ symbol indicates a subword boundary. The subword unit *br⋆* is also used to represent the word *british* as *br⋆ it⋆ ish*.

2. **Adding noise.** To help improve regularization during training, we add Gaussian noise with standard deviation of 0.25 to the speech features during training, and drop frames with a probability of 0.10. After 20 epochs, we corrupt the true decoder labels by sampling a random output label with a probability of 0.3. This penalizes high model confidence while predicting the target word/subword unit and is similar to *label smoothing* (Szegedy et al., 2016). These techniques are based on preliminary experiments we carried out using the development set.

3. **MFCC features.** We use 13 dimensional Mel-Frequency Cepstral Coefficients (MFCCs) extracted from the raw speech input, instead of using 80 dimensional Mel filterbanks. The MFFC feature vectors are fed into a stack of two CNN layers, with 128 and 512 filters with a filter width of 9 frames each, and a stride of 2 along time. The switch to MFFCs was motivated primarily for computational reasons (especially disk space).

Our final neural network model architecture is shown in Figure 4.1. For comparison, the Weiss et al. (2017) model is shown in Figure 4.2.

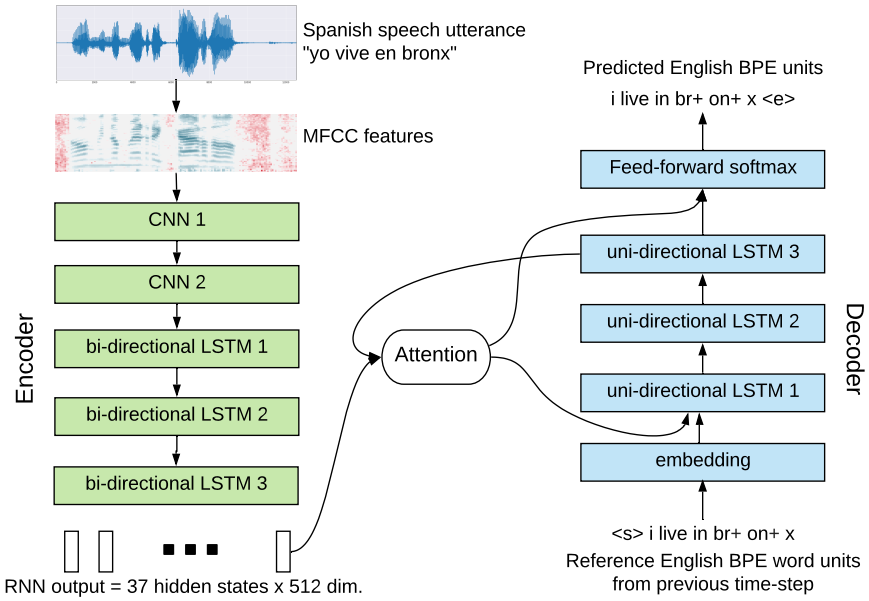


Figure 4.1: Proposed Encoder-decoder with attention model architecture for low-resource ST. The encoder input is the Spanish speech utterance: *yo vive en bronx* translated as *i live in bronx*. The decoder output is English subword units.

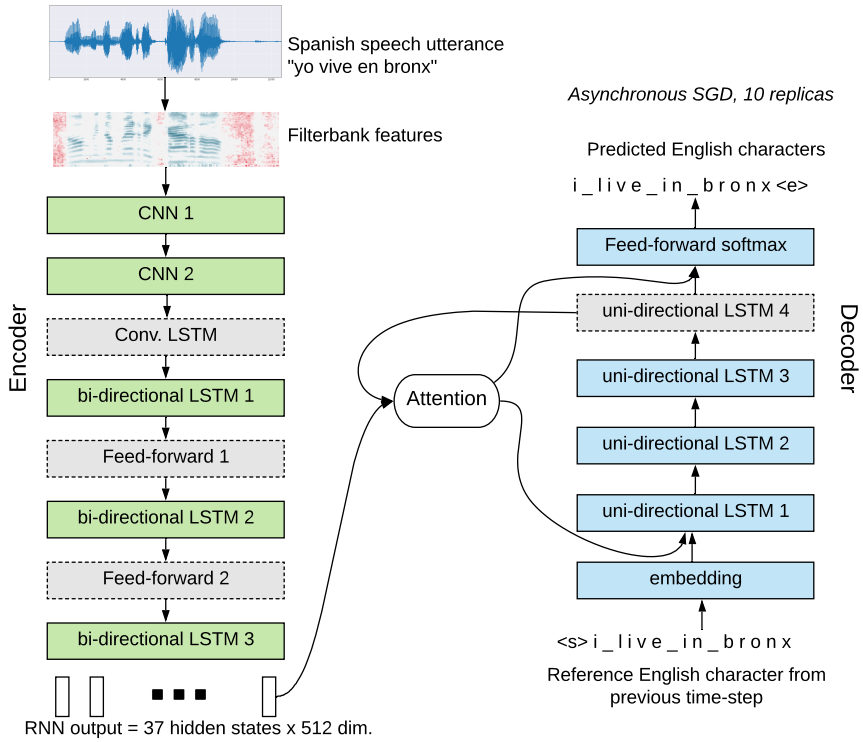


Figure 4.2: Weiss et al. encoder-decoder with attention model architecture for ST. The decoder output is English characters. Layers in grey/dotted are not used in our adaptation.

	BLEU	Precision	Recall
top 15 training words baseline	—	24.9	20.0
20hrs (Bansal et al., 2018)	5.2	35.1	28.7
20hrs + mfcc + word	7.9	38.5	34.0
20hrs + mfcc + subword	8.0	39.8	34.5
20hrs + mfcc + word + noise	8.1	39.2	33.9
20hrs + mfcc + subword + noise	**10.8** (↑ 5.6)	45.1	38.7
50hrs (Bansal et al., 2018)	18.2	54.9	46.7
50hrs + mfcc + subword + noise	**22.7** (↑ 4.5)	58.7	51.9

Table 4.2: Fisher test set BLEU, and precision/recall scores for Spanish-English ST.

Results. In the following, we refer to our previously published results as *Nhrs (Bansal et al., 2018)*, where *N* is the size of the training set in hours. For our new results, we add the suffixes *+word* or *+subword* to denote word-level or subword-level decoding respectively. We use the suffix *+noise* if we add noise during training; and *+mfcc* for MFCC features. For example, *20hrs+subword+noise+mfcc* denotes a Spanish-English ST model trained using 20 hours of data, with subword-level decoding, noise addition during training, and MFCC features. We also include precision and recall scores for a naive baseline model which outputs the K most frequent training words for each test set utterance. We tried a range of values for K (up to 50) and show the results for $K = 15$, where the precision/recall are most similar.

Table 4.2 shows the test set BLEU, precision and recall scores for Spanish-English ST models trained using 20 hours and 50 hours of data, before and after making the changes described above. We observe that the BLEU scores improve by around 5 points; precision/recall by almost 5-10% points, compared to results published in Bansal et al. (2018). For the 20 hour setting, we conduct ablation experiments and observe that the best word-level model (*20hrs + word + noise + mfcc*) achieves a BLEU score of 8.1; or around 2.5 BLEU points below our best model with subword-level decoding (*20hrs + subword + noise + mfcc*), which achieves a BLEU score of 10.8.

4.3.2 Stability of ST models in low-resource settings

We focus on the 20 hours Spanish-English ST setting, where our improved ST model achieved a BLEU score of 10.8 (Table 4.2) and test the following:

1. **Weight initialization.** We test whether the results are robust to the neural network weight initialization, by training several models with different starting values for all the layers (LSTM, CNN, embedding, etc. weights).

2. **Training sample.** We test whether our results are strongly tied to the specific 20 hours of training data we sampled from the entire Fisher corpus, by re-sampling a new set of 20 hours of ST training data.

The mean BLEU score on the Fisher dev set for these models (total 5) is 10.5 with a standard error of 0.17. The highest BLEU score measured is 11.2 and the lowest 9.9. The relatively similar score values suggests that ST models trained under such low-resource settings are fairly stable.

4.3.3 Comparison with state-of-the-art ST and human topline

As a final step, we trained ST models using the entire 160 hours of data in the Fisher corpus and report the results in Table 4.3. Our new model, *160hrs* + *word* + *noise* + *mfcc*, with changes described in Section 4.3.1, achieved a BLEU score of 33.3 on the test set with a training time of around a week; an improvement of around 4 BLEU points from Bansal et al. (2018). With extended training times (around 3 weeks) and further hyperparameter tuning, our ST model (+ *extended training*) achieved a BLEU score of 35.8 on the test set.[2]

However, even with the improvements, we were not able to replicate the score of 47.3 achieved by Weiss et al. (2017) when using the same amount of training data. Our results are closer to recent work by Sperber et al. (2019) and Salesky et al. (2019b), who report a BLEU score of 35.3 and 33.7, respectively, on the same dataset and also note that they weren't able to match the results of Weiss et al. (2017).[3] Training ST models to achieve state-of-the-art performance in high-resource settings requires

[2]The model architecture (number of layers, etc.) was not modified, and instead the hyperparameters related to regularization were changed.

[3]We only compare against the BLEU scores for the direct ST baselines reported by Weiss et al. (2017), Sperber et al. (2019), and Salesky et al. (2019b).

	BLEU
160hrs (Bansal et al., 2018)	29.4
160hrs + mfcc + subword + noise	33.2
+ extended training	35.8
Weiss et al. (2017)	47.3
Sperber et al. (2019)	35.3
Salesky et al. (2019b)	33.7

Table 4.3: Fisher test set BLEU scores for Spanish-English direct ST.

considerable hyperparameter tuning and computational resources (Weiss et al. used 10 model replicas for ASGD training), and is outside the scope of our work. Nevertheless, it is also important to ensure our model scores are comparable to contemporaneous work and the work by Sperber et al. (2019) and Salesky et al. (2019b) helps provide external (and independent) validation.

Comparison with human topline. To put the reported BLEU scores for our low-resource ST models in perspective, we compute an approximate upper bound for BLEU on the Fisher dev set. For each dev set utterance, the dataset provides 4 reference human translations. By holding out a reference, and comparing it to the remaining 3, we compute the BLEU score for the human translations themselves. We then recompute the BLEU scores for our 20 hour ST model and the translations produced by the Weiss et al. (2017) model using only 3 references as well. Table 4.4 shows that there is over a 30 BLEU point difference between our 20 hour model and the state-of-the-art; and only 10 BLEU points between the state-of-the-art and the human reference BLEU score. We therefore expect the translations produced by our model to be noisy.

	BLEU (3 references)
human	52.8
Weiss et al. (160 hours)	43.0
Our 20 hours	9.8

Table 4.4: Fisher dev set BLEU scores for Spanish-English direct ST. Computed using 3 references (instead of all 4).

4.3.4 Comparison with pattern-detection based ST.

In Chapter 3, we trained Spanish-English ST models using a speech pattern detection based algorithm (UTD-ST). We find that neural ST models offer several advantages compared to UTD-ST:

1. **Faster to train.** Both methods take about a day to complete training on 20 hours of ST data. UTD-ST, however, scales poorly as more training data is added: in order to detect speech patterns, each speech utterance has to be compared against the entire training set, resulting in a time complexity of $O(N^2 \times M \log M)$, where M is a typical speech utterance duration; and N is the number of training set utterances (around 18K for 20 hours of data). Note that as N>>M, for the purposes of this analysis we will consider M as a constant and focus on time complexity as a function of training set size. Therefore, for UTD-ST the time complexity is effectively $O(N^2)$, scaling quadratically with the training set size.

   The neural model in contrast has a training time complexity of $O(N)$ which scales linearly as more training data is added.

   In addition, the neural model is fully parameterized by the weights of all its layers. Once trained, we no longer need to retain the training data, and just save the neural network weights, which is a fixed size. If more training data becomes available in the future, the neural model can be further trained on this new set of speech utterances, with a complexity of $O(Q)$, where Q is the number of new training utterances added. Note that this no longer depends upon N.

   For UTD-ST, we have to retain the entire set of training data utterances (similar to a nonparametric model) and compare each new training example against the old examples for pattern detection. The time complexity of training will now be: $O(Q^2)$.[4] Therefore, with respect to both space and runtime, the neural models offer much better performance.

2. **Faster at making predictions.** Importantly, the neural model is far more efficient at making predictions for new audio utterances — taking about 1 second per prediction (using beam search) for an utterance several seconds long, with a complexity of $O(1)$ per test set utterance.

[4] The total number of operations for UTD pattern discovery when Q new utterances are added is equal to $(N+Q) \times Q$.

UTD-ST has a prediction time complexity of $O(K)$, where K is approximately equal to the number of word types in the training set (for our datasets this number is around 10K). A large value of K results in a longer prediction time per utterance in practice.

In addition, the neural model predicts a sequence of text, and is an open vocabulary system (subword modeling), which can potentially predict unseen words. UTD-ST in contrast predicts a bag-of-words and is a limited vocabulary system.

4.3.5 Other methods to improve ST

We tried several methods to improve translation performance:

1. Curriculum-learning (Bengio et al., 2009), where we initially train the model on simpler (shorter duration) utterances.

2. Using monolingual English text to learn a language model and transferring the decoder embeddings to the Spanish-English ST model (Ramachandran et al., 2017). We refer to this method as *LM-emb* and train a language model using the entire 160 hours of English text translations in the Fisher corpus.[5]

These methods seemed to work well initially and we observed up to 1.5 BLEU point improvement over a preliminary baseline (Table 4.5). However, most of these gains disappeared when we added random noise during training (Section 4.3.1). This perhaps suggests that these methods themselves are doing some form of regularization as well (Felbo et al., 2017). And unfortunately, the gains are not additive.

	w/o noise	**with noise**
20hrs + mfcc + subword	7.5	**10.8**
20hrs + mfcc + subword + curriculum	9.2	9.9
20hrs + mfcc + subword + LM-emb	8.0	10.2

Table 4.5: Fisher dev set BLEU for Spanish-English ST.

[5]Ramachandran et al. (2017) suggest transferring the LSTM and softmax layers as well. Due to differences in model architecture, we only tried transferring decoder embeddings.

4.4 Review and next steps

In this chapter we used a neural model to train ST systems in low-resource data settings. Our goal was to develop a system that's feasible in academic settings and to explore performance with less training data. Important factors in achieving this were the use of subword modeling at the decoder and regularization methods such as adding noise at the encoder and decoder layers. Our 20 hour results are promising and outperform most-frequent-word baselines, but are still very low. In the next chapter, we work towards improving the performance of our ST models by leveraging data from high-resource languages.

Chapter 5

Transfer learning for Speech-to-Text Translation

5.1 Introduction

In this chapter, we use external training data from a high-resource language to improve translation quality in low-resource ST settings. Neural models allow us to conveniently leverage such data by using transfer learning (Thrun, 1995): training a model on a preliminary task, and then transferring parameters (complete model transfer or selected layers) and continuing training on the primary task. Our earlier attempts at transfer learning were limited to using only English text data, and were not successful (Section 4.5). Here, we expand this to include audio paired with text transcriptions (ASR data). ASR data for high-resource languages is widely available and relatively easier to obtain, compared to labeling additional speech data in a low-resource language.

There is a rich history of work which has used transfer-learning to bootstrap ASR for a new target language with limited amounts of training data, using speech data from other languages. Schultz and Waibel (2001) trained an acoustic model, based on a conventional HMM-GMM architecture, by using transcribed speech data from multiple languages simultaneously. They fine-tuned this language independent model to build ASR for a new target language and showed improvements during evaluation. A similar approach was used by Niesler (2007). Where Schultz and Waibel used transcribed speech data for pre-training, Swietojanski et al. (2012) describe an unsupervised approach which uses untranscribed speech data from different languages to pre-train a

neural acoustic model for the low-resource target language. Both Schultz and Waibel (2001) and Swietojanski et al. (2012) show that data from different languages can help improve ASR performance. Closer to our work in this chapter, Ghoshal et al. (2013); Huang et al. (2013); Heigold et al. (2013) pre-train neural acoustic models using transcribed speech. They show that sharing hidden layers and training on multiple languages helps ASR performance for a new language.

5.2 Paper: Pre-training on High-Resource Speech Recognition Improves Low-Resource Speech-to-Text Translation

Publication status. This work was published in NAACL 2018.

Contributions. The ideas presented in this paper were developed jointly involving all the co-authors. They also provided regular feedback on all results and helped identify areas for improvement. Each co-author also played a key role in the publication writing process.

My individual contributions in this work were extending our neural model code base (discussed in Chapter 4) for transfer-learning; pre-training all the ASR models; experimental setup for transfer-learning between ASR and ST; managing the execution of all experiments on the GPUs; generating evaluation metrics and visualizations.

Pre-training on High-Resource Speech Recognition Improves Low-Resource Speech-to-Text Translation

Sameer Bansal[1] **Herman Kamper**[2] **Karen Livescu**[3] **Adam Lopez**[1] **Sharon Goldwater**[1]
[1]School of Informatics, University of Edinburgh, UK
[2]E&E Engineering, Stellenbosch University, South Africa
[3]Toyota Technological Institute at Chicago, USA
{sameer.bansal, sgwater, alopez}@inf.ed.ac.uk
kamperh@sun.ac.za
klivescu@ttic.edu

Abstract

We present a simple approach to improve direct speech-to-text translation (ST) when the source language is low-resource: we pre-train the model on a high-resource automatic speech recognition (ASR) task, and then fine-tune its parameters for ST. We demonstrate that our approach is effective by pre-training on 300 hours of English ASR data to improve Spanish-English ST from 10.8 to 20.2 BLEU when only 20 hours of Spanish-English ST training data are available. Through an ablation study, we find that the pre-trained encoder (acoustic model) accounts for most of the improvement, despite the fact that the shared language in these tasks is the target language text, not the source language audio. Applying this insight, we show that pre-training on ASR helps ST even when the ASR language differs from both source and target ST languages: pre-training on French ASR also improves Spanish-English ST. Finally, we show that the approach improves performance on a true low-resource task: pre-training on a combination of English ASR and French ASR improves Mboshi-French ST, where only 4 hours of data are available, from 3.5 to 7.1 BLEU.

1 Introduction

Speech-to-text Translation (ST) has many potential applications for low-resource languages: for example in language documentation, where the source language is often unwritten or endangered (Besacier et al., 2006; Martin et al., 2015; Adams et al., 2016a,b; Anastasopoulos and Chiang, 2017); or in crisis relief, where emergency workers might need to respond to calls or requests in a foreign language (Munro, 2010). Traditional ST is a pipeline of automatic speech recognition (ASR) and machine translation (MT), and thus requires transcribed source audio to train ASR and parallel text to train MT. These resources are often unavailable for low-resource languages, but for our potential applications, there may be some source language audio paired with target language text translations. In these scenarios, end-to-end ST is appealing.

Recently, Weiss et al. (2017) showed that end-to-end ST can be very effective, achieving an impressive BLEU score of 47.3 on Spanish-English ST. But this result required over 150 hours of translated audio for training, still a substantial resource requirement. By comparison, a similar system trained on only 20 hours of data for the same task achieved a BLEU score of 5.3 (Bansal et al., 2018). Other low-resource systems have similarly low accuracies (Anastasopoulos and Chiang, 2018; Bérard et al., 2018).

To improve end-to-end ST in low-resource settings, we can try to leverage other data resources. For example, if we have transcribed audio in the source language, we can use multi-task learning to improve ST (Anastasopoulos and Chiang, 2018; Weiss et al., 2017; Bérard et al., 2018). But source language transcriptions are unlikely to be available in our scenarios of interest.

Could we improve low-resource ST by leveraging data from a high-resource language? For ASR, training a single model on multiple languages can be effective for all of them (Toshniwal et al., 2018b; Deng et al., 2013). For MT, *transfer learning* (Thrun, 1995) has been very effective: pre-training a model for a high-resource language pair and transferring its parameters to a low-resource language pair when the target language is shared (Zoph et al., 2016; Johnson et al., 2017). Inspired by these successes, we show that low-resource ST can leverage transcribed audio in a high-resource target language, or even a different language altogether, simply by pre-training a model for the high-resource ASR task, and then transferring and fine-tuning some or all of the model's parameters for low-resource ST.

Proceedings of NAACL-HLT 2019, pages 58–68
Minneapolis, Minnesota, June 2 - June 7, 2019.

We first test our approach using Spanish as the source language and English as the target. After training an ASR system on 300 hours of English, fine-tuning on 20 hours of Spanish-English yields a BLEU score of 20.2, compared to only 10.8 for an ST model without ASR pre-training. Analyzing this result, we discover that the main benefit of pre-training arises from the transfer of the *encoder* parameters, which model the input acoustic signal. In fact, this effect is so strong that we also obtain improvements by pre-training on a language that differs from both the source and the target: pre-training on French and fine-tuning on Spanish-English. We hypothesize that pre-training the encoder parameters, even on a different language, allows the model to better learn about linguistically meaningful phonetic variation while normalizing over acoustic variability such as speaker and channel differences. We conclude that the acoustic-phonetic learning problem, rather than translation itself, is one of the main difficulties in low-resource ST. A final set of experiments confirm that ASR pre-training also helps on another language pair where the input is truly low-resource: Mboshi-French.

2 Method

For both ASR and ST, we use an encoder-decoder model with attention adapted from Weiss et al. (2017), Bérard et al. (2018) and Bansal et al. (2018), as shown in Figure 1. We use the same model architecture for all our models, allowing us to conveniently transfer parameters between them. We also constrain the hyper-parameter search to fit a model into a single Titan X GPU, allowing us to maximize available compute resources.

We use a pre-trained English ASR model to initialize training of Spanish-English ST models, and a pre-trained French ASR model to initialize training of Mboshi-French ST models. During ST training, all model parameters are updated. In these configurations, the decoder shares the same vocabulary across the ASR and ST tasks. This is practical for settings where the target text language is high-resource with ASR data available.

In settings where both ST languages are low-resource, ASR data may only be available in a third language. To test whether transfer learning will help in this setting, we use a pre-trained French ASR model to train Spanish-English ST models; and English ASR for Mboshi-French models. In these cases, the ST languages are different from the ASR language, so we can only transfer the encoder parameters of the ASR model, since the dimensions of the decoder's output softmax layer are indexed by the vocabulary, which is not shared.[1] Sharing only the speech encoder parameters is much easier, since the speech input can be preprocessed in the same manner for all languages. This form of transfer learning is more flexible, as there are no constraints on the ASR language used.

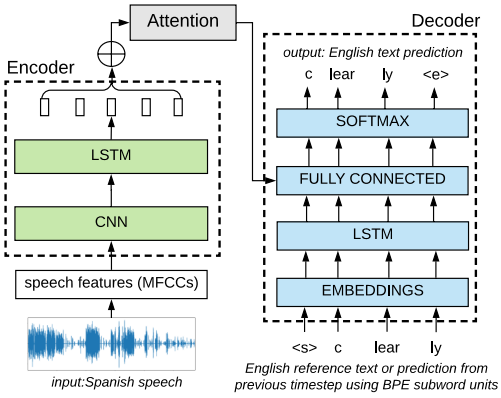


Figure 1: Encoder-decoder with attention model architecture for both ASR and ST. The encoder input is the Spanish speech utterance *claro*, translated as *clearly*, represented as BPE (subword) units.

3 Experimental Setup

3.1 Data sets

English ASR. We use the Switchboard Telephone speech corpus (Godfrey and Holliman, 1993), which consists of around 300 hours of English speech and transcripts, split into 260k utterances. The development set consists of 5 hours that we removed from the training set, split into 4k utterances.

French ASR. We use the French speech corpus from the GlobalPhone collection (Schultz, 2002), which consists of around 20 hours of high quality read speech and transcripts, split into 9k utterances. The development set consists of 2 hours, split into 800 utterances.

Spanish-English ST. We use the Fisher Spanish speech corpus (Graff et al., 2010), which consists of 160 hours of telephone speech in a variety of Spanish dialects, split into 140K utterances. To simulate low-resource conditions, we construct smaller train-

[1] Using a shared vocabulary of characters or subwords is an interesting direction for future work, but not explored here.

ing corpora consisting of 50, 20, 10, 5, or 2.5 hours of data, selected at random from the full training data. The development and test sets each consist of around 4.5 hours of speech, split into 4K utterances. We do not use the corresponding Spanish transcripts; our target text consists of English translations that were collected through crowdsourcing (Post et al., 2013, 2014).

Mboshi-French ST. Mboshi is a Bantu language spoken in the Republic of Congo, with around 160,000 speakers.[2] We use the Mboshi-French parallel corpus (Godard et al., 2018), which consists of around 4 hours of Mboshi speech, split into a training set of 5K utterances and a development set of 500 utterances. Since this corpus does not include a designated test set, we randomly sampled and removed 200 utterances from training to use as a development set, and use the designated development data as a test set.

3.2 Preprocessing

Speech. We convert raw speech input to 13-dimensional MFCCs using Kaldi (Povey et al., 2011).[3] We also perform speaker-level mean and variance normalization.

Text. The target text of the Spanish-English data set contains 1.5M word tokens and 17K word types. If we model text as sequences of words, our model cannot produce any of the unseen word types in the test data and is penalized for this, but it can be trained very quickly (Bansal et al., 2018). If we instead model text as sequences of characters as done by Weiss et al. (2017), we would have 7M tokens and 100 types, resulting in a model that is open-vocabulary, but very slow to train (Bansal et al., 2018). As an effective middle ground, we use byte pair encoding (BPE; Sennrich et al., 2016) to segment each word into subwords, each of which is a character or a high-frequency sequence of characters—we use 1000 of these high-frequency sequences. Since the set of subwords includes the full set of characters, the model is still open vocabulary; but it results in a text with only 1.9M tokens and just over 1K types, which can be trained almost as fast as the word-level model.

The vocabulary for BPE depends on the frequency of character sequences, so it must be computed with respect to a specific corpus. For English, we use the full 160-hour Spanish-English ST target training text. For French, we use the Mboshi-French ST target training text.

3.3 Model architecture for ASR and ST

Speech encoder. As shown schematically in Figure 1, MFCC feature vectors, extracted using a window size of 25 ms and a step size of 10ms, are fed into a stack of two CNN layers, with 128 and 512 filters with a filter width of 9 frames each. In each CNN layer we stride with a factor of 2 along time, apply a ReLU activation (Nair and Hinton, 2010), and apply batch normalization (Ioffe and Szegedy, 2015). The output of the CNN layers is fed into a three-layer bi-directional long short term memory network (LSTM; Hochreiter and Schmidhuber, 1997); each hidden layer has 512 dimensions.

Text decoder. At each time step, the decoder chooses the most probable token from the output of a softmax layer produced by a fully-connected layer, which in turn receives the current state of a recurrent layer computed from previous time steps and an attention vector computed over the input. Attention is computed using the *global attentional model* with *general* score function and *input-feeding*, as described in Luong et al. (2015). The predicted token is then fed into a 128-dimensional embedding layer followed by a three-layer LSTM to update the recurrent state; each hidden state has 256 dimensions. While training, we use the predicted token 20% of the time as input to the next decoder step and the training token for the remaining 80% of the time (Williams and Zipser, 1989). At test time we use beam decoding with a beam size of 5 and length normalization (Wu et al., 2016) with a weight of 0.6.

Training and implementation. Parameters for the CNN and RNN layers are initialized using the scheme from (He et al., 2015). For the embedding and fully-connected layers, we use Chainer's (Tokui et al., 2015) default initialition.

We regularize using dropout (Srivastava et al., 2014), with a ratio of 0.3 over the embedding and LSTM layers (Gal, 2016), and a weight decay rate of 0.0001. The parameters are optimized using Adam (Kingma and Ba, 2015), with a starting alpha of 0.001.

[2] ethnologue.com/language/mdw

[3] In preliminary experiments, we did not find much difference between between MFCCs and more raw spectral representations like Mel filterbank features.

Following some preliminary experimentation on our development set, we add Gaussian noise with standard deviation of 0.25 to the MFCC features during training, and drop frames with a probability of 0.10. After 20 epochs, we corrupt the true decoder labels by sampling a random output label with a probability of 0.3.

Our code is implemented in Chainer (Tokui et al., 2015) and is freely available.[4]

3.4 Evaluation

Metrics. We report BLEU (Papineni et al., 2002) for all our models.[5] In low-resource settings, BLEU scores tend to be low, difficult to interpret, and poorly correlated with model performance. This is because BLEU requires exact four-gram matches only, but low four-gram accuracy may obscure a high unigram accuracy and inexact translations that partially capture the semantics of an utterance, and these can still be very useful in situations like language documentation and crisis response. Therefore, we also report word-level unigram precision and recall, taking into account *stem*, *synonym*, and *paraphrase* matches. To compute these scores, we use METEOR (Lavie and Agarwal, 2007) with default settings for English and French.[6] For example, METEOR assigns "eat" a recall of 1 against reference "eat" and a recall of 0.8 against reference "feed", which it considers a synonym match.

Naive baselines. We also include evaluation scores for a naive baseline model that predicts the K most frequent words of the training set as a bag of words for each test utterance. We set K to be the value at which precision/recall are most similar, which is always between 5 and 20 words. This provides an empirical lower bound on precision and recall, since we would expect any usable model to outperform a system that does not even depend on the input utterance. We do not compute BLEU for these baselines, since they do not predict sequences, only bags of words.

4 ASR results

Using the experimental setup of Section 3, we pretrained ASR models in English and French, and report their word error rates (WER) on development data in Table 1.[7] We denote each ASR model by *L-Nh*, where L is a language code and N is the size of the training set in hours. For example, *en-300h* denotes an English ASR model trained on 300 hours of data.

	en-100h	en-300h	fr-20h
WER	35.4	27.3	29.6

Table 1: Word Error Rate (WER, in %) for the ASR models used as pretraining, computed on Switchboard *train-dev* for English and Globalphone dev for French.

Training ASR models for state-of-the-art performance requires substantial hyper-parameter tuning and long training times. Since our goal is simply to see whether pre-training is useful, we stopped pretraining our models after around 30 epochs (3 days) to focus on transfer experiments. As a consequence, our ASR results are far from state-of-the-art: current end-to-end Kaldi systems obtain 16% WER on Switchboard *train-dev*, and 22.7% WER on the French Globalphone dev set.[8] We believe that better ASR pre-training may produce better ST results, but we leave this for future work.

5 Spanish-English ST

In the following, we denote an ST model by *S-T-Nh*, where S and T are source and target language codes, and N is the size of the training set in hours. For example, *sp-en-20h* denotes a Spanish-English ST model trained using 20 hours of data. We use the code *mb* for Mboshi and *fr* for French.

5.1 Using English ASR to improve ST

Figure 2 shows the BLEU and unigram precision/recall scores on the development set for baseline Spanish-English ST models and those trained after initializing with the *en-300h* model. Corresponding results on the test set (Table 2) reveal very similar patterns. The remainder of our analysis is confined to the development set. The naive baseline, which predicts the 15 most frequent English words in the training set, achieves a precision/recall of around 20%, setting a performance lower bound.

Low-resource: 20-50 hours of ST training data. Our baseline ST models substantially improve over

[4] github.com/0xSameer/ast

[5] We compute BLEU with `multi-bleu.pl` from the Moses toolkit (Koehn et al., 2007).

[6] cs.cmu.edu/~alavie/METEOR

[7] We computed WER with the NIST `sclite` script.

[8] These WER results taken from respective Kaldi recipes on GitHub, and may not represent the very best results on these data sets.

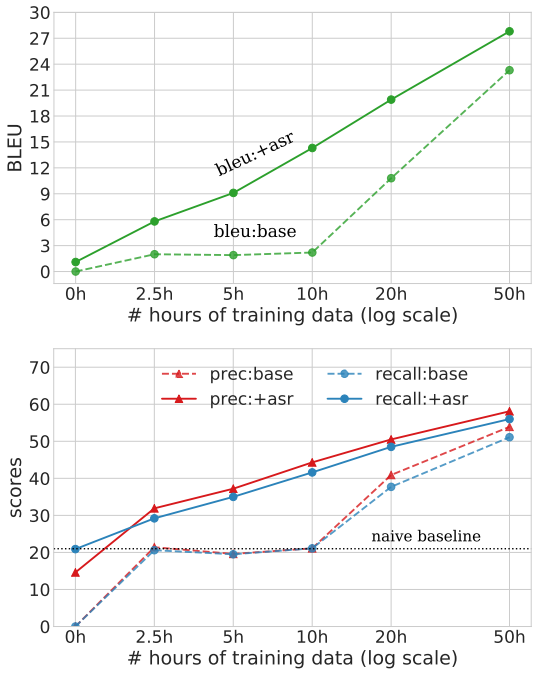


Figure 2: (top) BLEU and (bottom) Unigram precision/recall for Spanish-English ST models computed on Fisher dev set. base indicates no transfer learning; +asr are models trained by fine-tuning *en-300h* model parameters. *naive baseline* indicates the score when we predict the 15 most frequent English words in the training set.

$N =$	0	2.5	5	10	20	50
base	0	2.1	1.8	2.1	10.8	22.7
+asr	0.5	5.7	9.1	14.5	20.2	28.2

Table 2: BLEU scores for Spanish-English ST on the Fisher test set, using N hours of training data. base: no transfer learning. +asr: using model parameters from English ASR (en-300h).

Spanish	super caliente pero muy bonito
English	super hot but very nice
20h	you support it but it was very nice
20h+asr	you can get alright but it's very nice
50h	super expensive but very nice
50h+asr	super hot but it's very nice
Spanish	sí y usted hace mucho tiempo que que vive aquí
English	yes and have you been living here a long time
20h	yes i've been a long time what did you come here
20h+asr	yes and you have a long time that you live here
50h	yes you are a long time that you live here
50h+asr	yes and have you been here long

Table 3: Example translations on selected sentences from the Fisher development set, with stem-level n-gram matches to the reference sentence underlined. 20h and 50h are Spanish-English models without pre-training; 20h+asr and 50h+asr are pre-trained on 300 hours of English ASR.

previous results (Bansal et al., 2018) using the same train/test splits, primarily due to better regularization and modeling of subwords rather than words. Yet transfer learning still substantially improves over these strong baselines. For *sp-en-20h*, transfer learning improves dev set BLEU from 10.8 to 19.9, precision from 41% to 51%, and recall from 38% to 49%. For *sp-en-50h*, transfer learning improves BLEU from 23.3 to 27.8, precision from 54% to 58%, and recall from 51% to 56%.

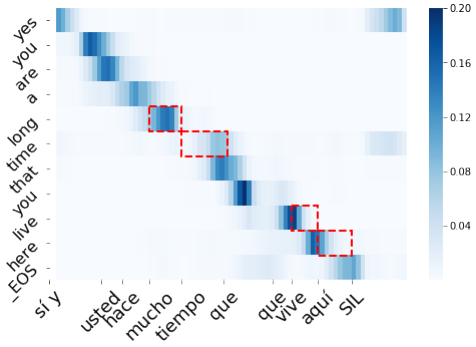
Very low-resource: 10 hours or less of ST training data. Figure 2 shows that without transfer learning, ST models trained on less than 10 hours of data struggle to learn, with precision/recall scores close to or below that of the naive baseline. But with transfer learning, we see gains in precision and recall of between 10 and 20 points.

We also see that with transfer learning, a model trained on only 5 hours of ST data achieves a BLEU of 9.1, nearly as good as the 10.8 of a model trained on 20 hours of ST data without transfer learning. In other words, fine-tuning an English ASR model—which is relatively easy to obtain—produces similar results to training an ST model on four times as much data, which may be difficult to obtain.

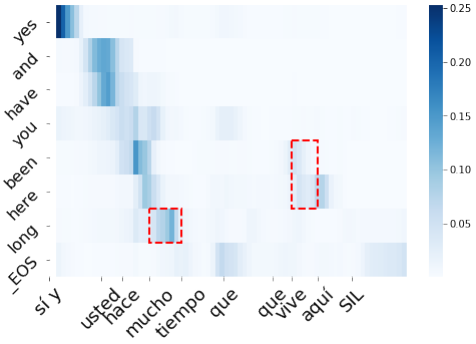
We even find that in the very low-resource setting of just 2.5 hours of ST data, with transfer learning the model achieves a precision/recall of around 30% and improves by more than 10 points over the naive baseline. In very low-resource scenarios with time constraints—such as in disaster relief—it is possible that even this level of performance may be useful, since it can be used to spot keywords in speech and can be trained in just three hours.

Sample translations. Table 3 shows example translations for models *sp-en-20h* and *sp-en-50h* with and without transfer learning using *en-300h*.

Figure 3 shows the attention weights for the last sample utterance in Table 3. For this utterance, the Spanish and English text have a different word order: *mucho tiempo* occurs in the middle of the speech utterance, and its translation, *long time*, is at the end of the English reference. Similarly, *vive aquí* occurs at the end of the speech utterance, while the translation, *living here*, is in the middle of the English reference. The baseline *sp-en-50h* model translates the words correctly but doesn't get



(a) **50h:baseline**



(b) **50h:asr**

Figure 3: Attention plots for the final example in Table 3, using 50h models with and without pre-training. The x-axis shows the reference Spanish word positions in the input; the y-axis shows the predicted English subwords. In the reference, *mucho tiempo* is translated to *long time*, and *vive aquí* to *living here*, but their order is reversed, and this is reflected in (b).

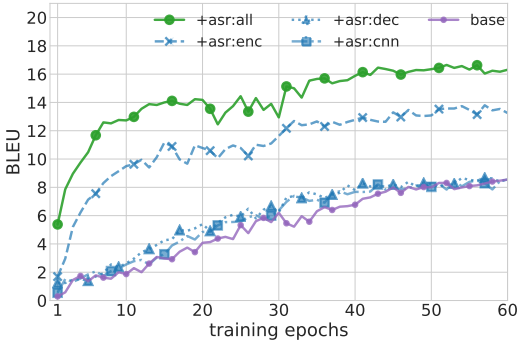


Figure 4: Fisher development set training curves (reported using BLEU) for *sp-en-20h* using selected parameters from *en-300h*: none (base); encoder CNN only (+asr:cnn); encoder CNN and LSTM only (+asr:enc); decoder only (+asr:dec); and all: encoder, attention, and decoder (+asr:all). These scores do not use beam search and are therefore lower than the best scores reported in Figure 2.

the English word order right. With transfer learning, the model produces a shorter but still accurate translation in the correct word order.

5.2 Analysis

To understand the source of these improvements, we carried out a set of ablation experiments. For most of these experiments, we focus on Spanish-English ST with 20 hours of training data, with and without transfer learning.

Transfer learning with selected parameters. In our first set of experiments, we transferred all parameters of the *en-300h* model, including the speech encoder CNN and LSTM; the text decoder embedding, LSTM and output layer parameters; and attention parameters. To see which set of parameters has the most impact, we train the *sp-en-20h* model by transferring only selected parameters from *en-300h*, and randomly initializing the rest.

The results (Figure 4) show that transferring all parameters is most effective, and that the speech encoder parameters account for most of the gains. We hypothesize that the encoder learns transferable low-level acoustic features that normalize across variability like speaker and channel differences to better capture meaningful phonetic differences, and that much of this learning is language-independent. This hypothesis is supported by other work showing the benefits of cross-lingual and multilingual training for speech technology in low-resource target languages (Carlin et al., 2011; Jansen et al., 2010; Deng et al., 2013; Vu et al., 2012; Thomas et al., 2012; Cui et al., 2015; Alumäe et al., 2016; Yuan et al., 2016; Renshaw et al., 2015; Hermann and Goldwater, 2018).

By contrast, transferring only decoder parameters does not improve accuracy. Since decoder parameters help when used in tandem with encoder parameters, we suspect that the dependency in parameter training order might explain this: the transferred decoder parameters have been trained to expect particular input representations from the encoder, so transferring only the decoder parameters without the encoder might not be useful.

Figure 4 also suggests that models make strong gains early on in the training when using transfer learning. The *sp-en-20h* model initialized with all model parameters (*+asr:all*) from *en-300h* reaches a higher BLEU score after just 5 epochs (2 hours) of training than the model without transfer learning trained for 60 epochs/20 hours. This again can be useful in disaster-recovery scenarios, where the

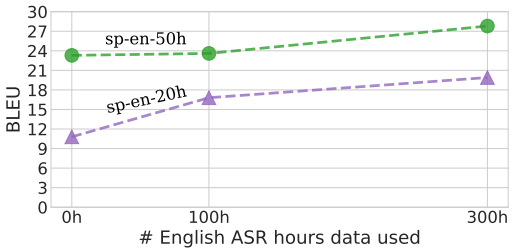


Figure 5: Spanish-to-English BLEU scores on Fisher dev set, with 0h (no transfer learning), 100h and 300h of English ASR data used.

time to deploy a working system must be minimized.

Amount of ASR data required. Figure 5 shows the impact of increasing the amount of English ASR data used on Spanish-English ST performance for two models: *sp-en-20h* and *sp-en-50h*.

For *sp-en-20h*, we see that using *en-100h* improves performance by almost 6 BLEU points. By using more English ASR training data (*en-300h*) model, the BLEU score increases by almost 9 points. However, for *sp-en-50h*, we only see improvements when using *en-300h*. This implies that transfer learning is most useful when only a few tens of hours of training data are available for ST. As the amount of ST training data increases, the benefits of transfer learning tail off, although it's possible that using even more monolingual data, or improving the training at the ASR step, could extend the benefits to larger ST data sets.

Impact of code-switching. We also tried using the *en-300h* ASR model without any fine-tuning to translate Spanish audio to English text. This model achieved a BLEU score of 1.1, with a precision of 15 and recall of 21. The non-zero BLEU score indicates that the model is matching *some* 4-grams in the reference. This seems to be due to code-switching in the Fisher-Spanish speech data set. Looking at the dev set utterances, we find several examples where the Spanish transcriptions match the English translations, indicating that the speaker switched into English. For example, there is an utterance whose Spanish transcription and English translation are both "right yeah", and this English expression is indeed present in the source audio. The English ASR model correctly translates this utterance, which is unsurprising since the phrase "right yeah" occurs nearly 500 times in Switchboard.

Overall, we find that in nearly 500 of the 4,000 development set utterances (14%), the Spanish transcription and English translations share more than half of their tokens, indicating likely code-switching. This suggests that transfer learning from English ASR models might help more than from other languages. To isolate this effect from transfer learning of language-independent speech features, we carried out a further experiment.

5.3 Using French ASR to improve Spanish-English ST

In this experiment, we pre-train using French ASR data for a Spanish-English translation task. Here, we can only transfer the speech encoder parameters, and there should be little if any benefit due to code-switching.

Because our French data set (20 hours) is much smaller than our English one (300 hours), for a fair comparison we used a 20 hour subset of the English data for pre-training in this experiment. For both the English and French models, we transferred only the encoder parameters.

Table 4 shows that both the English and French 20-hour pre-trained models improve performance on Spanish-English ST. The English model works slightly better, as would be predicted given our discussion of code-switching, but the French model is also useful, improving BLEU from 10.8 to 12.5. This result strengthens the claim that ASR pre-training on a completely distinct third language can help low-resource ST. Presumably benefits would be much greater if we used a larger ASR data set, as we did with English above.

In this experiment, the French pre-trained model used a French BPE output vocabulary, distinct from the English BPE vocabulary used in the ST system. In the future it would be interesting to try combining the French and English text to create a combined output vocabulary, which would allow transferring both the encoder and decoder parameters, and may be useful for translating names or cognates. More generally, it would also be possible to pre-train on multiple languages simultaneously using a shared BPE vocabulary. There is evidence that speech features trained on multiple languages transfer better than those trained on the same amount of data from a single language (Hermann and Goldwater, 2018), so multilingual pre-training for ST could improve results.

	baseline	+fr-20h	+en-20h
sp-en-20h	10.8	12.5	13.2

Table 4: Fisher dev set BLEU scores for *sp-en-20h*. baseline: model without transfer learning. Last two columns: Using encoder parameters from French ASR (+fr-20h), and English ASR (+en-20h).

model	pretrain	BLEU	Pr.	Rec.
fr-top-8w	–	0	23.5	22.2
fr-top-10w	–	0	20.6	24.5
en-300h	–	0	0.2	5.7
fr-20h	–	0	4.1	3.2
mb-fr-4h	–	3.5	18.6	19.4
	fr-20h	5.9	23.6	20.9
	en-300h	5.3	23.5	22.6
	en + fr	**7.1**	**26.7**	**23.1**

Table 5: Mboshi-to-French translation scores, with and without ASR pre-training. Pr. is the precision, and Rec. the recall score. fr-top-8w and fr-top-10w are *naive baselines* that, respectively, predict the 8 or 10 most frequent training words. For en + fr, we use encoder parameters from *en-300h* and attention+decoder parameters from *fr-20h*

6 Mboshi-French ST

Our final set of experiments test our transfer method on ST for the low-resource language Mboshi, where we have only 4 hours of ST training data: Mboshi speech input paired with French text output.

Table 5 shows the ST model scores for Mboshi-French with and without using transfer learning. The first two rows *fr-top-8w*, *fr-top-10w*, show precision and recall scores for the *naive baselines* where we predict the top 8 or 10 most frequent French words in the Mboshi-French training set. These show that a precision/recall in the low 20s is easy to achieve, although with no n-gram matches (0 BLEU). The pre-trained ASR models by themselves (next two lines) are much worse.

The baseline model trained only on ST data actually has lower precision/recall than the naive baseline, although its non-zero BLEU score indicates that it is able to correctly predict some n-grams. We see comparable precision/recall to the naive baseline with improvements in BLEU by transferring either French ASR parameters (both encoder and decoder, *fr-20h*) or English ASR parameters (encoder only, *en-300h*).

Finally, to achieve the benefits of both the larger training set size for the encoder and the matching language of the decoder, we tried transferring the encoding parameters from the *en-300h* model and the decoding parameters from the *fr-20h* model. This configuration (*en+fr*) gives us the best evaluation scores on all metrics, and highlights the flexibility of our framework. Nevertheless, the 4-hour scenario is clearly a very challenging one.

7 Conclusion

This paper introduced the idea of pre-training an end-to-end speech translation system involving a low-resource language using ASR training data from a higher-resource language. We showed that large gains are possible: for example, we achieved an improvement of 9 BLEU points for a Spanish-English ST model with 20 hours of parallel data and 300 hours of English ASR data. Moreover, the pre-trained model trains faster than the baseline, achieving higher BLEU in only a couple of hours, while the baseline trains for more than a day.

We also showed that these methods can be used effectively on a real low-resource language, Mboshi, with only 4 hours of parallel data. The very small size of the data set makes the task challenging, but by combining parameters from an English encoder and French decoder, we outperformed baseline models to obtain a BLEU score of 7.1 and precision/recall of about 25%. We believe ours is the first paper to report word-level BLEU scores on this data set.

Our analysis indicates that, other things being equal, transferring both encoder and decoder parameters works better than just transferring one or the other. However, transferring the encoder parameters is where most of the benefit comes from. Pre-training using a large ASR corpus from a mismatched language will therefore probably work better than using a smaller ASR corpus that matches the output language.

Our analysis suggests several avenues for further exploration. On the speech side, it might be even more effective to use multilingual training; or to replace the MFCC input features with pre-trained multilingual features, or features that are targeted to low-resource multispeaker settings (Kamper et al., 2015, 2017; Thomas et al., 2012; Cui et al., 2015; Yuan et al., 2016; Renshaw et al., 2015). On

the language modeling side, simply transferring decoder parameters from an ASR model did not work; it might work better to use pre-trained decoder parameters from a language model, as proposed by Ramachandran et al. (2017), or *shallow fusion* (Gülçehre et al., 2015; Toshniwal et al., 2018a), which interpolates a pre-trained language model during beam search. In these methods, the decoder parameters are independent, and can therefore be used on their own. We plan to explore these strategies in future work.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. This work was supported in part by a James S McDonnell Foundation Scholar Award, a Google faculty research award, and NSF grant 1816627. We thank Ida Szubert and Clara Vania for helpful comments on previous drafts of this paper and Antonios Anastasopoulos for tips on experimental setup.

References

Oliver Adams, Graham Neubig, Trevor Cohn, and Steven Bird. 2016a. Learning a translation model from word lattices. In *Proc. Interspeech*.

Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura. 2016b. Learning a lexicon and translation model from phoneme lattices. In *Proc. EMNLP*.

Tanel Alumäe, Stavros Tsakalidis, and Richard M Schwartz. 2016. Improved multilingual training of stacked neural network acoustic models for low resource languages. In *Proc. Interspeech*.

Antonios Anastasopoulos and David Chiang. 2017. A case study on using speech-to-translation alignments for language documentation. In *Proc. ACL*.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proc. NAACL HLT*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Low-resource speech-to-text translation. In *Proc. Interspeech*.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *Proc. ICASSP*.

Laurent Besacier, Bowen Zhou, and Yuqing Gao. 2006. Towards speech translation of non written languages. In *Proc. SLT*.

Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky. 2011. Rapid evaluation of speech representations for spoken term discovery. In *Proc. Interspeech*.

Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al. 2015. Multilingual representations for low resource speech recognition and keyword search. In *Proc. ASRU*.

Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero. 2013. Recent advances in deep learning for speech research at Microsoft. In *Proc. ICASSP*.

Yarin Gal. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proc. NIPS*.

Pierre Godard, Gilles Adda, Martine Adda-Decker, Juan Benjumea, Laurent Besacier, Jamison Cooper-Leavitt, Guy-Noël Kouarata, Lori Lamel, Hélène Maynard, Markus Müller, Annie Rialland, Sebastian Stüker, François Yvon, and Marcely Zanon Boito. 2018. A very low resource language speech corpus for computational language documentation experiments. In *Proc. LREC*.

John Godfrey and Edward Holliman. 1993. Switchboard-1 Release 2 (LDC97S62). https://catalog.ldc.upenn.edu/ldc97s62.

David Graff, Shudong Huang, Ingrid Cartagena, Kevin Walker, and Christopher Cieri. 2010. Fisher Spanish Speech (LDC2010S01). https://catalog.ldc.upenn.edu/ldc2010s01.

Caglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loıc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. ICCV*.

Enno Hermann and Sharon Goldwater. 2018. Multilingual bottleneck features for subword modeling in zero-resource languages. In *Proc. Interspeech*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*.

Aren Jansen, Kenneth Church, and Hynek Hermansky. 2010. Towards spoken term discovery at scale with zero resources. In *Proc. Interspeech*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. ACL*, 5:339–351.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. In *Proc. ICASSP*.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2017. A segmental framework for fully-unsupervised large-vocabulary speech recognition. *Comput. Speech Lang.*, 46:154–174.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proc. WMT*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*.

Lara J Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W Black. 2015. Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain. In *Proc. ASRU*.

Robert Munro. 2010. Crowdsourced translation for emergency response in Haiti: The global collaboration of local knowledge. In *AMTA Workshop Collaborative Crowdsourcing Transl.*

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proc. ICML*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

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*.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2014. Fisher and CALLHOME Spanish–English Speech Translation. https://catalog.ldc.upenn.edu/ldc2014t23.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *Proc. ASRU*.

Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proc. EMNLP*.

Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater. 2015. A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge. In *Proc. Interspeech*.

Tanja Schultz. 2002. Globalphone: a multilingual speech and text database developed at karlsruhe university. In *Seventh International Conference on Spoken Language Processing*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*

Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky. 2012. Multilingual mlp features for low-resource LVCSR systems. In *Proc. ICASSP*.

Sebastian Thrun. 1995. Is learning the n-th thing any easier than learning the first? In *Proc. NIPS*.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: A next-generation open source framework for deep learning. In *Proc. LearningSys*.

Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. 2018a. A Comparison of Techniques for Language Model Integration in Encoder-Decoder Speech Recognition. In *Proc. SLT*.

Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao. 2018b. Multilingual Speech Recognition with A Single End-To-End Model. In *Proc. ICASSP*.

Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz. 2012. An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance. In *Proc. Interspeech*.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly transcribe foreign speech. In *Proc. Interspeech*.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yougen Yuan, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li. 2016. Learning neural network representations using cross-lingual bottleneck features with word-pair information. In *Proc. Interspeech*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proc. EMNLP*.

5.3 Further analysis

In this section, we include further analysis and training procedure details for our pre-training method.

5.3.1 Impact of the number of speakers in the training set

In our published paper, we hypothesized that training on large amounts of ASR data may be helping the speech encoder generalize across speaker differences. There are around 130 different speakers in the Spanish ST corpus; around 500 in English ASR; 3 in the Mboshi ST, and 100 in the French ASR. These numbers suggest that the ASR data greatly increases the speaker variations which the speech encoder encounters during training.

We test the impact that the number of speakers in the training data (*#spkrs*) has on the translation performance. Our baseline model: *sp-en-20h* model achieved a BLEU score of 10.8. We created the training data for this model by randomly sampling 20 hours of speech from the entire 160 hours of the Fisher Spanish corpus, without controlling for the number of speakers. As a result, it consists of utterances from 136 speakers, a relatively large number. We now sample a different 20 hours of data where we limit *#spkrs* to less than 50.

Table 5.1 shows that without pre-training, when we increase the *#spkrs* in the training set from 50 to 136 (172% increase), the BLEU score improves from 7.2 to 10.8, a difference of 3.6 BLEU points (50%). With pre-training on English ASR, we are effectively increasing *#spkrs* from 550 to 636 (15% increase), and observe that the BLEU score improves from from 17.5 to 19.9 (around 14%).[1] This suggests that even if we have fewer speakers in our ST language pair leading to lower BLEU scores, ASR pre-training on a corpus with large number of speakers helps translation performance. This insight is especially important for endangered languages where it can be difficult to find more native speakers to translate speech data. Our results indicate that the improvement in BLEU scores due to pre-training, 7.2 to 17.5 BLEU points, is far greater than the improvement due to the addition of native speakers in the ST training set, 7.2 to 10.8 BLEU points.

[1] Calculated by adding together the *#spkrs* in the ASR and the ST training sets: 500 ASR +50 ST =550, and 500 ASR +136 ST =636.

# speakers	BLEU	
	Baseline	**+English ASR (500 speakers)**
50 speakers	7.2	17.5 (↑ 143%)
136 speakers	10.8 (↑ 50%)	19.9

Table 5.1: Fisher dev set BLEU for ST models trained on 20 hours of Spanish-English ST with and without pre-training on English ASR. **# speakers** is the number of speakers in the ST training data used. **+English ASR** includes English speech data from 500+ speakers.

5.3.2 What's improving? A closer look at precision and recall

To measure translation performance we calculate BLEU and unigram precision/recall scores, which are aggregated measures on an entire evaluation set. We have shown that these scores go up after pre-training on ASR, compared to baseline models, but this does not tell us much about what's improving in the predicted translations themselves. Here, we further analyze the translations produced by ST models to check if pre-training helps improve the prediction for rarely (or less frequently) occurring content words in the training set. This would be more advantageous than only observing improvements for frequently occurring word types. We perform this analysis for ST models trained on *sp-en-20h*.

There are a total of 7K/180K types/tokens in the *sp-en-20h* training subset. We only consider content word types, which we define as words that are more than five characters long and are not in the NLTK stopword lists. Table 5.2 shows the precision/recall scores for words of different frequencies: *frequent*, *medium*, *rare*, and *very rare*. We observe that the baseline model recall drops rapidly for medium frequency words, and for rare word types it has less than 5% recall. With pre-training, both precision and recall scores improve across the frequency categories. For medium frequency words, pre-training on English ASR improves precision from 26.1% to 48.3% and recall from 20.3% to 43%, around a 100% increase over the baseline for both these metrics. From this, we see that an important benefit of pre-training is in handling of less frequently occurring words.

	Very Rare	Rare	Medium	Frequent
training types	5K	1K	236	57
training tokens	12K	11K	11K	13K
Precision (%):				
baseline	11.8	16.2	26.1	36.6
+English ASR (300h)	26.7	30.6	48.3	59.4
+French ASR (20h)	12.7	17.6	32.8	42.9
Recall (%):				
baseline	0.4	4.8	20.3	41.4
+English ASR (300h)	6.3	20.5	43.0	57.4
+French ASR (20h)	1.5	8.4	26.4	45.9

Table 5.2: Content word frequency vs. dev precision/recall. *Very rare* words have $\leq$10 tokens per type in the training text; *rare* have 5–25 tokens; *medium* have 25–100 tokens; *frequent* have $\geq$100 tokens.

5.3.3 When and what to fine-tune?

After pre-training a neural model with ASR data, we switch to ST and fine-tune all the model parameters together (Erhan et al., 2010). These parameters include: CNN, encoder-LSTM, attention, decoder-embedding, decoder-LSTM and output-softmax layers, listed in order of first (closest to the input speech) to last (closest to the output). Recent work on transfer-learning has shown that controlling the order in which layer parameters are fine-tuned can improve performance. Felbo et al. (2017) propose a method called *chain-thaw*, where only one layer is fine-tuned at a time, keeping all others fixed. For our model architecture, this implies that we first fine-tune the CNN layers, keeping all others fixed, and then move up the chain of layers — fine-tune encoder LSTM next, and so on until finally fine-tuning the output-softmax layer. This potentially increases training time as each fine-tuning step can take a few epochs. Howard and Ruder (2018) propose an alternative method called *gradual unfreezing*, where they *freeze* (no weight updates permitted) all layer parameters except the last one, which is then fine-tuned for one epoch. In each successive epoch, an additional layer is unfrozen, until finally all parameters are allowed to be updated until convergence. We carry out experiments using the following fine-tuning schemes:

	Fine-tuning scheme		
ST model	**full**	**gradual**	**freeze-cnn**
es-en-2.5h	5.8	5.8	5.9
mb-fr-4h	7.1	5.7	7.7

Table 5.3: BLEU scores for ST models on the Fisher dev and Mboshi test sets. We fine-tune speech encoder parameters pre-trained on: English ASR data for Spanish-English ST; and French ASR for Mboshi-French ST. **full** implies that all model parameters are fine-tuned together; **gradual** fine-tunes an additional layer every epoch (Howard and Ruder, 2018); **freeze-cnn** fixes the CNN parameters for the first several epochs, and then fine-tunes the full model.

1. *gradual unfreezing* (Howard and Ruder, 2018) over the speech encoder parameters. We limit this experiment to the encoder as this is the more flexible pre-training setting where the ASR language can be different from both ST languages. In addition, we have observed that speech encoder parameters provide the largest gains during transfer-learning.

2. hybrid approach (*freeze-cnn*) combining aspects of *gradual unfreezing* and *chain-thaw* (Felbo et al., 2017), where we freeze the CNN layers for the first few epochs, and then let the full model train. We do this with the assumption that CNN layers, which take MFCC features as input, would have learned speech features after ASR pre-training, and we let fine-tuning focus on the translation task initially.

Table 5.3 shows the BLEU scores for Spanish-English and Mboshi-French ST models using these fine-tuning schemes. As the scores are quite similar, our takeaway is that more sophisticated fine-tuning methods might help improve scores marginally in certain cases, but overall the simple method of fine-tuning the entire model together works quite well for our datasets and task.

5.4 Follow-up work

Pre-training on Chinese ASR for Spanish-English ST. One of the main findings in our published paper was that ASR data from a language different than the ST language pair still helps improve performance: for example, French ASR improved Spanish-

	baseline	**+fr-20h**	**+en-20h**	**+zh-150h**	**+en-300h**
sp-en-20h	10.8	12.5	13.2	13.9	16.6

Table 5.4: Fisher dev set BLEU scores for *sp-en-20h*. baseline: model without transfer learning. Last four columns: Using encoder parameters from 20 hours of French ASR (+fr-20h), 20 hours of English ASR (+en-20h), 150 hours of Chinese ASR (+zh-20h) and 300 hours of English ASR (+en-300h) respectively.

English ST. However, in our experiments we were constrained by the relatively small size of the French ASR corpus (around 20 hours) and also, arguably, French is not as distinct from Spanish or English, as say Chinese. Recently, Stoian et al. (2019) carried out experiments to test whether Chinese ASR helps Spanish-English ST. They pre-trained an ASR model on 150 hours of Chinese speech (Aishell corpus; Hui Bu, 2017). For a fair comparison with the results we have presented so far, we take their pre-trained Chinese ASR model and fine-tune it on the same 20 hours of Spanish-English ST data used for all our experiments in this work. Table 5.4 shows the BLEU scores on the Fisher dev set. We see that fine-tuning the Chinese ASR model improves Spanish-English ST performance by 3 BLEU points, from 10.8 to 13.9. These results further strengthen our claim that using a larger corpus of ASR data from a third distinct language can help low-resource ST.

Analysing the linguistic knowledge captured by the neural model. Tian (2019) used ASR pre-training to improve translation for Swahili. They show that pre-training on English ASR helps improve Swahili-English ST and used a read speech corpus in their experiments.[2] They investigate why ASR pre-training helps ST and show evidence that the lower neural encoder layers capture phonology knowledge which is relatively agnostic to the input speech language, tested by switching between English and Swahili speech. Based on this analysis, they hypothesize that such knowledge, once acquired by pre-training on a high-resource language, is more easily transferable and helps low-resource ST.

[2]Swahili-English dataset available at `https://www.figure-eight.com/dataset/english-to-swahili-audio-recording-and-transcription`

Back-translation for ST. For text-to-text translation (MT), *back-translation* (Sennrich et al., 2016a) is an effective method to improve translation performance in low-resource scenarios by leveraging external monolingual data in the target text language. The external target text data is fed through an MT system trained in the back (or reverse) direction to generate additional synthetic training data. In the context of ST, back-translation would constitute training a cross-lingual text-to-speech synthesis (TTS) system. This is challenging as we expect a cross-lingual TTS system trained in low-resource settings to produce poor quality synthetic data compared to a back-translation system trained for MT. To the best of our knowledge, this hasn't been tried for ST.

Jia et al. (2019) used a method inspired by back-translation for English-Spanish ST. To train their system, they used a corpus of parallel English speech and Spanish text to build an ST baseline. They then used an additional corpus of parallel English-Spanish text to augment the ST training data, by using an English TTS system to synthesize the English speech from the Englisth text. This training set, composed of real and augmented ST data, improved the translation performance over the baseline by several BLEU points.

Though promising, the Jia et al. method relies on the availability of source language text, which is unavailable in low-resource settings of our interest. Here, as a simpler starting point than back-translation, we propose adapting the method of Currey et al. (2017) to leverage monolingual target text data. On an MT task, Currey et al. created additional "parallel" training data by copying target text as the source text. To apply this method for ST, where the encoder accepts speech input in place of text, we can use a TTS system to convert target language (high-resource) text into speech and use it for pre-training. In our Spanish-English ST scenario, this would mean using augmented English ASR data synthesized from English text for pre-training. Although English ASR (real speech) data is widely available, synthesizing English ASR data can help when English text from the same domain is available.

5.5 Review and next steps

Using our pre-training method, we were able to close the gap in translation performance between our ST model trained on 20 hours of Spanish-English data and the state-of-the-art Weiss et al. (2017) model trained on 160 hours. The gains observed were quite

encouraging and motivated us to explore even more challenging ST data scenarios than the lowest setting of 20 hours which we had tried before in Bansal et al. (2018) (Chapter 4). We showed that a pre-trained English ASR model fine-tuned on 2.5 hours of Spanish-English ST data was still able to outperform the naive baselines. Importantly, we found that transferring the encoder parameters is where most of the benefit comes from. Therefore, in scenarios where both languages in an ST pair are low-resource, a completely distinct third language (high-resource) can be used for pre-training the encoder parameters. We also observed faster ST training times for our models pre-trained on ASR, which can be especially useful in time-critical scenarios.

Although the BLEU scores have improved for our low-resource ST models, a manual review reveals that the translations produced are still of mediocre quality (Bansal et al., 2019, Table 3), so may not be suitable as complete and accurate sentence-level translations. In the next chapter, we show that there are useful real-world applications which can be built using ST systems, even if they produce poor translations (Kay, 1997; Church and Hovy, 1993). We are encouraged by the observation that although BLEU score of our 20 hour ST model remains low, the unigram word precision/recall scores are around 50%, compared to 70% for the state-of-the-art. This implies that the predicted text contains 50% of the tokens in the reference human text, many of which we expect to carry meaning and are not just stopwords.

Chapter 6

Applications for low-resource speech translation

6.1 Introduction

In Chapters 4 and 5, we showed that neural models produce only partially accurate translations when trained under simulated low-resource settings. This can be problematic in scenarios where automated translations are expected to be directly read and acted upon by humans. The performance requirements for such a system should be high, as a poor translation can potentially increase the workload of a user instead of aiding them.

However, providing *Fully Automatic High Quality Translation (FAHQT)* (Bar-Hillel, 1960) is not the only purpose of building ST systems. There are useful real-world applications where ST systems complement, not replace, human translators (Kay, 1997; Church and Hovy, 1993). For example, during disaster-recovery scenarios, even a poor ST system can potentially help human operators perform a quick first-pass analysis of large volumes of audio by translating it into text. Neural models are faster than real-time at making text predictions for audio (Chapter 4); for example, they can predict the English text for a 10 seconds Spanish speech utterance in around a second.[1] The English text can then be automatically scanned for detecting important keywords or predicting topics. While not 100% accurate, this entire process will be much faster than the current alternative where a human operator has to manually listen and process audio.

[1]There is further room for improvement as our implementation has not been optimized for speed.

Developing useful applications for low-resource languages, where accurate translation or speech recognition systems are not available, has been the focus of programs such as Low Resource Languages for Emergent Incidents (LORELEI), set up by DARPA, and more recently, the Open Cross Language Information Retrieval (OpenCLIR) task, set up by IARPA. These programs encourage building applications such as topic classification and keyword detection for speech or text in a low-resource language, using English topic names and queries respectively. We refer the reader to these program websites for more details.[2]

In this chapter, we explore two possible applications for low-resource ST:

1. **Classifying speech utterances by topic.** For a given speech utterance (around 1-minute duration), here our goal is to predict the topic of discussion. Such an application can be used to automatically cluster vast amounts of speech data into broad categories, potentially reducing the burden of having to manually listen and sort each audio file individually. We describe this experiment in Section 6.2 (included as an in-work publication), with additional analysis in Section 6.2.1.

2. **Keyword spotting.** Given a curated list of keywords in English, the goal here is to predict whether a speech utterance contains any of these keywords. This can be used to search for specific terms of interest in speech data. For example, a human operator can search for all telephone calls which mention keywords *medical* or *police*, in order to prioritize response. We describe this work in Section 6.3.

For consistency, we use the same experimental setup — ST models, training and evaluation splits — and notation for both sets of experiments.

In our work, we use a pipelined approach to build a topic classifier and keyword detection system that works on the output of the ST system. An alternative method would be to classify topics on speech data without using translations, as previously explored by Dredze et al. (2010) and Siu et al. (2014). They use unsupervised methods to convert speech data into a symbolic form, which is then used for topic classification and keyword discovery. These methods require speech to be directly annotated with topic labels or keywords. Therefore, if new topic labels or keywords emerge, the classification system will have to be re-trained. We do not explore these methods in

[2]LORELEI program details are available at: `https://www.darpa.mil/program/low-resource-languages-for-emergent-incidents`
OpenCLIR details available here: `https://www.nist.gov/itl/iad/mig/openclir-evaluation`

our work. In our case, by decoupling the speech-to-text component from downstream tasks, we make as few possible assumptions about the specifics of these tasks. Using our approach, if a user wants to expand the set of English keyword queries, they can simply search for them in the output of the ST system. This approach has also been taken by Sheridan et al. (1997) and Quinn and Hidalgo-Sanchis (2017).

To the best of our knowledge, there are few published examples of real-world applications built using low-resource ST systems, and therefore, it is challenging to forecast what level of performance we can expect out of them. One of the few examples, although in a monolingual setting, is the work by Quinn and Hidalgo-Sanchis (2017) who use an ASR system to quickly analyze radio content in Uganda. The work, conducted as part of a UN study, involved recording broadcast radio data and converting it to text using ASR. The text data was then processed further to spot keywords and detect topics of interest. The ASR systems used in this study were trained in very low-resource settings (Saeb et al., 2017), with less than 10 hours of transcribed speech data, and reported WER scores around 50%, indicating that the ASR output was noisy.

6.2 Paper: Cross-lingual topic prediction for speech using translations

Publication status. This work was originally formatted for submission to ACL 2019 where it was rejected. We are now in the process of re-submitting it to a different conference.

Contributions. The ideas presented in this paper were developed jointly involving all the co-authors. They also provided regular feedback on all results and helped identify areas for improvement. Each co-author also played a key role in the publication writing process.

My individual contributions in this work were developing the code base to train topic models on the text data; experimental setup; generating evaluation metrics and visualizations.

CROSS-LINGUAL TOPIC PREDICTION FOR SPEECH USING TRANSLATIONS

Sameer Bansal[1], *Herman Kamper*[2], *Adam Lopez*[1], *Sharon Goldwater*[1]

[1]School of Informatics, University of Edinburgh, UK
[2]Dept. E&E Engineering Stellenbosch University, South Africa

sameer.bansal@ed.ac.uk, {sgwater, alopez}@inf.ed.ac.uk
kamperh@sun.ac.za

ABSTRACT

Given a large amount of unannotated speech in a low-resource language, can we classify the speech utterances by topic? We consider this question in the setting where a small amount of speech in the low-resource language is paired with text translations in a high-resource language. We develop an effective cross-lingual topic classifier by training on just 20 hours of translated speech, using a recent model for direct speech-to-text translation. While the translations are poor, they are still good enough to correctly classify the topic of 1-minute speech segments over 70% of the time—a 20% improvement over a majority-class baseline. Such a system could be useful for humanitarian applications like crisis response, where incoming speech in a foreign low-resource language must be quickly assessed for further action.

Index Terms— speech translation, low-resource speech processing, speech classification, unwritten languages

1. INTRODUCTION

Quickly making sense of large amounts of linguistic data is an important application of language technology. For example, after the 2011 Japanese tsunami, natural language processing was used to quickly filter social media streams for messages about the safety of individuals, and to populate a person finder database [1]. Japanese text is high-resource, but there are many cases where it would be useful to make sense of *speech* in *low-resource* languages. For example, in Uganda, as in many parts of the world, the primary source of news is local radio stations, which is broadcast in many languages. A pilot study from the United Nations Global Pulse Lab identified these radio stations as a potentially useful source of information about a variety of urgent topics related to refugees, small-scale disasters, disease outbreaks, and healthcare [2, 3]. With many radio broadcasts coming in simultaneously, even simple classification of speech for known topics would be helpful to decision-makers working on humanitarian projects.

Speech classification systems have traditionally used automatic speech recognition (ASR) systems to first convert speech to text, which is then used as input to a classifier. However, this pipelined approach is impractical for unwritten languages, spoken by millions of people around the world. Although transcriptions cannot be obtained in these settings, translations could provide a viable alternative supervision source [4–7]. Recent research has shown that it is possible to train direct **S**peech-to-text **T**ranslation (ST) systems from speech paired only with translations [8–10]. Since no transcription is required, this is useful in very low-resource settings. However, in realistic low-resource settings where only a few hours of training data is available, these end-to-end ST systems produce poor translations [11]. But it has long been recognized that there are good uses for bad translations [12]. Could classifying the original speech be another one of these use cases?

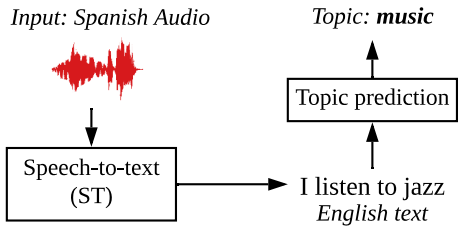


Fig. 1. Spanish speech is translated to English text, and a classifier then predicts its topic.

We answer this question affirmatively: we first use ST to translate speech to text, which we then classify by topic using supervised models (Figure 1). Although our ultimate goal is to work with truly low-resource languages, available datasets of this type are still too small to thoroughly evaluate and analyse. We therefore test our method on a corpus of conversational Spanish speech paired with English text translations that has been widely used in ST research [9, 13], enabling us to put our results in context. Using an ST model trained on 20 hours of Spanish-English data, we predict topics correctly 71% of the time, and we outperform the majority class baseline with less than 10 hours of training data. These promising results are the first we know of for this task, and open the door to future work on cross-lingual topic prediction from speech.

2. METHODS

Speech-to-text translation. We use the method of Bansal et al. [11] to train neural sequence-to-sequence Spanish-English ST models. As in that study, before training ST, we pre-train the models using English ASR data from the Switchboard Telephone speech corpus [14], which consists of around 300 hours of English speech and transcripts. In [11] this was found to substantially improve translation quality when the training set for ST was only tens of hours.

Topic modeling and classification. To classify the translated documents, we first need a set of topic labels, which were not already available for our dataset. We therefore initially discover a set of topics from the target-language (English) training text using a topic model. To classify the translations of the test data, we choose the most probable topic according to the learned topic model. To train our topic model, we use Nonnegative Matrix Factorization (NMF) [15, 16].We also experimented with Latent Dirichlet Allocation [17], but manual inspection revealed that NMF produced better topics.

3. EXPERIMENTAL SETUP

Data. We use the Fisher Spanish speech corpus [18], which consists of 819 phone calls, with an average duration of 12 minutes, giving a total of 160 hours of data. We discard the associated transcripts and pair the speech with English translations [19]. To simulate a low-resource scenario, we sampled 90 calls (20h) of data (*train20h*) to train both ST and topic models, reserving 450 calls (100h) to evaluate topic models (*eval100h*). We investigate ST models of varying quality, so we also trained models with decreasing amounts of data: *ST-10h*, *ST-5h*, and *ST-2.5h* are trained on 10, 5, and 2.5 hours of data, respectively, sampled from *train20h*. To evaluate ST only, we use the designated Fisher test set, as in previous work.

Fine-grained topic analysis. In the Fisher protocol, callers were prompted with one of 25 possible topics. It would seem appealing to use the prompts as topic labels, but we observed that many conversations quickly departed from the initial prompt and meandered from topic to topic. For example, one call starts: "Ok today's topic is marriage or we can talk about anything else ...." Within minutes, the topic shifts to jobs: "I'm working oh I do tattoos." To isolate different topics within a single call, we split each call into 1-minute long segments to use as 'documents'. This gives 1K training and 5.5K test segments, but leaves us with no human-annotated topic labels for them.

Obtaining gold topic labels for our data would require substantial manual annotation, so we instead use the human translations from the 1K (*train20h*) training set utterances to train the NMF topic model with *scikit-learn* [20], and then use this model to infer topics on the evaluation set. These *silver* topics act as an oracle: they tell us what a topic model would infer if it had perfect translations.

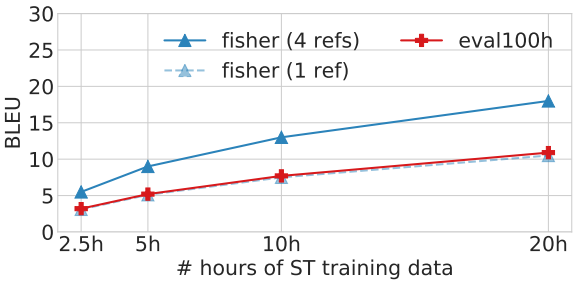


Fig. 2. BLEU scores for Spanish-English ST models computed on Fisher test set, using all 4 human references available, and using only 1 reference, and on *eval100h*, for which we have only 1 human reference.

To evaluate our ST models, we apply our ST model to test audio, and then predict topics from the translations using the NMF model trained on the human translations of the training data (Figure 1). To report accuracy we compare the predicted labels and silver labels, i.e., we ask whether the topic inferred from our predicted translation (ST) agrees with one inferred from a gold translation (human).

4. RESULTS

Spanish-English ST. To put our topic modeling results in context, we first report ST results. Figure 2 plots the BLEU scores on the Fisher test set and on *eval100h* for Spanish-English ST models. The scores are very similar for both sets when computed using a single human reference; scores are 8 points higher on the Fisher test set if all 4 of its available references are used. The state-of-the-art BLEU score on the Fisher test set is 47.3 (using 4 references), reported by [9], who trained an ST model on the entire 160 hours of data in the Fisher training corpus. By contrast, our 20 hour model (*ST-20h*) achieves a BLEU score of 18.1. Examining the translations (Table 1), we see that while they are mediocre, they contain words that might enable correct topic classification.

Topic modeling on training data. Turning to our main task of classification, we first review the set of topics discovered from the human translations of *train20h* (Table 2). We explored different numbers of topics, and chose 10 after reviewing the results. We assigned a name to each topic after manually reviewing the most informative terms; for topics with less coherent informative terms, we include *misc* in their names.

For evaluation, silver labels are obtained by applying this topic model to human translations on the test data. We argued above that the silver labels are sensible for evaluation despite not always matching the assigned call topic prompts, since they indicate what an automatic topic classifier would predict given correct translations and they capture finer-grained changes in topic. Table 3 shows a few examples where the silver

audio	yo eh oigo la música en inglés o americana
human	i eh listen to music in english or american
ST	i eh listen to the music in english
topic	*music*
audio	soy católica pero no en realidad casi no voy a la iglesia
human	i am catholic but actually i hardly go to church
ST	i'm catholics but reality i don't go to the church
topic	*religion*

Table 1. Examples of Spanish **audio** shown as Spanish text. An **ST** system translates the audio into English text, and we give the **human** reference. Our task is to predict the **topic** of discussion in the audio, which are potentially signaled by the underlined words.

Topic	Most informative terms
family-misc	married, kids, huh, love, three
music	music, listen, dance, listening, hear
intro-misc	hello, fine, name, hi, york
religion	religion, god, religions, believe, bible
movies-tv	movies, movie, watch, theater
welfare	insurance, money, pay, expensive
languages-misc	english, spanish, speak, learn
tech-marketing	phone, cell, computer, call, number
dating	internet, met, old, dating, someone
politics	power, world, positive, china, agree

Table 2. Topics discovered using human translated text from *train20h*, with manually-assigned topic names.

labels differ from the assigned call topic prompts. In the first example, the topic model was arguably incorrect, failing to pick up the prompt *juries*, and instead focusing on the other words, predicting *intro-misc*. But in the other examples the topic model is reasonable, correctly identifying the topic in the third example where the transcripts indicate that the annotation was wrong (specifying the topic prompt as *music*). In general, the topic model classifies a large proportion of discussions as *intro-misc* (typically at the start of the call) and *family-misc* (often where the callers stray from their assigned topic).

Our analysis also supports our observation that discussed topics stray from the prompted topic in most speech segments. For example, among segments in the 17 training data calls with the prompt *religion*, only 36% have the silver label *religion*, and the most frequently assigned label is *family-misc* (46%).

Topic classification on test data. We have four ST model translations: *ST-2.5h, 5h, 10h, 20h* (in increasing order of quality). We feed each each of the audio utterences in *eval100h* into the topic model from Table 2 to get the topic distribution and use the highest scoring topic as the predicted label.

human translation	Assigned	Silver
hello good afternoon have you ever been in a jury in a trial	juries	intro-misc
i also receive many letters of life insurance from banks	spam	welfare
they tell us we have to talk about marriage	music	family-misc

Table 3. Example audio utterances from *eval100h*. We show a part of the human translation here. **Assigned** is the topic assigned to speakers in the current call to prompt discussion. **Silver** is topic inferred by feeding the human translation through the topic model.

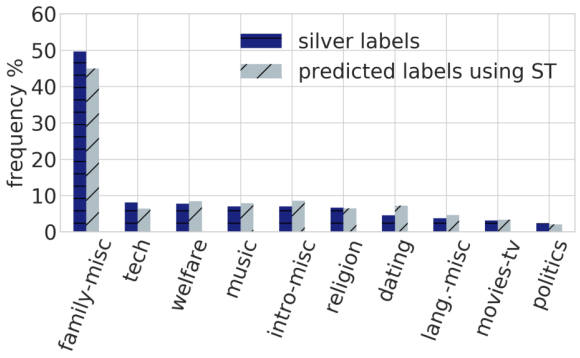


Fig. 3. Distribution of topics predicted for the 5K audio utterances in *eval100h*. **silver** labels are predicted using human translations. The **ST** model has been trained on 20 hours of Spanish-English data.

Figure 3 compares the frequencies of the silver labels with the predictions from the *ST-20h* model. The *family-misc* topic is predicted most often—almost 50% of the time. This is reasonable since this topic includes words associated with small-talk. Other topics such as *music*, *religion* and *welfare* also occur with a high enough frequency to allow for a reasonable evaluation.

Figure 4 shows the accuracy for all ST models, treating the silver topic labels as the correct topics. We use the *family-misc* topic as a majority class *naive baseline*, giving an accuracy of 49.6%. We observe that ST models trained on 10 hours or more of data outperform the *naive-baseline* by more than 10% absolute, with *ST-20h* scoring 71.8% and *ST-10h* scoring 61.6%. Those trained on less than 5 hours of data score close to or below that of the naive baseline: 51% for *ST-5h* and 48% for *ST-2.5h*.

Since topics vary in frequency, we look at label-specific accuracy to see if the ST models are simply predicting frequent topics correctly. Figure 5 shows a normalized confusion matrix for the *ST-20h* model. Each row sums to 100%, repre-

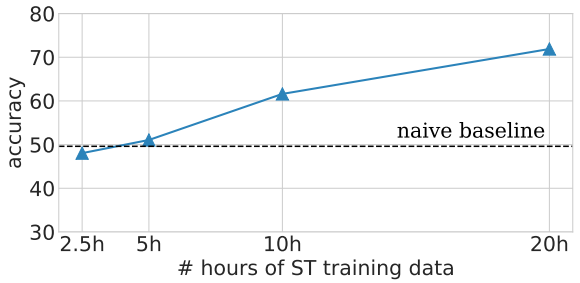


Fig. 4. Accuracy of topic prediction using ST model output. The **naive baseline** is calculated using majority class prediction, which is the topic *family-misc*.

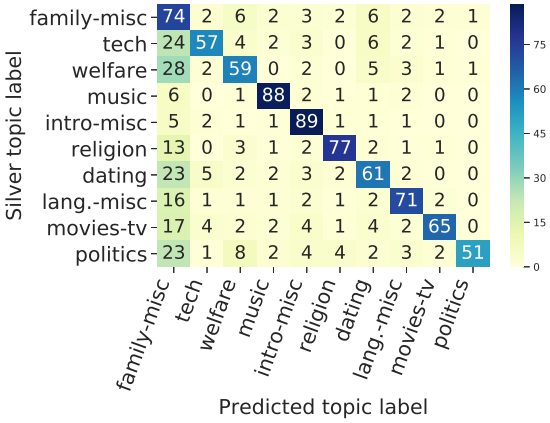


Fig. 5. Confusion matrix for ST model trained on 20 hours of Spanish-English data. Each cell represents the percentage of the silver topic labels predicted as the x-axis label, with each row summing to 100%.

senting the distribution of predicted topics for any given silver topic, so the numbers on the diagonal can be interpreted as the topic-wise recall. For example, a prediction of *music* recalls 88% of the relevant speech segments. We see that the model has a recall of more than 50% for all 10 topics, making it quite effective for our motivating task. The *family-misc* topic (capturing small-talk) is often predicted when other silver topics are present, with, for instance, 23% of the silver *dating* topics predicted as *family-misc*.

5. RELATED WORK

We have shown that low-quality ST can be useful for speech classification. Previous work has also looked at speech analysis without high-quality ASR. In a task quite related to ours, [21] showed how to cluster speech segments in a completely unsupervised way. In contrast, we learn to classify speech using supervision, but what is important about our result is it shows that a small amount of supervision goes a long way.

A slightly different approach to quickly analyse speech, is the established task of *keyword spotting*, which asks whether any of a specific set of keywords appears in a segment [22, 23]. Recent studies have extended the early work to end-to-end keyword spotting [24,25] and to semantic keyword retrieval, where non-exact but relevant keyword matches are retrieved [26–28]. In all these studies, the query and search languages are the same, while we consider the cross-lingual case.

There has been some limited work on cross-lingual keyword spotting. [29] introduced a baseline system which combined ASR and text translation to build a German speech retrieval system using French text queries. But source language transcriptions to train ASR are unlikely to be available in our scenarios of interest. Some recent studies have attempted to use vision as a complementary modality to do cross-lingual retrieval [30,31]. However, to the best of our knowledge, cross-lingual topic classification for speech has not been considered elsewhere.

6. CONCLUSIONS AND FUTURE WORK

Our results show that poor speech translation can still be useful for speech classification in low-resource settings. By varying the amount of training data, we found that ST systems trained on as little as 10 hours (around 8K parallel utterances) of Spanish-English data produce translations which still allow topics to be correctly classified in 61% of input speech segments, outperforming a majority baseline. With 20 hours of parallel data, accuracy is more than 70%.

Since this is the first work in cross-lingual topic classification, there are a number of interesting avenues for future work. We used our ST model as an off-the-shelf system, and did not tune its performance for the topic prediction task. We hope future work will improve accuracy further. We used silver labels to evaluate our approach—this allowed us to compare several different settings using an objective metric. However, human annotations of topics will be the next step. We also used a pipelined approach of ST followed by classification. An alternative would be to train a topic classifier on input speech directly, but we speculate that this would require more substantial resources. Cross-lingual topic modeling may also be useful when the target language is high-resource; we learned target topics just from the 20 hours of translations, but in future work, we could use a larger text corpus in the high-resource language to learn a more general topic model covering a wider set of topics, and/or combine it with keyword lists curated for specific scenarios like disaster recovery [32].

7. ACKNOWLEDGEMENTS

This work was supported in part by a James S McDonnell Foundation Scholar Award for SG and a Google Faculty Research Award for HK. We thank Ida Szubert, Marco Damonte, and Clara Vania for helpful comments on drafts of this paper.

8. REFERENCES

[1] G. Neubig, Y. Matsubayashi, M. Hagiwara, and K. Murakami, "Safety information mining—what can NLP do in a disaster—," in *Proc. IJCNLP*, 2011.

[2] J. Quinn and P. Hidalgo-Sanchis, "Using machine learning to analyse radio content in Uganda: Opportunities for sustainable development and humanitarian action," United Nations Global Pulse Lab Kampala, Tech. Rep., 2017. [Online]. Available: http://air.ug/~jquinn/papers/UNGP_radio_analysis_report_2017.pdf

[3] R. Menon, H. Kamper, E. Van Der Westhuizen, J. Quinn, and T. R. Niesler, "Feature exploration for almost zero-resource asr-free keyword spotting using a multilingual bottleneck extractor and correspondence autoencoders," in *Proc. Interspeech*, 2019.

[4] S. Bird, L. Gawne, K. Gelbart, and I. McAlister, "Collecting bilingual audio in remote indigenous communities," in *Proc. COLING*, 2014.

[5] D. Blachon, E. Gauthier, L. Besacier, G.-N. Kouarata, M. Adda-Decker, and A. Rialland, "Parallel speech collection for under-resourced language studies using the Lig-Aikuma mobile device app," *Procedia Computer Science*, vol. 81, pp. 61–66, 2016.

[6] G. Adda, S. Stüker, M. Adda-Decker, O. Ambouroue, L. Besacier, D. Blachon, H. Bonneau-Maynard, P. Godard, F. Hamlaoui, D. Idiatov *et al.*, "Breaking the Unwritten Language Barrier: The BULB project," *Procedia Computer Science*, vol. 81, pp. 8–14, 2016.

[7] L. Besacier, B. Zhou, and Y. Gao, "Towards speech translation of non written languages," in *Proc. SLT*, 2006.

[8] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop on end-to-end learning for speech and audio processing.*, 2016.

[9] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," in *Proc. Interspeech*, 2017.

[10] S. Bansal, H. Kamper, A. Lopez, and S. Goldwater, "Towards speech-to-text translation without speech recognition," in *Proc. EACL*, 2017.

[11] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proc. NAACL*, 2019.

[12] K. W. Church and E. H. Hovy, "Good applications for crummy machine translation," *Machine Translation*, vol. 8, no. 4, pp. 239–258, 1993.

[13] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-resource speech-to-text translation," in *Proc. Interspeech*, 2018.

[14] J. Godfrey and E. Holliman, "Switchboard-1 Release 2 (LDC97S62)," 1993.

[15] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, 2007.

[16] S. Arora, R. Ge, and A. Moitra, "Learning topic models–going beyond svd," in *Proc. FOCS*, 2012.

[17] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, 2003.

[18] D. Graff, S. Huang, I. Cartagena, K. Walker, and C. Cieri, "Fisher Spanish Speech (LDC2010S01)," 2010.

[19] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved speech-to-text translation with the Fisher and Callhome Spanish-English speech translation corpus," in *Proc. IWSLT*, 2013.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, 2011.

[21] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proc. EMNLP*, 2010.

[22] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 11, pp. 1870–1878, 1990.

[23] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. ICASSP*, 2006.

[24] D. Palaz, G. Synnaeve, and R. Collobert, "Jointly learning to locate and classify words using convolutional networks," in *Proc. Interspeech*, 2016.

[25] R. Menon, H. Kamper, J. Quinn, and T. Niesler, "Fast ASR-free and almost zero-resource keyword spotting using DTW and CNNs for humanitarian monitoring," in *Proc. Interspeech*, 2018.

[26] C. Chelba, T. J. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Proc. Mag.*, vol. 25, no. 3, 2008.

[27] Y.-C. Li, H.-y. Lee, C.-T. Chung, C.-a. Chan, and L.-s. Lee, "Towards unsupervised semantic retrieval of spoken content with query expansion based on automatically discovered acoustic patterns," in *Proc. ASRU*, 2013.

[28] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval—beyond cascading speech recognition with text retrieval," *IEEE Trans. Audio, Speech, Language Process.*, vol. 23, no. 9, pp. 1389–1420, 2015.

[29] P. Sheridan, M. Wechsler, and P. Schäuble, "Cross-language speech retrieval: Establishing a baseline performance," in *Proc. SIGIR*, 1997.

[30] H. Kamper and M. Roth, "Visually grounded cross-lingual keyword spotting in speech," in *Proc. SLTU*, 2018.

[31] D. Harwath, G. Chuang, and J. Glass, "Vision as an interlingua: Learning multilingual semantic embeddings of untranscribed speech," in *Proc. ICASSP*, 2018.

[32] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "CrisisLex: A lexicon for collecting and filtering microblogged communications in crises," in *Proc. ICWSM*, 2014.

[33] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2001.

A. USING NMF FOR TOPIC MODELING

We now describe how we learn topics using NMF. Given a set of text documents as input, the model will output (1) for each document, a distribution over the selected number of topics (henceforth, the *document-topic* distribution), and (2) for each topic, a distribution over the set of unique terms in the text (henceforth, the *topic-term* distribution).

A.1. Text processing

Our training set (*train20h*) has 1080 English sentences. We start by generating a *tf-idf* representation for each of these. The English text contains 170K tokens and 6K terms (vocabulary size). As we are looking for topics which are coarse-level categories, we do not use the entire vocabulary, but instead focus only on the high importance terms. We lowercase the English translations and remove all punctuation, and stopwords. We further remove the terms occurring in more than 10% of the documents and those which occur in less than 2 documents, keeping only the 1000 most frequent out of the remaining.

After preprocessing the training set, we have a feature matrix V with dimensions 1080×1000, where each row is a document, and each column represents the *tf-idf* scores over the 1000 selected terms. The feature matrix will be sparse as only a few terms would occur in a document, and will also be non-negative as *tf-idf* values are greater than or equal to 0.

A.2. Learning topics

NMF is a matrix factorization method, which given the matrix V, factorizes it into two matrices: W with dimensions $1080 \times t$ (long-narrow), and H with dimensions $t \times 1000$ (short-wide), where t is a hyper-parameter. Figure 6 shows this decomposition when t is set to 10.

$$V \approx W \times H$$

In the context of topic modeling, t is the number of topics we want to learn; W is the *document-topic* distribution, where for each document (row) the column with the highest value is the most-likely topic; and H is the *topic-term* distribution, where each row is a topic, and the columns with the highest values are terms most relevant to it.

The values for W and H are numerically approximated using a multiplicative update rule [33], with the Frobenius norm of the reconstruction error as the objective function. In this work, we use the machine-learning toolkit *scikit-learn* [20] for feature extraction, and to perform NMF, using default values as described at scikit-learn.org.

A.3. Making topic predictions

Using our *topic-term* distribution matrix H, we can now make topic predictions for new text input. Our evaluation set

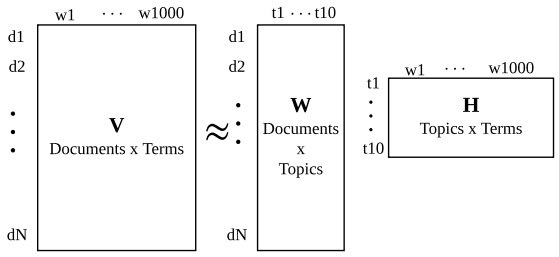


Fig. 6. Nonnegative Matrix Factorization. V is the document-term matrix, where d is each document; N is the number of documents; $w1$ to $w1000$ are the terms selected as features; and $t1$ to $t10$ are the topics.

(*eval100h*) has 5376 English sentences. For each of these, we have the *gold* text, and also the ST model output. We preprocess and represent these using the same procedure as before (A.1) giving us the feature matrix $V^{'}_{gold}$ for *gold*, and $V^{'}_{ST}$ for ST output, each with dimensions 5376×1000. Our goal is to learn the *document-topic* distributions $W^{'}_{gold}$ and $W^{'}_{ST}$, where:

$$V^{'}_{gold} \approx W^{'}_{gold} \times H$$

$$V^{'}_{ST} \approx W^{'}_{ST} \times H$$

The values for each $W^{'}$ matrix are again numerically approximated using the same objective function as before, but keeping H fixed.

A.4. Silver labels and evaluation

We use the highest scoring topic for each document as the prediction. The *silver* labels are therefore computed as $argmax(W^{'}_{gold})$, and for ST as $argmax(W^{'}_{ST})$. We can now compute the accuracy over these two sets of predictions.

B. FISHER CORPUS: ASSIGNED TOPICS

Figure 7 shows the topics assigned to callers in the Fisher speech corpus. Some topic prompts overlap, for example, *music-preference* asks callers to discuss what kind of music they like to listen to, and *music-social-message* asks them to discuss the social impact of music. For both these topics, we would expect the text to contain similar terms. Similarly the topics *cellphones-usage*, *tech-devices* and *telemarketing-spam* also overlap. Such differences might be difficult for an unsupervised topic modeling algorithm to pick up.

Table 4 shows the topics learned by NMF by using human English translations from the entire 160 hours of training data as input, when the number of topics is set to 25. We observe that some new topics are found that were not discovered by the 20hr/10-topic model and that match the assigned topic prompts,

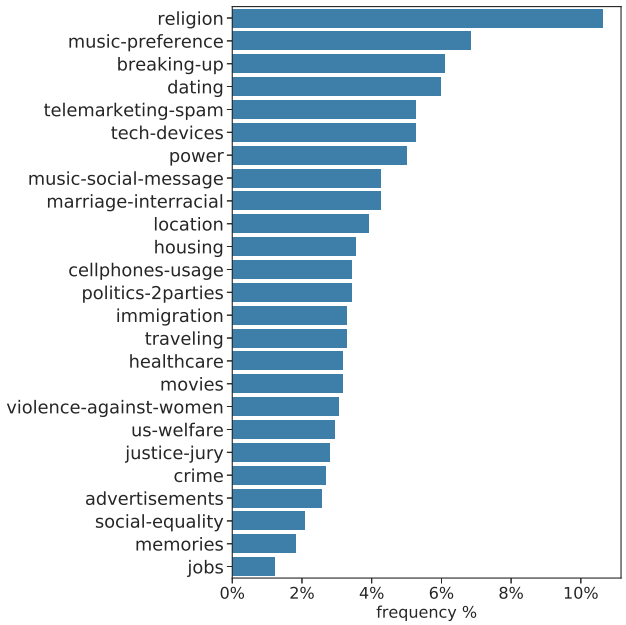


Fig. 7. Topics assigned to callers in the Fisher dataset, as a percentage of the 819 calls.

such as *juries* and *housing*. However, there are also several incoherent topics, and we don't find a major improvement over the topics learned by just using 20 hours of training data, with the number of topics set to 10.

C. TRACKING TOPIC DRIFT OVER CONVERSATIONS

To measure how often speakers stray from assigned topic prompts, we take a closer look at the calls in *train20h* with the assigned prompt of *religion*. This is the most frequently assigned prompt in the Fisher dataset (17 calls in *train20h*). We also select this topic for further analysis as it contains terms which are strongly indicative, such as *god*, *bible*, etc. and should be relatively easier for our topic model to detect.

Figure 8 shows the trend of discussion topics over time. Overall, only 36% of the total dialog segments in these calls have the silver label *religion*, and the most frequently assigned label is *family-misc* with 46%. We observe that the first segment is often labeled as *intro-misc*, around 70% of the time, which is expected as speakers begin by introducing themselves. Figure 9 shows that a similar trend emerges for calls assigned the prompt *music* (14 calls in *train20h*). Silver labels for *music* account for 45% of the call segments and *family-misc* for around 38%.

id	Assigned name	Most informative words
1	—	told, went, maybe, take, ll
2	music	music, listen, dance, play, classical
3	intro	hello, name, speaking, topic, talked
4	religion	religion, religions, catholic, church, religious
5	welfare	pay, insurance, expensive, doctor, health
6	languages	spanish, speak, english, language, learn
7	relationships	married, marriage, got, divorced, together
8	tech-marketing	phone, cell, telephone, calls, cellular
9	—	hundred, dollars, thousand, five, fifty
10	chatter	cold, snow, winter, hot, weather
11	—	puerto, rico, rican, born, ricans
12	movies-tv	watch, movies, movie, tv, kids
13	—	city, mexico, big, lived, living
14	—	huh, gonna, give, us, lets
15	—	yea, tv, lots, pretty, expensive
16	locations	york, manhattan, bronx, carolina, panama
17	internet-dating	internet, computer, use, met, information
18	—	old, twenty, kids, thirty, five
19	politics	power, countries, world, government, help
20	housing	house, buy, rent, apartment, houses
21	juries	system, jury, health, social, help
22	religion	god, believe, church, bible, thank
23	violence	women, man, woman, men, abuse
24	intro	hi, fine, name, philadelphia, evening
25	welfare	money, give, make, help, need

Table 4. Topics discovered using human translated text from the full 160hr Fisher training set. We set the number of topics to 25. We assign the topic names manually, and use — where the topic clustering is not very clear.

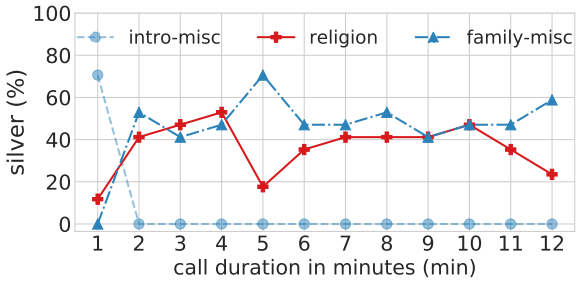


Fig. 8. Tracking silver labels over time for calls where the assigned prompt is *religion*. Total of 17 calls in *train20h*.

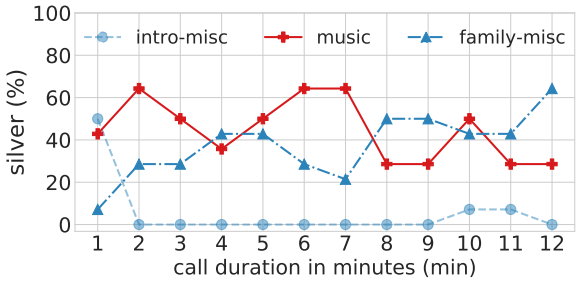


Fig. 9. Tracking silver labels over time for calls where the assigned prompt is *music*. Total of 14 calls in *train20h*.

id	Assigned name	Most informative words
1	—	exactly, person, true, would, many
2	music	music, listen, dance, type, classical
3	intro-misc2	hello, name, fine, philadelphia, evening
4	politics	states, united, country, countries, puerto
5	religion	religion, religions, church, catholic, catholics
6	—	new, york, city, university, okay
7	—	old, twenty, two, five, married
8	languages-misc	spanish, english, speak, okay, puerto
9	tech-marketing	call, phone, telephone, cellular, number
10	—	able, wouldn, get, would, hang

Table 6.1: Topics discovered using English translations produced by *ST-20h* on 100 hours of speech data (*eval-100h*). We set the number of topics to 10 and manually assign the topic names; "—" is used where the topic clustering is not very clear.

6.2.1 Topic modeling on predicted text

The topic model used in our experiments was trained on English text translations generated by human annotators from 20 hours of Spanish speech data (*train-20h*). As a result we can only detect topics in future/unseen speech data which occurs in this relatively small training set. For example, given the English text: *"hello good afternoon have you ever been in a jury in a trial"*, our system predicts the topic *intro-misc* based on the terms *hello, good, afternoon*; it is unable to predict the assigned discussion prompt *juries*, which it never learnt as it occurs rarely (around 4% of speech) in *train-20h*. To learn new topics we would require additional speech data to be translated.

We test whether we can use predicted translations (using ST) instead of human annotations to learn the topic model. Here, we don't have to limit ourselves to only 20 of speech data, and can potentially train a topic model on all available speech. Table 6.1 shows the topics discovered using NMF on predicted text from the 100 hours of Spanish speech data in *eval-100h* set. We observe that only a few coherent topics emerge, such as *music* and *religion*, whereas the majority are composed of loosely connected terms. In addition, no new topics such as *juries* were discovered. Our takeaway is that a small quantity of high-quality translations are essential to learn a good topic classifier, compared to a larger quantity (up to 5 times more in this case) of poor quality text which is too noisy to learn from.

6.3 Cross-lingual keyword detection in speech

With topic prediction, we were assigning speech utterances to broad thematic categories such as *music* and *religion*. This clusters together utterances which contain different, but related terms. In certain situations, however, a human operator may want to search for specific terms in a large volume of audio data. For example, instead of all music related speech utterances (around 500 in *eval-100h*) they would only want to retrieve those which mention *jazz* (around 16). This is a well established task in monolingual settings — speech and query text are in the same language — and is referred to as *Keyword spotting* (Wilpon et al., 1990; Garcia and Gish, 2006; Menon et al., 2018). However, there has been limited work on cross-lingual keyword spotting. Sheridan et al. (1997) introduced a baseline system which combined ASR and text translation to build a German speech retrieval system using French text queries. To the best of our knowledge, there has been no published work on a cross-lingual keyword spotting system trained without using source language text (or using ASR).

Our aim in this work is to build a baseline cross-lingual keyword spotting system using ST. We test whether we can reliably detect the presence of selected keywords in Spanish speech utterances using English text queries. As in our topic prediction study, we convert a speech utterance into English text using ST, and then use the text for keyword detection. The detection method itself is a simple string match check for each of the keywords (Figure 6.1). For this task, we use a list of English keywords selected due to their frequent use during disaster scenarios. This list was created by Olteanu et al. (2014) — referred to as *CrisisLex* — and contains keywords such as: *victims, flood crisis, police people, rescuers, send help*. We target keywords which occur rarely during both training and evaluation, and therefore, the majority of speech utterances will have no keywords present.

6.3.1 Experimental setup

CrisisLex consists of 380 terms, around 300 of which are two words long phrases such as *send help*. We were constrained in our choice of keywords to use in this study, as crisis related terms are out-of-domain for our speech dataset; which consists of telephone conversations on general topics such as music, religion, and politics. Therefore most of these 380 terms are either not present or extremely rare. Nevertheless, we were able to

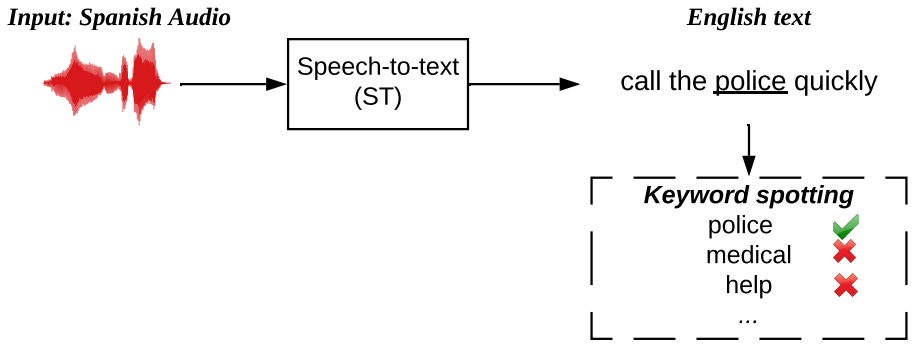


Figure 6.1: Spanish speech is translated to English text, and string matching is used to detect presence of keywords: *police, medical, etc.*.

families	community	please	water	town
waiting	kill	strong	police	news
cost	medical	service	public	lost
high	terrible	government	found	free
power	number	change	women	send

Table 6.2: Selected keywords from *CrisisLex* used for keyword spotting.

include relevant keywords such as *medical* and *police*, using the following criteria. We split these phrases into individual words: *send help* gives us two keywords *send* and *help*. We discard any keyword with more than 50 tokens in *train-20h* (total # of tokens is 168K). We also filter out keywords which occur very rarely (less than 10 tokens).[3] From the remaining, we manually select 25 keywords for our analysis. These are shown in Table 6.2. For each keyword, the token count as a percentage of the total number of tokens in the ST training set is around 0.05%, suggesting that the ST model would have seen few examples for these keywords during training.

6.3.2 Evaluation

We frame keyword spotting as a binary label task, where a *true* or *false* label indicates the presence or absence respectively for a keyword in the text translation for a given speech segment. We compute the ground-truth label using the human translations; the predicted label is determined using the ST model output from *ST-20h*.

[3]We do not check the token count in the English ASR data used for pre-training our ST model.

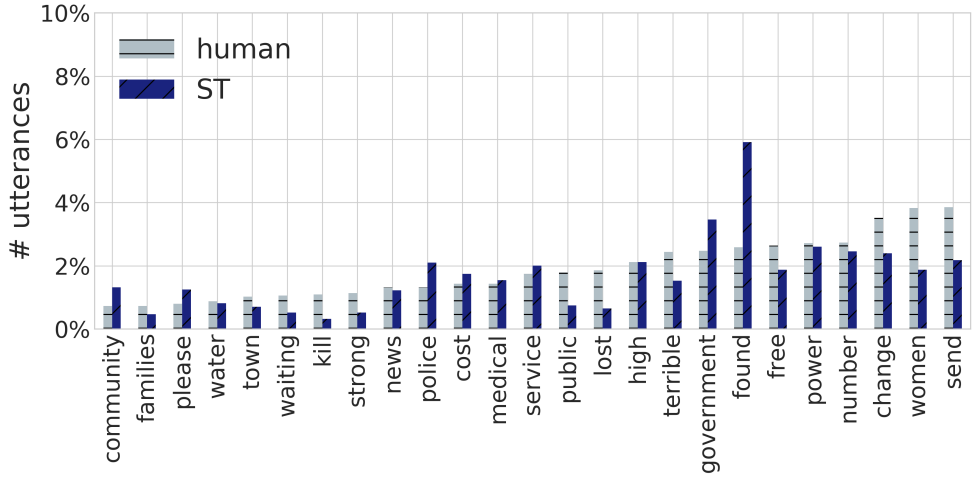


Figure 6.2: Keyword frequency (%) distribution over the 5K audio utterances in *eval-100h*. **human** denotes *#utts* computed using human annotated English translations; and **ST** using predicted text.

Figure 6.2 plots the number of utterances (*#utts*) which contain each selected keyword in *eval-100h* using both the human annotated and the ST model predicted text. The average *#utts* for each keyword is around 100, or 2% out of the total 5K utterances.

For evaluation, we compare the predicted and the ground-truth labels and compute individual F1 (weighted precision and recall) scores for each of the selected keywords. In the topic prediction task we computed an accuracy score, but do not use it here as it is a poor metric when there is a large imbalance between the number of true and false ground-truth labels. For example, there are 71 utterances which contain the keyword *police*. A system which predicts false for each of the 5K utterances (*naive-false*), achieves an accuracy of 98.5%, but a recall of 0%. Similarly, predicting true everytime (*naive-true*) achieves a recall of 100%, but poor precision. Therefore, to get a balanced view of the prediction performance we compute the F1 score:

$$F1 = \frac{2 \times (precision \times recall)}{(precision + recall)}$$

The F1 score is around 0% for *naive-false* and 3% for *naive-true*, indicating that these systems are of little practical value. We use *naive-true* for comparison in our experiments, as it is stronger compared to both *naive-false* and a random baseline.

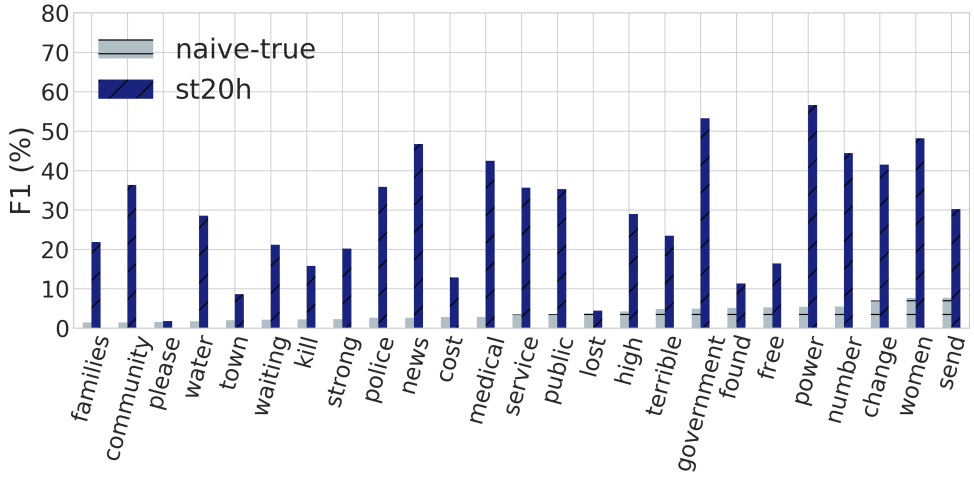


Figure 6.3: F1 scores for Keyword spotting using ST model output. *naive-true* is a baseline which always predicts true.

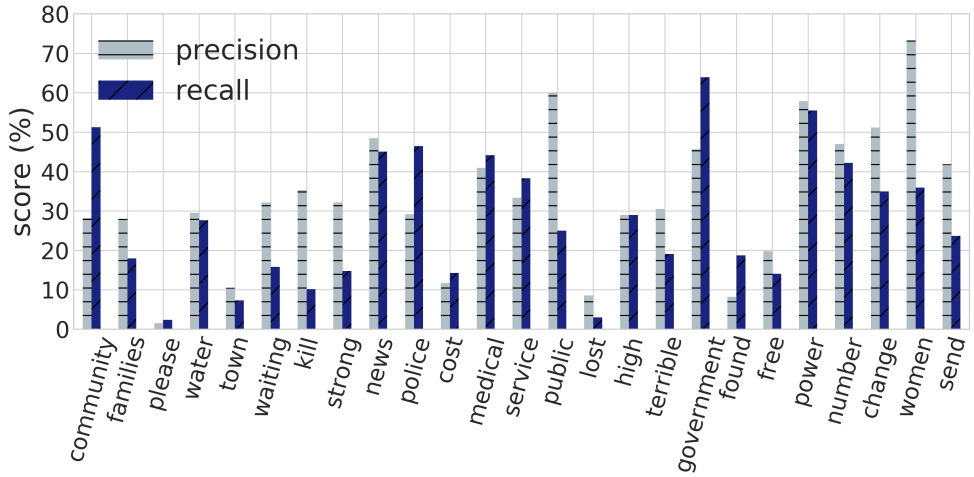


Figure 6.4: Precision and recall scores for Keyword spotting using ST model output.

6.3.3 Results

Figures 6.3 and 6.4 show the F1, and precision/recall scores for our ST model for each selected keyword. We observe an F1 score greater than 30% for several of the keywords. To help interpret these scores, consider the keyword *medical* which occurs in only 77 utterances (out of 5K) in the evaluation set. The ST output achieves an F1 score of 42% for *medical*, with a precision of 41% and recall of 44%. This implies that the ST model correctly identifies 34 out of 77 utterances (*true-positives*) which contain this keyword; and incorrectly labels 49 utterances as true (*false-positives*). Arguably, the benefit of quickly identifying relevant audio utterances, justifies the cost of filtering out false-positives.

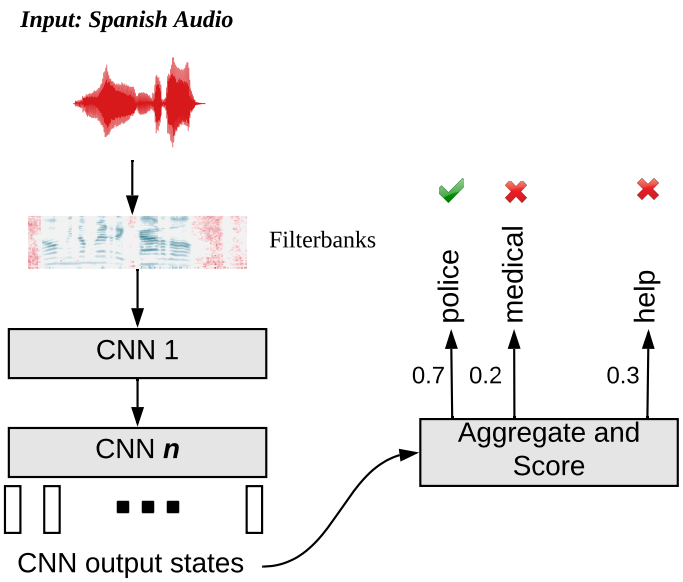


Figure 6.5: Palaz et al. (2016) model for Keyword spotting.

6.3.4 Discussion and future work

Our results show that noisy speech translations can still be used to detect keywords in speech utterances and serve as a preliminary baseline for this task. In this study, we used our ST model as an off-the-shelf system, and did not tune its performance for the keyword spotting task. An alternative would be to train a classifier on input speech to directly predict keywords, as described by Palaz et al. (2016) (Figure 6.5). In their model, referred to as *Palaz-KeySpot*, speech (represented using filterbank features) is fed into a deep neural network composed of successive CNN layers. In the final step (*Aggregate and Score*), a prediction score is computed for each of their target 1K keywords (pre-selected). To make a *true* or *false* prediction, they threshold the score for each keyword, with the threshold value determined using tuning on a development set. The advantage of this approach is that it allows for finer control over the prediction decision. For example, to prioritize precision over recall, we could set the prediction threshold to a higher value (reducing false-positives, but also true-positives); to prioritize recall, we would lower the threshold (increasing true-positives, but also false-positives).

Palaz et al. (2016) demonstrated the effectiveness of their model on a monolingual (English) keyword spotting task using around 1000 hours of read speech for training. In comparison, our ST model was trained on only 20 hours of telephone quality Spanish-English ST data. An important direction for future work would be to train *Palaz-KeySpot* on our dataset and compare the performance against our baseline results. Although we

do not explore this experimental setting in detail, we next discuss a few preliminary experiments which we carried out.

We used the *Palaz-KeySpot* model to train a keyword spotting system on the entire 160 hours of Spanish-English ST data. However, these results were not encouraging, with the classifier achieving lower precision/recall scores compared to our ST pipeline based method described in Section 6.3. When trained in the 20 hours low-resource ST setting, the classifier performance further deteriorated to a very low precision/recall of less than 10%. We speculate that the poor performance we observed may be due to the differences in our experimental setup compared to Palaz et al. (2016), such as the lower quality of speech utterances in the Spanish corpus (conversation vs read speech), smaller training set size and the cross-lingual setting we explore.

To improve performance, instead of training the *Palaz-KeySpot* model from scratch, it might be even more effective to use transfer learning from the ST model: replace the input speech features (filterbanks or MFCCs) with the ST speech encoder output (LSTM hidden states). Or, we can jointly train the ST model and keyword spotting parameters using multitask learning (Caruana, 1997), with several configurations possible. For example, we can share the CNN layers which take speech as input, in both the ST and *Palaz-KeySpot* models; or we can add *Palaz-KeySpot* specific layers on top of the ST speech encoder parameters as shown in Figure 6.6.

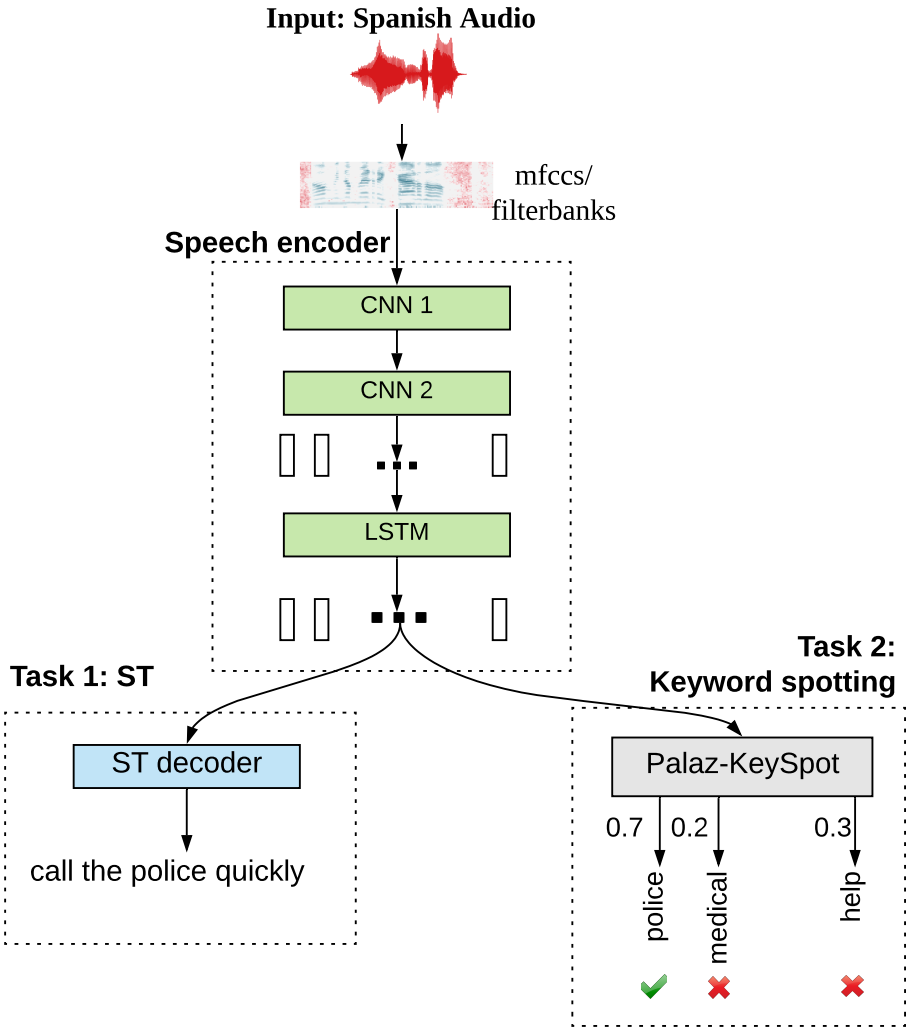


Figure 6.6: Multitask learning for ST and Keyword spotting tasks. The speech *Speech encoder* parameters: CNN and LSTM, are shared.

Chapter 7

In Summary

In the previous chapters we explored the feasibility of building a direct speech-to-text translation system given only a few hours of speech paired with text translations.

Our overall goal was to explore methods to build ST systems for unwritten languages, which constitute about 40% of the total number of spoken languages in the world (Ethnologue, 2019c). The lack of source language text for training implied that we could not use the conventional ASR and MT pipeline for ST. We therefore explored building direct ST systems given only a few hours of speech data paired with text translations for training. At the start of our research, it was unclear whether a speech translation system could be built under these conditions and we were not sure what level performance we could expect. Over the course of our work, we conducted an in-depth study for the task of direct ST across a range of training data settings on a realistic multi-speaker speech corpus. We found that useful ST systems can be built for unwritten languages using neural models and their performance can be further improved using data from high-resource languages.

For our first attempt at building an ST system (Chapter 3) we used a state-of-the-art software library for speech pattern detection (*ZRTools*) and tested it on a telephone speech corpus. We discovered that ZRTools struggles to discover patterns in multi-speaker speech data recorded in realistic noise conditions which in turn leads to poor downstream translation performance. Follow up analysis revealed that this ST system failed to outperform a naive baseline which simply output the most frequent words in the training set for each test set utterance.

Other research groups (Berard et al., 2016; Duong et al., 2016) experimented with using

neural models for ST, but initial results were inconclusive about their ability to make predictions on real speech data. The first successful application of a neural model for ST was demonstrated by Weiss et al. (2017). They trained a neural sequence-to-sequence model on around 150 hours of Spanish telephone speech data paired with English text translations. Their ST model achieved a high BLEU score, coming close to an oracle text-to-text translation system, and produced good quality translations on the held out evaluation sets. Taking inspiration from this work, we tested a similar model on various low-resource settings (Chapter 4) and showed that translation performance rapidly drops as we lower the amount of training data with around 20 hours being an inflection point below which the models fail to outperform naive baselines on our corpus.

There are many ways to try and improve translation performance. To stay true to our motivating scenarios where the source audio language is low-resource and therefore labeling additional data can be difficult, we try leveraging labeled speech data from different languages instead. We show that transfer-learning using ASR data from high-resource languages is a simple yet extremely effective method for improving translation quality (Chapter 5).

Finally, we discuss potential use cases for ST systems trained under low-resource settings (Chapter 6). We build proof-of-concept systems for two downstream applications: topic prediction for speech and cross-lingual keyword spotting, and show that ST systems trained on as little as 10 hours of Spanish-English ST data can still outperform simple baselines for these tasks, leaving the door open for future work.

Our contributions include an open-source software library for training ST systems[1] and our methods and analyses are particularly suitable for scenarios with limited training data. A limitation of our work is that we have not empirically demonstrated the effectiveness of our methods on truly low-resource languages or unwritten languages, relying instead on simulated low-resource conditions using a Spanish-English corpus.[2]

[1] github.com/0xSameer/ast

[2] We did conduct experiments on the truly low-resource language of Mboshi and demonstrated the effectiveness of our pre-training method on a Mboshi-French ST task. However, the dataset we used was extremely small and the BLEU scores achieved were very low and therefore we are cautious in our claims.

7.1 Future work

Interest in building end-to-end ST systems has been growing over the last few years and the field has evolved from having no baseline systems in 2015 to now multiple research groups building ST systems and releasing open-source software and speech datasets from different languages.[3] In this section we briefly discuss avenues for further exploration based on lessons learned from our study.

Improving the training time. The neural model used in this work comprises of a deep LSTM based encoder and decoder. With a time complexity of $O(N)$ the LSTM layers are slow to train, especially for speech input which consists of long sequences of speech frames. For example, our 20 hours Spanish-English ST model consists of around 16K training set utterances and takes about 24 hours to converge. Our most effective method for ST involves pre-training the model on hundred of hours of ASR data which further increases the overall training time. Such long training times can lead to suboptimal hyperparameter search as we are often faced with limited computational resources and cannot afford to train several models over multiple days. These long lead times also impact the pace at which we can carry out research as we need to wait to see if a particular method helps improve performance. A common technique to reduce training time during preliminary investigations is to first test on a smaller subset of data. However, this does not work well in our low-resource settings where the training data is already very limited.

Our suggestion for future research is to explore the highly parallelizable *Transformer* model (Vaswani et al., 2017) for building ST. As with encoder-decoder with attention models, initial work using Transformer has been on text data with the original paper demonstrating its effectiveness on MT. Recently, the model has also been used for ASR (Dong et al., 2018; Zhou et al., 2018) and an open-source software implementation of these models is already available.[4] In addition to faster training times, the Transformer model has also been shown to outperform their LSTM based precursors in high-resource settings, but whether they can achieve or surpass performance of our current models in low-resource settings remains open.

[3] By end-to-end ST systems we mean that source language text is not used.

[4] `github.com/kaituoxu/Speech-Transformer`

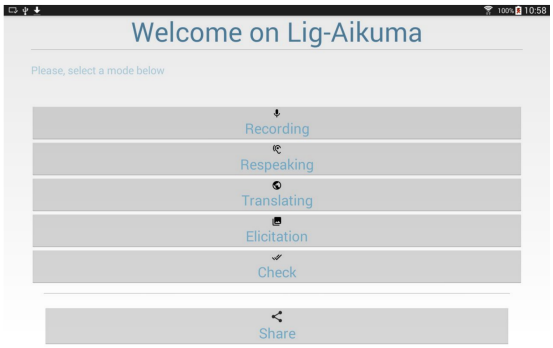


Figure 7.1: Lig-Aikuma: user interface showing the set of features provided by the data collection app. (Image source: `lig-aikuma.imag.fr`).

Aiding preservation efforts for endangered languages In Chapter 5 we built an ST system for the truly low-resource language Mboshi, translating it to French text. To train this system we had access to only around 4 hours of training data (Godard et al., 2018) which was collected using the Lig-Aikuma app (Gauthier et al., 2016) shown in Figure 7.1.[5] The Lig-Aikuma app, along with its predecessor Aikuma (Bird et al., 2014), were developed especially to aid language preservation efforts and allow field researchers to record audio and collect translations using an app on a smartphone. By fitting all the required functionality in a single portable device they overcome the typical challenges for data collection in remote locations where recording equipment may not be available and power supply can be unreliable. The apps are available for free and have a low footprint and therefore can be installed on relatively cheaper devices. The users can also replay the audio instantly to check if the recording quality is acceptable. And there is an option to preload a set of elicitation commands which the researchers would like to record. As the collected data is already in a digital format, transferring it for research use is also straightforward.

While the data collection features are very useful, the apps currently do not provide any guidance on how much data to collect to ensure satisfactory performance on downstream tasks (such as ST). Discovering the downstream task performance early on while in the field itself can help researchers better plan to collect more data if required. For example, we tried our best models and methods to train an ST system on the 4 hours of Mboshi-French training data but the translation performance remains very poor with a BLEU score of around 7 and the trained systems barely outperform naive baselines. This leaves the question open as to whether we really need more labeled data for Mboshi-French

[5] Webpage for the Lig-Aikuma app: `lig-aikuma.imag.fr`.

to train useful ST models. Our proposal would be to study whether we can integrate ST systems directly into similar apps, train it on the data collected so far, and generate evaluation metrics on a test set. The data collection team can plot improvements in the translation quality as more data is collected and can decide when to stop. In addition to ST, which might not be the desired end goal, we can also incorporate our topic prediction and keyword spotting systems as well. The challenge here would be porting computationally expensive neural models onto a device with limited resources. For a detailed discussion on developing computational tools for endangered languages, we refer readers to the work of Anastasopoulos (2019).

Using unaligned speech and text translations for ST. There has been recent promising work which explores building text-to-text translation systems using unaligned (monolingual) data from the source and target languages (Lample et al., 2018; Artetxe et al., 2018). These methods are completely unsupervised as they do not require any cross-lingual information, such as sentence level alignment or a bilingual word dictionary, and still achieve competitive translation performance compared to supervised systems. A crucial step to build this system is to train language models or word embeddings (Mikolov et al., 2013; Bojanowski et al., 2017) for the source and target languages, and then align the embedding spaces using an unsupervised method. Previous methods relied on supervision in the form of a bilingual word dictionary to align the embedding spaces. After alignment, the translation for a word can be obtained using a nearest neighbor search. Adapting this method for speech input is challenging as it is not obvious how we can train a language model for speech. Text data can be segmented into character/subword/word level tokens over which we can train a language model, but speech input is continuous and detecting word boundaries in conversational speech is an open research problem. Chung et al. (2019) recently adapted these unsupervised translation methods for speech input but they have relied on source language text to learn word boundaries in the speech data (Chung and Glass, 2018). Extending this work towards a truly unsupervised setting will be an interesting direction for future work.

Bibliography

Abdel-Hamid, O., Deng, L., and Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Proc. Interspeech*.

Adda, G., Stüker, S., Adda-Decker, M., Ambouroue, O., Besacier, L., Blachon, D., Bonneau-Maynard, H., Godard, P., Hamlaoui, F., Idiatov, D., et al. (2016). Breaking the Unwritten Language Barrier: The BULB project. *Procedia Computer Science*, 81:8–14.

Alumäe, T., Tsakalidis, S., and Schwartz, R. M. (2016). Improved multilingual training of stacked neural network acoustic models for low resource languages. In *Proc. Interspeech*.

Anastasopoulos, A. (2019). *Computational Tools for Endangered Language Documentation*. PhD thesis, University of Notre Dame.

Anastasopoulos, A., Bansal, S., Chiang, D., Goldwater, S., and Lopez, A. (2017). Spoken term discovery for language documentation using translations. In *Proc. EMNLP Workshop SCNLP*.

Artetxe, M., Labaka, G., and Agirre, E. (2018). Unsupervised statistical machine translation. In *Proc. EMNLP*.

Badino, L., Mereta, A., and Rosasco, L. (2015). Discovering discrete subword units with binarized autoencoders and Hidden-Markov-Model encoders. In *Proc. Interspeech*.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

Bansal, S., Kamper, H., Livescu, K., Lopez, A., and Goldwater, S. (2018). Low-resource speech-to-text translation. In *Proc. Interspeech*.

Bansal, S., Kamper, H., Livescu, K., Lopez, A., and Goldwater, S. (2019). Pre-training

on high-resource speech recognition improves low-resource speech-to-text translation. In *Proc. NAACL*.

Bansal, S., Kamper, H., Lopez, A., and Goldwater, S. (2017). Towards speech-to-text translation without speech recognition. In *Proc. EACL*.

Bar-Hillel, Y. (1960). A demonstration of the nonfeasibility of fully automatic high quality machine translation. *Advances in Computers*, 1:158–163.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proc. ICML*.

Berard, A., Pietquin, O., Servan, C., and Besacier, L. (2016). Listen and translate: A proof of concept for end-to-end speech-to-text translation. In *Proc. NeurIPS Workshop on End-to-end Learning for Speech and Audio Processing*.

Besacier, L., Zhou, B., and Gao, Y. (2006). Towards speech translation of non written languages. In *Proc. SLT*.

Bhargava, M. and Rose, R. (2015). Architectures for deep neural network based acoustic models defined over windowed speech waveforms. In *Proc. Interspeech*.

Bird, S., Gawne, L., Gelbart, K., and McAlister, I. (2014). Collecting bilingual audio in remote indigenous communities. In *Proc. COLING*.

Blachon, D., Gauthier, E., Besacier, L., Kouarata, G.-N., Adda-Decker, M., and Rialland, A. (2016). Parallel speech collection for under-resourced language studies using the Lig-Aikuma mobile device app. *Procedia Computer Science*, 81:61–66.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Trans. ACL*.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

Chan, R. H. Y. and Woodland, P. C. (2004). Improving broadcast news transcription by lightly supervised discriminative training. In *Proc. ICASSP*.

Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*.

Chan, W. and Lane, I. (2015). Deep convolutional neural networks for acoustic modeling in low resource languages. In *Proc. ICASSP*.

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Schuster, M., Shazeer, N., Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Chen, Z., Wu, Y., and Hughes, M. (2018). The best of both worlds: Combining recent advances in neural machine translation. In *Proc. ACL*.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proc. SSST-8, Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Chung, Y.-A. and Glass, J. R. (2018). Speech2Vec: A sequence-to-sequence framework for learning word embeddings from speech. In *Proc. Interspeech*.

Chung, Y.-A., Weng, W.-H., Tong, S., and Glass, J. (2019). Towards unsupervised speech-to-text translation. In *Proc. ICASSP*.

Church, K. W. and Hovy, E. H. (1993). Good applications for crummy machine translation. *Machine Translation*, 8(4):239–258.

Cui, J., Kingsbury, B., Ramabhadran, B., Sethy, A., Audhkhasi, K., Cui, X., Kislal, E., Mangu, L., Nussbaum-Thom, M., Picheny, M., et al. (2015). Multilingual representations for low resource speech recognition and keyword search. In *Proc. ASRU*.

Currey, A., Miceli Barone, A. V., and Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In *WMT at Proc. EMNLP*.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoustics, Speech, Signal Process.*, 28(4):357–366.

Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y., and Acero, A. (2013). Recent advances in deep learning for speech research at Microsoft. In *Proc. ICASSP*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*.

Dong, L., Xu, S., and Xu, B. (2018). Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *Proc. ICASSP*.

Dredze, M., Jansen, A., Coppersmith, G., and Church, K. (2010). NLP on spoken documents without ASR. In *Proc. EMNLP*.

Dunbar, E., Algayres, R., Karadayi, J., Bernard, M., Benjumea, J., Cao, X.-N., Miskic, L., Dugrain, C., Ondel, L., Black, A. W., et al. (2019). The Zero Resource Speech Challenge 2019: TTS without T. In *Proc. Interspeech*.

Dunbar, E., Cao, X. N., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., and Dupoux, E. (2017). The Zero Resource Speech Challenge 2017. In *Proc. ASRU*.

Duong, L., Anastasopoulos, A., Chiang, D., Bird, S., and Cohn, T. (2016). An attentional model for speech translation without transcription. In *Proc. NAACL HLT*.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*.

Ethnologue (2019a). How many languages are endangered? `https://www.ethnologue.com/guides/how-many-languages-endangered`. [Online; accessed 27-November-2019].

Ethnologue (2019b). How many languages are there in the world? `https://www.ethnologue.com/guides/how-many-languages`. [Online; accessed 27-November-2019].

Ethnologue (2019c). How many languages in the world are unwritten? `https://www.ethnologue.com/enterprise-faq/how-many-languages-world-are-unwritten-0`. [Online; accessed 27-November-2019].

Fainberg, J., Klejch, O., Renals, S., and Bell, P. (2019). Lattice-based lightly-supervised acoustic model training. In *Proc. Interspeech*.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proc. EMNLP*.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Garcia, A. and Gish, H. (2006). Keyword spotting of arbitrary words using minimal speech resources. In *Proc. ICASSP*.

Gauthier, E., Blachon, D., Besacier, L., Kouarata, G.-N., Adda-Decker, M., Rialland,

A., Adda, G., and Bachman, G. (2016). Lig-aikuma: A mobile app to collect parallel speech for under-resourced language studies. In *Proc. Interspeech*.

Ghoshal, A., Swietojanski, P., and Renals, S. (2013). Multilingual training of deep neural networks. In *Proc. ICASSP*.

Godard, P., Adda, G., Adda-Decker, M., Benjumea, J., Besacier, L., Cooper-Leavitt, J., Kouarata, G., Lamel, L., Maynard, H., M"uller, M., Rialland, A., St"uker, S., Yvon, F., and Boito, M. Z. (2018). A very low resource language speech corpus for computational language documentation experiments. In *Proc. LREC*.

Golik, P., Tüske, Z., Schlüter, R., and Ney, H. (2015). Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In *Proc. Interspeech*.

Graff, D., Huang, S., Cartagena, I., Walker, K., and Cieri, C. (2010). Fisher Spanish Speech (LDC2010S01).

Heigold, G., Vanhoucke, V., Senior, A., Nguyen, P., Ranzato, M., Devin, M., and Dean, J. (2013). Multilingual acoustic models using distributed deep neural networks. In *Proc. ICASSP*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proc. ACL*.

Huang, J.-T., Li, J., Yu, D., Deng, L., and Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proc. ICASSP*.

Hui Bu, Jiayu Du, X. N. B. W. H. Z. (2017). Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *Oriental COCOSDA 2017*.

Jansen, A. and Van Durme, B. (2011). Efficient spoken term discovery using randomized algorithms. In *Proc. ASRU*.

Jia, Y., Johnson, M., Macherey, W., Weiss, R. J., Cao, Y., Chiu, C.-C., Ari, N., Laurenzo, S., and Wu, Y. (2019). Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *Proc. ICASSP*.

Kamper, H., Elsner, M., Jansen, A., and Goldwater, S. (2015). Unsupervised neural network based feature extraction using weak top-down constraints. In *Proc. ICASSP*.

Kamper, H., Jansen, A., and Goldwater, S. (2016). Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 24(4):669–679.

Kay, M. (1980/1997). The proper place of men and machines in language translation. *Machine Translation*, pages 3–23.

Kemp, T. and Waibel, A. H. (1999). Unsupervised training of a speech recognizer: recent experiments. In *Proc. Eurospeech*.

Lamel, L., Gauvain, J.-L., and Adda, G. (2002). Lightly supervised and unsupervised acoustic model training. *Computer Speech & Language*, 16:115–129.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proc. EMNLP*.

Laurent, A., Fraga-Silva, T., Lamel, L., and Gauvain, J.-L. (2016). Investigating techniques for low resource conversational speech recognition. In *Proc. ICASSP*.

Lee, C.-y. and Glass, J. (2012). A nonparametric Bayesian approach to acoustic model discovery. In *Proc. ACL*.

Lee, C.-y., O'Donnell, T., and Glass, J. (2015). Unsupervised lexicon discovery from acoustic input. *Trans. ACL*, 3:389–403.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*.

Levin, K., Jansen, A., and Van Durme, B. (2015). Segmental acoustic indexing for zero resource keyword search. In *Proc. ICASSP*.

Menon, R., Kamper, H., Quinn, J., and Niesler, T. (2018). Fast ASR-free and almost zero-resource keyword spotting using DTW and CNNs for humanitarian monitoring. In *Proc. Interspeech*.

Metze, F., Anguera, X., Barnard, E., Davel, M., and Gravier, G. (2013). The spoken web search task at MediaEval 2012. In *Proc. ICASSP*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*.

Mohamed, A.-r. (2014). *Deep neural network acoustic models for ASR*. PhD thesis, University of Toronto.

Munro, R. (2010). Crowdsourced translation for emergency response in Haiti: The global collaboration of local knowledge. In *AMTA Workshop on Collaborative Crowdsourcing for Translation*.

Niesler, T. (2007). Language-dependent state clustering for multilingual acoustic modelling. *Speech Communication*, 49(6):453–463.

Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). CrisisLex: A lexicon for collecting and filtering microblogged communications in crises. In *Proc. ICWSM*.

Palaz, D., Magimai-Doss, M., and Collobert, R. (2015). Analysis of CNN-based speech recognition system using raw speech as input. In *Proc. Interspeech*.

Palaz, D., Synnaeve, G., and Collobert, R. (2016). Jointly learning to locate and classify words using convolutional networks. In *Proc. Interspeech*.

Park, A. S. and Glass, J. R. (2008). Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):186–197.

Post, M., Kumar, G., Lopez, A., Karakos, D., Callison-Burch, C., and Khudanpur, S. (2014). Fisher and CALLHOME Spanish–English Speech Translation (LDC2014T23).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *Proc. ASRU*.

Quinn, J. and Hidalgo-Sanchis, P. (2017). Using machine learning to analyse radio content in Uganda: Opportunities for sustainable development and humanitarian action. Technical report, United Nations Global Pulse Lab Kampala.

Ramachandran, P., Liu, P. J., and Le, Q. V. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proc. EMNLP*.

Räsänen, O. J., Doyle, G., and Frank, M. C. (2015). Unsupervised word discovery from speech using automatic segmentation into syllable-like units. In *Proc. Interspeech*.

Renshaw, D. (2016). Representation learning for unsupervised speech processing. Master's thesis, University of Edinburgh.

Renshaw, D., Kamper, H., Jansen, A., and Goldwater, S. J. (2015). A comparison of neural network methods for unsupervised representation learning on the Zero Resource Speech Challenge. In *Proc. Interspeech*.

Saeb, A., Menon, R., Cameron, H., Kibira, W., Quinn, J., and Niesler, T. (2017). Very low resource radio browsing for agile developmental and humanitarian monitoring. In *Proc. Interspeech*.

Sainath, T. N., rahman Mohamed, A., Kingsbury, B., and Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In *Proc. ICASSP*.

Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., and Vinyals, O. (2015). Learning the speech front-end with raw waveform CLDNNs. In *Proc. Interspeech*.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Salesky, E., Sperber, M., and Black, A. W. (2019a). Exploring phoneme-level speech representations for end-to-end speech translation. In *Proc. ACL*.

Salesky, E., Sperber, M., and Waibel, A. (2019b). Fluent translations from disfluent speech in end-to-end speech translation. In *Proc. NAACL*.

Schultz, T. (2002). Globalphone: a multilingual speech and text database developed at karlsruhe university. In *Seventh International Conference on Spoken Language Processing*.

Schultz, T. and Waibel, A. (2001). Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1):31 – 51.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proc. ACL*.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proc. ACL*.

Sercu, T. and Goel, V. (2016). Advances in very deep convolutional neural networks for LVCSR. In *Proc. Interspeech*.

Sercu, T., Puhrsch, C., Kingsbury, B., and LeCun, Y. (2016). Very deep multilingual convolutional neural networks for LVCSR. In *Proc. ICASSP*.

Sheridan, P., Wechsler, M., and Schäuble, P. (1997). Cross-language speech retrieval: Establishing a baseline performance. In *Proc. SIGIR*.

Siu, M.-H., Gish, H., Chan, A., Belfield, W., and Lowe, S. (2014). Unsupervised training of an HMM-based self-organizing unit recognizer with applications to topic classification and keyword discovery. *Computer Speech & Language*, 28(1):210–223.

Sperber, M., Neubig, G., Niehues, J., and Waibel, A. (2019). Attention-passing models for robust and data-efficient end-to-end speech translation. In *Trans. ACL*.

Stahlberg, F., Schlippe, T., Vogel, S., and Schultz, T. (2014). Towards automatic speech recognition without pronunciation dictionary, transcribed speech and text resources in the target language using cross-lingual word-to-phoneme alignment. In *Proc. SLTU*.

Stoian, M. C., Bansal, S., and Goldwater, S. (2019). Analyzing asr pretraining for low-resource speech-to-text translation. *arXiv preprint arXiv:1910.10762*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proc. NeurIPS*.

Swietojanski, P., Ghoshal, A., and Renals, S. (2012). Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR. In *Proc. SLT*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proc. CVPR*.

Thiollière, R., Dunbar, E., Synnaeve, G., Versteegh, M., and Dupoux, E. (2015). A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In *Proc. Interspeech*.

Thomas, S., Ganapathy, S., and Hermansky, H. (2012). Multilingual mlp features for low-resource LVCSR systems. In *Proc. ICASSP*.

Thrun, S. (1995). Is learning the n-th thing any easier than learning the first? In *Proc. NeurIPS*.

Tian, Y. (2019). How does pre-training improve low-resource speech-to-text translation? — a case study on a Swahili-English dataset. Master's thesis, University of Edinburgh.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proc. NeurIPS*.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). Sequence to sequence – video to text. In *Proc. ICCV*.

Versteegh, M., Miró, X. A., Jansen, A., and Dupoux, E. (2016). The Zero Resource Speech Challenge 2015: Proposed approaches and results. In *Proc. SLTU*.

Versteegh, M., Thiollière, R., Schatz, T., Cao, X. N., Anguera, X., Jansen, A., and Dupoux, E. (2015). The Zero Resource Speech Challenge 2015. In *Proc. Interspeech*.

Vu, N. T., Breiter, W., Metze, F., and Schultz, T. (2012). An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance. In *Proc. Interspeech*.

Walter, O., Korthals, T., Haeb-Umbach, R., and Raj, B. (2013). A hierarchical system for word discovery exploiting DTW-based initialization. In *Proc. ASRU*.

Weiss, R. J., Chorowski, J., Jaitly, N., Wu, Y., and Chen, Z. (2017). Sequence-to-sequence models can directly transcribe foreign speech. In *Proc. Interspeech*.

Wilkinson, A., Zhao, T., and Black, A. W. (2016). Deriving phonetic transcriptions and discovering word segmentations for speech-to-speech translation in low-resource settings. In *Proc. Interspeech*.

Wilpon, J. G., Rabiner, L. R., Lee, C.-H., and Goldman, E. R. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38:1870–1878.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proc. ICML*.

Yuan, Y., Leung, C.-C., Xie, L., Ma, B., and Li, H. (2016). Learning neural network representations using cross-lingual bottleneck features with word-pair information. In *Proc. Interspeech*.

Zhang, Y. and Glass, J. R. (2009). Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *Proc. ASRU*.

Zhang, Y. and Glass, J. R. (2010). Towards multi-speaker unsupervised speech pattern discovery. In *Proc. ICASSP*.

Zhang, Y., Salakhutdinov, R., Chang, H.-A., and Glass, J. R. (2012). Resource configurable spoken query detection using deep Boltzmann machines. In *Proc. ICASSP*.

Zhang, Y. L., Chan, W., and Jaitly, N. (2017). Very deep convolutional networks for end-to-end speech recognition. In *Proc. ICASSP*.

Zhou, S., Dong, L., Xu, S., and Xu, B. (2018). Syllable-based sequence-to-sequence speech recognition with the transformer in Mandarin Chinese. In *Proc. Interspeech*.