Unsupervised Learning of Cross-Modal Mappings between Speech and Text

by

Yu-An Chung

B.S., National Taiwan University (2016)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Author .....
Department of Electrical Engineering and Computer Science
May 3, 2019

Certified by .....
James R. Glass
Senior Research Scientist in Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Unsupervised Learning of Cross-Modal Mappings between Speech and Text

by

Yu-An Chung

Submitted to the Department of Electrical Engineering and Computer Science
on May 3, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Deep learning is one of the most prominent machine learning techniques nowadays, being the state-of-the-art on a broad range of applications in computer vision, natural language processing, and speech and audio processing. Current deep learning models, however, rely on significant amounts of supervision for training to achieve exceptional performance. For example, commercial speech recognition systems are usually trained on tens of thousands of hours of annotated data, which take the form of audio paired with transcriptions for training acoustic models, collections of text for training language models, and (possibly) linguist-crafted lexicons mapping words to their pronunciations. The immense cost of collecting these resources makes applying state-of-the-art speech recognition algorithm to under-resourced languages infeasible.

In this thesis, we propose a general framework for mapping sequences between speech and text. Each component in this framework can be trained without any labeled data so the entire framework is unsupervised. We first propose a novel neural architecture that learns to represent a spoken word in an unlabeled speech corpus as an embedding vector in a latent space, in which word semantics and relationships between words are captured. In parallel, we train another latent space that captures similar information about written words using a corpus of unannotated text. By exploiting the geometrical properties exhibited in the speech and text embedding spaces, we develop an unsupervised learning algorithm that learns a cross-modal alignment between speech and text. As an example application of the learned alignment, we develop a unsupervised speech-to-text translation system using only unlabeled speech and text corpora.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist in Computer Science

Acknowledgments

Being admitted to MIT is definitely a dream come true to me. I really appreciate the Electrical Engineering and Computer Science department for giving me this opportunity. For me, doing research is never easy without the support from the others. There are too many people I would like to thank to for being part of this journey, witnessing my struggle as well as my growth.

First and foremost, I would like to thank my advisor, Jim, for being the best advisor imaginable. You are one of the nicest people I have ever met in my life. The thing I feel the most thankful to you is the freedom you offered me for exploring research topics whatever I found interesting for the past two years. I am super lucky and grateful to have you as my advisor.

I would also like to thank my labmates in the Spoken Language Systems Group, who are not only talented but also very easy to get along with. I always learn a lot from the discussions during our weekly reading group and meeting. Special thanks to my officemates at G442 for interesting daily chit-chat. As an international student, this is a great opportunity for me to practice my English speaking. Thank you, Hao and Di-Chia, for playing tennis with me during both weekdays and weekends. I know I became emotional during games from time to time. Thanks for enduring my disappointment and anger towards my own unforced errors. Here I just want you to know that's because I really want to play better!

I was fortunate to intern at Google in summer 2018. Thank you, Yuxuan, for being my host and leading me into the field of speech synthesis, a brand new research area to me at that time. From this internship experience, I understand how doing research at school differs from doing research in the industry. I will always remember you said to me once that your goal of being my host is to help me succeed.

To my friends from the Taiwanese Student Association, thank you for enriching my life. I have always believed that better life can lead to better work. My part of life outside of doing research is just as important as my research. To Schrasing, one of my two best friends at MIT and also my fantastic roommate. Thank you for

setting up all the facilities including Wi-Fi and furniture before I moved in to our new apartment. Thank you, Wei-Hung, for being the other of my two best friends here. Thanks for sharing your life and traveling experience with me. I also enjoy every research collaboration we have had, and am looking forward to our next project. Thank you both for hanging out with me and I hope this friendship will last forever.

I would like to thank my family, especially my grandparents, parents Chun-Wei (Wilson) and Yu-Hsin (Christine), and my older brother Yu-Hao (Edison), for being extremely caring and supportive throughout my entire life—I would not be here, finishing my Master's thesis at MIT today, if not for you. Your love and support mean so much to me, and my gratitude to you is beyond words.

To Jo-Chi (Ashley), thank you for keeping me company all these years and putting up with my emotion when things were not going well, especially during the time when I was applying for graduate schools and when I was serving in military. I cherish every memory we share by heart.

This work was supported in part by iFlytek.

Bibliographic Note

Much of the work presented in this thesis has previously appeared in peer-reviewed scientific publications. The content of Chapter 2 was largely published in Chung and Glass [2017] and Chung and Glass [2018] at NIPS 2017 Workshop on Machine Learning for Audio Signal Processing and INTERSPEECH 2018, respectively. Chapter 3 was published in Chung et al. [2018b] at NeurIPS 2018. Chapter 4 was published in Chung et al. [2019c] at ICASSP 2019.

Code and data in this thesis are available at `https://github.com/iamyuanchung`.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation of this Work

Machine learning, especially deep learning, has become the most prominent tool for fulfilling artificial intelligence. In the field of natural language and speech processing, where data is usually expressed as a sequence of smaller units (e.g., a sentence is a sequence of words, and a speech utterance is a sequence of acoustic features), many tasks can be formulated as the transformation—or transduction—of input sequences into output sequences: machine translation, speech recognition, and text-to-speech synthesis to name but a few. Due to their ability of handling variable-length sequences and capturing long-term dependency between units within a sequence, deep learning models such as the recurrent neural network and its variants the long short-term memory network [Hochreiter and Schmidhuber, 1997] and the gated recurrent unit network [Chung et al., 2014] have been largely used as a core component for modelling sequences when developing automatic sequence transducers. As the deep learning community continues to thrive, ground-breaking neural architectures and methods such as the sequence-to-sequence paradigm [Sutskever et al., 2014, Cho et al., 2014, Gehring et al., 2017], attention mechanism [Bahdanau et al., 2015, Luong et al., 2015], and the most recent Transformer model [Vaswani et al., 2017] have been proposed to further improve the state-of-the-art performance. Nowadays, machines are able to achieve performance that is close to human level on speech recognition [Chiu et al.,

2018, Saon et al., 2017, Xiong et al., 2017], machine translation [Wu et al., 2016], and speech synthesis [Shen et al., 2018, Wang et al., 2017].

However, these state-of-the-art models are built within a supervised learning framework, which often requires a significant amount of labeled data for training. For example, commercial speech recognition systems [Li et al., 2017] are often trained on tens of thousands of hours of annotated data, which take the form of audio with parallel transcriptions for training acoustic models, collections of text for training language models, and possibly linguist-crafted lexicons mapping words to their pronunciations. Recent end-to-end systems [Chiu et al., 2018] also require data to be in the form of paired audio and text for end-to-end training. The cost of accumulating such kind of data is immense, so it is no surprise that only major languages like English and Mandarin—which have plenty of annotated data readily available—are supported by high-quality speech recognition. The fact that these state-of-the-art models require significant amounts of labeled data for training poses a major obstacle for speech technology to be applied to low-resource languages, which account for most of the languages spoken around the world. Compared to annotated data, unlabeled data are relatively easy to collect (e.g., one can effortlessly crawl a massive amount of text from the Internet or record tens of thousands of hours of conversational speech). Therefore, it would be very useful (and desirable) if we can design unsupervised learning approaches that rely only on nonparallel speech and text corpora for solving sequence transduction tasks such as speech recognition and translation.

There has been some work in weakly- and semi-supervised learning for speech recognition and synthesis [Karita et al., 2018, Drexler and Glass, 2018, Subramanya and Bilmes, 2011, Yu et al., 2010, Huang and Hasegawa-Johnson, 2010, Chung et al., 2019b, Hsu et al., 2019], where plenty of unlabeled data and only a small amount of labeled data are available. In this thesis, we focus on an unsupervised setting, that is, not requiring any labeled data.

1.2 Unsupervised Speech Processing

Our work is closely related to unsupervised speech processing [Glass, 2012], a field that has attracted considerable attention in the last few years. This research field deals with the setting when unlabeled speech data are the only available resource for a language, and unsupervised learning methods are required to learn representations and linguistic structure directly and only from the speech signal. One of the major research streams in unsupervised speech processing is unsupervised representation learning [Chung et al., 2019a, Chorowski et al., 2019, Oord et al., 2018, Pascual et al., 2019, Chung and Glass, 2018, Chen et al., 2018, Hsu et al., 2017a,b, Zeghidour et al., 2016, Renshaw et al., 2015, Chen et al., 2015, Lee and Glass, 2012, Zhang and Glass, 2010, Varadarajan et al., 2008], where the task is to find speech features that make it easier to discriminate between meaningful linguistic units (e.g., phones or words). Unsupervised speech segmentation is another major areas of unsupervised speech processing, where given an unlabeled speech corpus, the goal is to find repeated word- or phrase-like patterns [Park and Glass, 2008, Jansen and Van Durme, 2011, Lyzinski et al., 2015], or, in a more difficult scenario, to predict word boundaries and lexical categories for the entire set [Lee et al., 2015, Räsänen et al., 2015, Kamper et al., 2017a,b].

1.3 Contributions

In this thesis, we propose a general framework for transducing sequences between the speech and text modalities. Each component in the framework can be trained without any labeled data so the entire framework is unsupervised.

To start with, in Chapter 2, we design a novel neural architecture that learns to represent any spoken word in an unlabeled speech corpus as an embedding vector in a latent space, in which word semantics and relationships between words are captured. In parallel, we train another latent space that captures similar information of written words using an unannotated text corpus. The two corpora, which are used to train

the embedding spaces of their respective modalities (speech and text), do not need to be parallel and can be collected independently. Chapter 3, the core of this thesis, exploits the similarity of geometrical structures of the speech and text embedding spaces and learn a *cross-modal* alignment between them. We then show how we can utilize this cross-modal alignment to develop a speech-to-text sequence transduction system in a completely unsupervised manner in Chapter 4. Specifically, a speech-to-text translation system is presented as the example application. Finally, we conclude this thesis and discuss future work in Chapter 5.

Chapter 2

Representing Words as Fixed-Dimensional Vectors

This chapter introduces Speech2Vec, a novel neural architecture for learning fixed-dimensional vector representations of speech segments corresponding to spoken words excised from a speech corpus. These vectors contain semantic information pertaining to the underlying spoken words, whose relationships are also encoded inside these vectors.

We start with providing some background knowledge about word embeddings[1] in Section 2.1. We then formally introduce Speech2Vec in Section 2.2. Experiments are presented in Section 2.3, which includes evaluation of the learned word embeddings, observations and discussions on the results, and visualization of the word embeddings. Parts of this chapter was published in Chung and Glass [2017, 2018].

[1]In this thesis, the term "word embeddings" will be used interchangeably with terms "word vectors" and "vector representations". All of them refer to dense representations of words in a low-dimensional vector space.

2.1 Background

2.1.1 Word Embeddings

To make machines understand and process natural language, we need to transform words coming in free text into numeric values. One of the simplest transformation approaches is one-hot encoding in which each distinct word stands for one dimension of the resulting vector and a binary (0 and 1) value indicates whether the word is present or not. However, one-hot encoding is computationally impractical when dealing with the entire vocabulary set, as the representation demands hundreds of thousands of dimensions. It is therefore desirable to have word embedding approaches capable of representing words and phrases in vectors of (non-binary) numeric values with much lower and thus denser dimensions.

The history of word embeddings can be traced back to the 1990s, when vector space models were used in distributional semantics and models for estimating continuous representations of words such as Latent Semantic Analysis [Landauer et al., 1998] and Latent Dirichlet Allocation [Blei et al., 2003] were proposed. The term "word embeddings" was coined in Bengio et al. [2003], which proposed a simple feed-forward neural network to perform language modeling and produces word embeddings as a by-product. Collobert and Weston [2008], Collobert et al. [2011], however, were probably the first to show the utility of pre-trained word embeddings. They showcased that word embeddings trained on a sufficiently large dataset carry syntactic and semantic information and improve performance on downstream tasks.

Word2Vec [Mikolov et al., 2013b] and GloVe [Pennington et al., 2014] are two of the most successful and prevalent word embedding models nowadays. They obtain word embeddings via unsupervised learning from co-occurrence information in text, producing word embeddings that encode general semantic relationships. A well-known example showcasing such relationship is $w2v(king) - w2v(man) + w2v(woman) \approx w2v(queen)$, where $w2v(\cdot)$ is a learned Word2Vec embedding function. Additionally, it is worth mentioning that the main benefit of these word embeddings arguably is that they don't require any annotation, but can be derived from large unannotated corpora

that are readily available. Pre-trained embeddings can then be used in downstream tasks that only have small amounts of labeled data. Some successful applications of word embeddings are dependency parsing [Yu and Vu, 2017, Ballesteros et al., 2015], named entity recognition [Lample et al., 2016], part-of-speech tagging [Plank et al., 2016], language modeling [Kim et al., 2016], just to name a few.

Word embedding approaches such as Word2Vec and GloVe, however, still have some drawbacks. First of all, they have trouble handling the so-called "polysemy" phenomenon, where a word has completely different meanings depending on the context it appears (for example, consider the word "bank" in "the bank was robbed" and "we had a picnic on the river bank"). This problem is inevitable for these approaches because they are always trying to use a single vector to represent each word during training. Secondly, their optimization objectives are usually based on very shallow language modeling tasks, so there is a limitation to what the learned word embeddings can capture. These disadvantages have motivated the recent development of deep language models (language models that use architectures like deep long short-term memory networks [Hochreiter and Schmidhuber, 1997] and Transformer [Vaswani et al., 2017]) for modeling "contextualized" word representations. Instead of always mapping the same word to the same vector regardless of the context, a contextualized word embedding is a function of the entire sentence, allowing the same word to be represented as different vectors that capture different semantics depending on its context. It is therefore no surprise that these contextualized word embeddings approaches achieve state-of-the-art performance on a wide range of natural language processing tasks [Peters et al., 2018, Howard and Ruder, 2018, Devlin et al., 2019, Radford et al., 2018].

2.1.2 Acoustic Word Embeddings

Researchers have also explored the concept of learning vector representations from audio data [Kamper, 2019, Holzenberger et al., 2018, Milde and Biemann, 2018, Wang et al., 2018, He et al., 2017, Settle and Livescu, 2016, Chung et al., 2016, Kamper et al., 2016b, Bengio and Heigold, 2014, Levin et al., 2013]. However, these approaches

are based on notions of acoustic-phonetic (rather than *semantic*) similarity, so that different instances of the same underlying word would map to the same point in a latent embedding space.

Another stream of research by Harwath and Glass [2017], Harwath et al. [2016], Harwath and Glass [2015] has presented a deep neural network model capable of rudimentary spoken language acquisition using raw speech training data paired with contextually relevant images. Using this contextual grounding, the model learns a latent semantic audio-visual embedding space. Other similar work that learns a joint embedding space between the language and vision modalities include Kamper et al. [2019, 2017c], Chrupała et al. [2017], Ephrat et al. [2018], which have shown interesting results in applications such as speech retrieval and speech separation. Our goal here, however, is to derive a model capable of learning word embeddings from *raw* speech without any forms of supervision from any other modalities.

2.2 Speech2Vec: Learning Word Embeddings from Speech

Given the observation that humans learn to speak before they can read or write, one might wonder that since machines can learn semantics from raw text, might they also be able to learn the semantics of a spoken language from raw speech as well?

Our goal is to learn a fixed-length embedding of a speech segment corresponding to a spoken word that is represented by a variable-length sequence of acoustic features such as Mel-Frequency Cepstral Coefficients (MFCCs), $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, where $\mathbf{x}_t$ is the acoustic feature at time t and T is the length of the sequence. We desire that this word embedding is able to describe the semantics of the original spoken word to some degree. Below we first review a deep neural network architecture commonly referred to as the RNN Encoder-Decoder framework, which is the backbone of our Speech2Vec model, followed by formally proposing it.

2.2.1 Model Architecture: RNN Encoder-Decoder

A Recurrent Neural Network (RNN) Encoder-Decoder consists of an Encoder RNN and a Decoder RNN [Sutskever et al., 2014, Cho et al., 2014]. For an input sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, the Encoder reads each of its symbol $\mathbf{x}_i$ sequentially, and the hidden state $\mathbf{h}_t$ of the RNN is updated accordingly. After the last symbol $\mathbf{x}_T$ is processed, the corresponding hidden state $\mathbf{h}_T$ is interpreted as the learned representation of the entire input sequence. Subsequently, by initializing its hidden state using $\mathbf{h}_T$, the Decoder generates an output sequence $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{T'})$ sequentially, where T and T' can be different. Such a sequence-to-sequence framework does not constrain the input or target sequences, and has been successfully applied to tasks such as speech recognition [Chiu et al., 2018], machine translation [Bahdanau et al., 2015], video caption generation [Venugopalan et al., 2015], abstract meaning representation parsing and generation [Konstas et al., 2017], and acoustic word embeddings acquisition [Chung et al., 2016].

With the RNN Encoder-Decoder as the backbone architecture, Speech2Vec, inspired by Word2Vec, uses two methodologies for training Speech2Vec: skipgrams and continuous bag-of-words (CBOW). The two methodologies are based on the distributional hypothesis, whose basic idea is that words that are used and occur in the same contexts tend to purport similar meanings.

2.2.2 Speech2Vec based on Skipgrams

The idea of training Speech2Vec with skipgrams is that for each speech segment $\mathbf{x}^{(n)} = (\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}, ..., \mathbf{x}_T^{(n)})$ (corresponding to the sequence representing the n-th word) in a speech corpus, the model is trained to predict the speech segments $\{\mathbf{x}^{(n-k)}, ..., \mathbf{x}^{(n-1)}, \mathbf{x}^{(n+1)}, ..., \mathbf{x}^{(n+k)}\}$ (corresponding to nearby words) within a certain range k before and after the sequence $\mathbf{x}^{(n)}$, where k is referred to as window size. During training, the Encoder first takes $\mathbf{x}^{(n)}$ as input and encodes it into a vector representation of fixed dimensionality $\mathbf{z}^{(n)}$. The Decoder then maps $\mathbf{z}^{(n)}$ to several output sequences $\mathbf{y}^{(i)}, i \in \{n-k, ..., n-1, n+1, ..., n+k\}$. The model is trained by

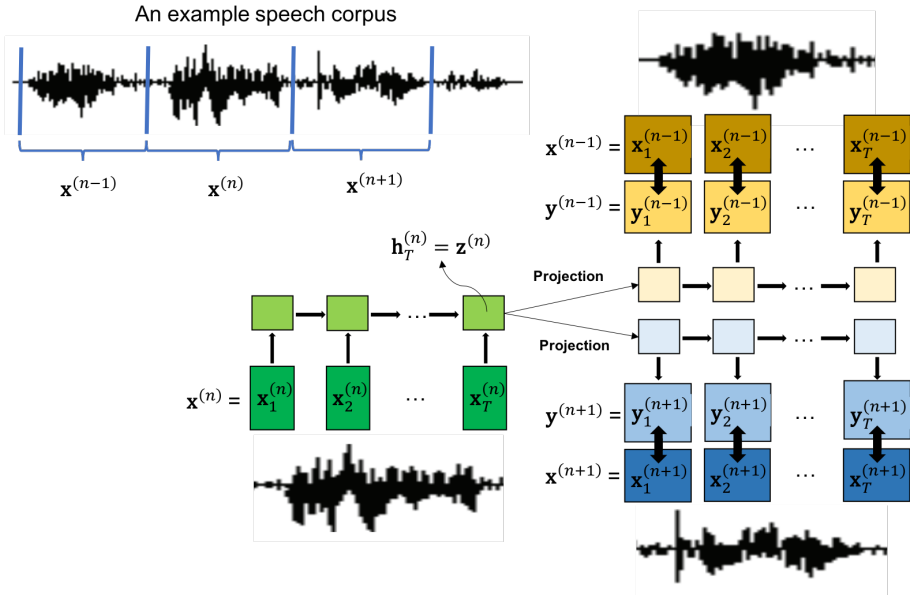

Figure 2-1: The illustration of Speech2Vec trained with skipgrams. All speech segments, with each corresponding to a spoken word and represented as a sequence of acoustic features, were padded by zero vectors into the same length T. During training, the model is given a speech segment and aims to predict its nearby speech segments within a certain window size k ($k = 1$ in this figure). Note that it is the same Decoder RNN that generates all the output speech segments.

minimizing the gap between the output sequences and their corresponding nearby speech segments, measured by the general mean squared error $\sum_i \left\| \mathbf{x}^{(i)} - \mathbf{y}^{(i)} \right\|^2$, $i \in \{n-k, ..., n-1, n+1, ..., n+k\}$. The intuition behind this approach is that, in order to successfully decode nearby speech segments, the encoded vector representation $\mathbf{z}^{(n)}$ should contain sufficient semantic information about the current speech segment $\mathbf{x}^{(n)}$. After training, $\mathbf{z}^{(n)}$ is taken as the word embedding of $\mathbf{x}^{(n)}$. Note that it is the same Decoder RNN that generates all the output speech segments, and all speech segments can have different lengths.

2.2.3 Speech2Vec based on CBOW

In contrast to training Speech2Vec with skipgrams that aims to predict nearby speech segments from $\mathbf{z}^{(n)}$, training Speech2Vec with CBOW sets $\mathbf{x}^{(n)}$ as the target and aims to infer it from nearby speech segments. During training, all nearby speech segments are encoded by a shared Encoder into $\mathbf{h}^{(i)}, i \in \{n-k, ..., n-1, n+1, ..., n+k\}$,

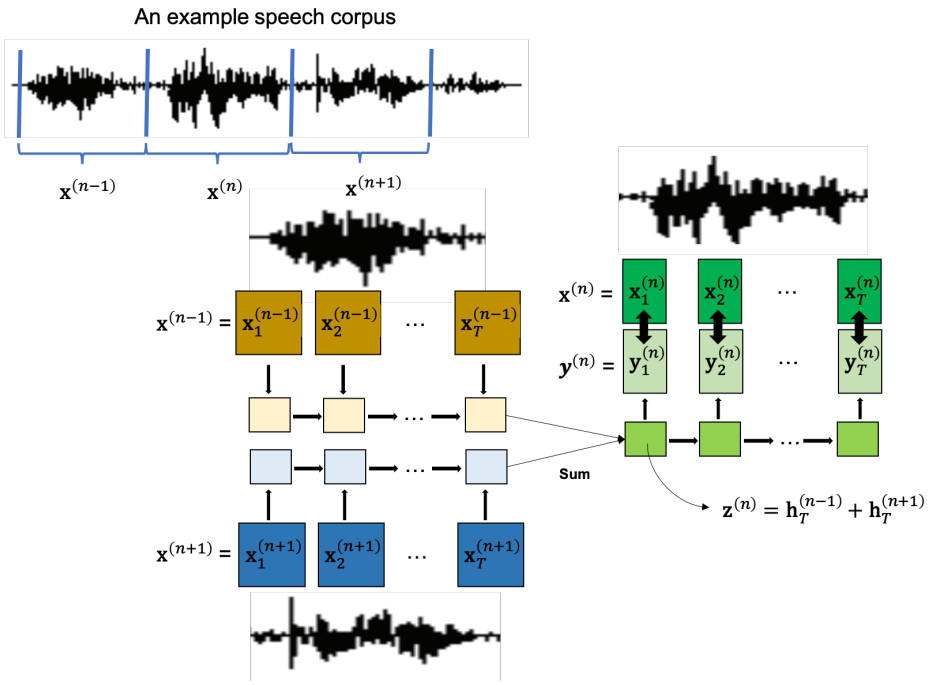

Figure 2-2: The illustration of Speech2Vec trained with CBOW. During training, the model aims to generate the target speech segment given its nearby speech segments within a window size k ($k = 1$ in this figure). Note that all input speech segments share the same Encoder RNN.

and their sum $\mathbf{z}^{(n)} = \sum_i \mathbf{h}^{(i)}$ is then used by the Decoder to generate $\mathbf{x}^{(n)}$. After training, $\mathbf{z}^{(n)}$ is taken as the word embedding for $\mathbf{x}^{(n)}$. In our experiments, we found that Speech2Vec trained with skipgrams consistently outperforms that trained with CBOW.

2.2.4 Differences between Speech2Vec and Word2Vec

Speech2Vec aims to learn a fixed-length embedding of a speech segment that captures the semantic information of the spoken word directly from speech data. It can be viewed as a speech version of Word2Vec. Although they have many properties in common, such as sharing the same training methodologies (skipgrams and CBOW), and learning word embeddings that capture semantic information from their respective modalities, it is important to identify two fundamental differences. First, the architecture of a Word2Vec model is a two-layered fully-connected neural network with one-hot encoded vectors as input and output. In contrast, the Speech2Vec model is

composed of Encoder and Decoder RNNs, in order to handle variable-length input and output sequences of acoustic features. Second, in a Word2Vec model, the embedding for a particular word is deterministic. Every instance of the same word will be represented by one, and only one, embedding vector. In contrast, in the Speech2Vec model, due to the fact that every instance of a spoken word will be different (due to speaker, channel, and other contextual differences etc.), every instance of the same underlying word will be represented by a *different* (though hopefully similar) embedding vector. For experimental purposes, in Section 2.3, all vectors representing instances of the same spoken word are averaged to obtain a single word embedding. The effect of this averaging operation is also discussed.

2.3 Experiments

2.3.1 Data and Preprocessing

For our experiments we used LibriSpeech [Panayotov et al., 2015], a corpus of read English speech, to learn Speech2Vec embeddings. In particular, we used a 500 hour subset of broadband speech produced by 1,252 speakers. Speech features consisting of 13 dimensional Mel Frequency Cepstral Coefficients (MFCCs) were produced every 10ms. The speech was pre-segmented according to word boundaries obtained by forced alignment with respect to the reference transcriptions such that each speech segment corresponds to a spoken word. This resulted in a large set of speech segments $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(|C|)}\}$, where $|C|$ denotes the total number of speech segments (words) in the corpus.

2.3.2 Model Implementation

We implemented the Speech2Vec model with PyTorch [Paszke et al., 2017]. The Encoder RNN is a single-layered bidirectional LSTM [Hochreiter and Schmidhuber, 1997], and the Decoder RNN is another single-layered unidirectional LSTM. To facilitate the learning process, we also adopted the attention mechanism similar to Subra-

manian et al. [2018] that allows the Decoder to condition every decoding step on the last hidden state of the Encoder, in other words, the Decoder can refer to $\mathbf{h}_T$ when generating every symbol $\mathbf{y}_t$ of the output sequence $\mathbf{y}$. The window size k for training the model with skipgrams and CBOW is set to three. The model was trained by stochastic gradient descent (SGD) with a fixed learning rate of $1e-3$ and 500 epochs. We experimented with hyperparameter combinations for training the Speech2Vec model, including the depths of the Encoder and Decoder RNNs, which memory cell (LSTM or GRU [Chung et al., 2014]) to use, and bidirectional or unidirectional RNNs. We conducted experiments using the specified architecture since it produced the most stable and satisfactory results.

2.3.3 Evaluation Setup

Existing schemes for evaluating methods for word embeddings fall into two major categories: extrinsic and intrinsic [Schnabel et al., 2015]. With the extrinsic method, the learned word embeddings are used as input features to a downstream task [Yu and Vu, 2017, Lample et al., 2016, Plank et al., 2016, Kim et al., 2016, Ballesteros et al., 2015], and the performance metric varies from task to task. The intrinsic method directly tests for semantic or syntactic relationships between words, and includes the tasks of word similarity and word analogy [Mikolov et al., 2013b]. In this work, we focus on the intrinsic method, especially the word similarity task, for evaluating and analyzing the Speech2Vec word embeddings.

We used 13 benchmarks [Faruqui and Dyer, 2014a] to measure word similarity, including **WS-353**, **WS-353-REL**, **WS-353-SIM**, **MC-30**, **RG-65**, **Rare-Word**, **MEN**, **MTurk-287**, **MTurk-771**, **YP-130**, **SimLex-999**, **Verb-143**, and **SimVerb-3500**. These 13 benchmarks contain different numbers of pairs of English words that have been assigned similarity ratings by humans, and each of them evaluates the word embeddings in terms of different aspects. For example, **RG-65** and **MC-30** focus on nouns, **YP-130** and **SimVerb-3500** focus on verbs, and **Rare-Word** focuses on rare-words. The similarity between a given pair of words was calculated by computing the cosine similarity between their corresponding word

embeddings. We then reported the Spearman's rank correlation coefficient ρ between the rankings produced by each model against the human rankings [Myers and Well, 1995]. Word embeddings that achieve higher ρ are considered better in terms of capturing word semantics. For more details about these word similarity benchmarks, please refer to Appendix A.

We compared Speech2Vec trained with skipgrams or CBOW with its Word2Vec counterpart trained on the transcriptions of the LibriSpeech corpus using the fastText implementation [Bojanowski et al., 2017]. Note that people usually train Word2Vec on a much larger text corpus such as Google News or Wikipedia. Here we trained Word2Vec and Speech2Vec on comparable sets of corpora from the same collection so as to given them a fair comparison. For convenience, we refer to these four models as skipgrams Speech2Vec, CBOW Speech2Vec, skipgrams Word2Vec, and CBOW Word2Vec, respectively.

2.3.4 Results and Discussions

We trained the four models with different embedding sizes to understand how large the embedding size should be to capture sufficient semantic information about the word. The results are shown in Table 2.1. We also varied the size of the corpus used for training the four models and report the results in Table 2.2. The numbers in both tables are the average of running the experiment 10 times and the standard deviations are negligible. From Table 2.1 and Table 2.2, we have the following discussions.

Embedding size impact on performance. We found that increasing the embedding size does not always result in improved performance. For CBOW Speech2Vec, skipgrams Speech2Vec, and CBOW Word2Vec, word embeddings of 50-dimensions are able to capture enough semantic information of the words, as the best performance (highest ρ) of each benchmark is mostly achieved by them. For skipgrams Word2Vec, although the best performance of 7 out of 13 benchmarks is achieved by word embeddings of 200-dims, there are 6 benchmarks whose best performance is achieved by word embeddings of other sizes. That being said, we believe that

Table 2.1: The relationship between the embedding size and the performance on 13 word similarity benchmarks. The results of Speech2Vec and Word2Vec are displayed in Table 2.1a and Table 2.1b, respectively.

(a) Speech2Vec trained with CBOW and skipgrams on the LibriSpeech speech data.

Model	Speech2Vec							
	CBOW				skipgrams			
Vector dim.	10	50	100	200	10	50	100	200
Verb-143	0.182	0.223	0.203	0.205	0.263	0.315	0.276	0.222
SimLex-999	0.183	0.235	0.238	0.237	0.200	0.292	0.317	0.335
MC-30	0.680	0.716	0.688	0.684	0.701	0.846	0.815	0.787
WS-353	0.305	0.343	0.336	0.335	0.370	0.508	0.502	0.498
WS-353-SIM	0.461	0.484	0.474	0.471	0.533	0.663	0.653	0.636
WS-353-REL	0.122	0.192	0.189	0.186	0.207	0.346	0.332	0.331
RG-65	0.676	0.705	0.699	0.697	0.702	0.790	0.756	0.740
MEN	0.476	0.509	0.501	0.498	0.543	0.619	0.606	0.573
MTurk-287	0.346	0.349	0.336	0.331	0.426	0.468	0.442	0.398
MTurk-771	0.356	0.391	0.380	0.377	0.445	0.521	0.503	0.463
SimVerb-3500	0.098	0.122	0.126	0.125	0.100	0.157	0.183	0.204
Rare-Word	0.240	0.273	0.275	0.269	0.249	0.323	0.321	0.317
YP-130	0.198	0.216	0.211	0.214	0.322	0.321	0.334	0.302

(b) Word2Vec trained with CBOW and skipgrams on the LibriSpeech transcriptions.

Model	Word2Vec							
	CBOW				skipgrams			
Vector dim.	10	50	100	200	10	50	100	200
Verb-143	0.296	0.380	0.383	0.385	0.307	0.378	0.384	0.365
SimLex-999	0.118	0.146	0.142	0.140	0.202	0.280	0.298	0.300
MC-30	0.524	0.539	0.532	0.521	0.726	0.762	0.746	0.713
WS-353	0.198	0.234	0.228	0.233	0.334	0.452	0.455	0.471
WS-353-SIM	0.313	0.335	0.330	0.334	0.491	0.602	0.599	0.605
WS-353-REL	0.051	0.106	0.095	0.100	0.172	0.308	0.308	0.327
RG-65	0.421	0.425	0.424	0.428	0.666	0.752	0.749	0.724
MEN	0.427	0.465	0.461	0.459	0.563	0.642	0.646	0.632
MTurk-287	0.368	0.387	0.390	0.389	0.430	0.504	0.503	0.469
MTurk-771	0.246	0.290	0.289	0.288	0.413	0.499	0.504	0.479
SimVerb-3500	0.049	0.075	0.072	0.069	0.090	0.149	0.176	0.193
Rare-Word	0.230	0.307	0.309	0.310	0.286	0.408	0.419	0.431
YP-130	0.231	0.261	0.257	0.253	0.345	0.391	0.431	0.448

Table 2.2: The relationship between the size of the training corpus and the performance on 13 word similarity benchmarks. The results of Speech2Vec and Word2Vec are displayed in Table 2.2a and Table 2.2b, respectively. The percentage denotes the proportion of the entire corpus that was used for training the models. The reported results are based on the word embeddings of 50-dim.

(a) Speech2Vec trained with CBOW and skipgrams on the LibriSpeech speech data.

Model	Speech2Vec							
	CBOW				skipgrams			
Training size	10%	40%	70%	100%	10%	40%	70%	100%
Verb-143	0.090	0.071	0.116	0.223	0.098	0.152	0.220	0.315
SimLex-999	0.073	0.181	0.205	0.235	0.171	0.272	0.286	0.292
MC-30	0.366	0.503	0.667	0.716	0.469	0.702	0.761	0.846
WS-353	-0.101	0.211	0.319	0.343	0.066	0.392	0.459	0.508
WS-353-SIM	0.001	0.376	0.494	0.484	0.117	0.489	0.609	0.663
WS-353-REL	-0.120	0.081	0.174	0.192	-0.084	0.258	0.304	0.346
RG-65	0.024	0.199	0.593	0.705	0.020	0.605	0.661	0.790
MEN	0.033	0.311	0.451	0.509	0.283	0.506	0.585	0.619
MTurk-287	0.059	0.156	0.236	0.349	0.133	0.312	0.399	0.468
MTurk-771	0.098	0.246	0.321	0.391	0.186	0.416	0.462	0.521
SimVerb-3500	-0.023	0.060	0.096	0.122	0.042	0.119	0.145	0.157
Rare-Word	0.071	0.200	0.249	0.273	0.210	0.329	0.308	0.323
YP-130	-0.027	0.067	0.181	0.216	0.097	0.196	0.311	0.321

(b) Word2Vec trained with CBOW and skipgrams on the LibriSpeech transcriptions.

Model	Word2Vec							
	CBOW				skipgrams			
Training size	10%	40%	70%	100%	10%	40%	70%	100%
Verb-143	0.196	0.257	0.331	0.380	0.148	0.259	0.328	0.378
SimLex-999	0.014	0.091	0.096	0.146	0.114	0.249	0.266	0.280
MC-30	0.487	0.367	0.456	0.532	0.657	0.625	0.662	0.762
WS-353	0.045	0.091	0.167	0.234	0.129	0.377	0.412	0.452
WS-353-SIM	0.083	0.190	0.303	0.335	0.181	0.463	0.559	0.602
WS-353-REL	-0.016	-0.046	0.003	0.106	0.013	0.237	0.256	0.308
RG-65	0.196	0.192	0.333	0.425	0.330	0.416	0.642	0.752
MEN	0.016	0.258	0.403	0.465	0.247	0.541	0.621	0.642
MTurk-287	0.101	0.357	0.367	0.387	0.286	0.440	0.494	0.504
MTurk-771	0.094	0.148	0.223	0.290	0.182	0.392	0.474	0.499
SimVerb-3500	-0.028	0.008	0.045	0.075	0.019	0.116	0.144	0.149
Rare-Word	0.151	0.261	0.275	0.307	0.324	0.408	0.415	0.408
YP-130	0.064	0.085	0.182	0.256	0.403	0.216	0.365	0.391

Speech2Vec would benefit from increasing the embedding sizes when a larger speech corpus is available.

Comparing Speech2Vec to Word2Vec. From Table 2.1 we see that skipgrams Speech2Vec achieves the highest ρ in 8 out of 13 benchmarks, outperforming CBOW and skipgrams Word2Vec in combination. We believe a possible reason for such results is due to skipgrams Speech2Vec's ability to capture semantic information present in speech such as prosody that is not in text.

Comparing skipgrams to CBOW Speech2Vec. From Table 2.1 we observe that skipgrams Speech2Vec consistently outperforms CBOW Speech2Vec on all benchmarks for all embedding sizes. This result aligns with the empirical fact that skipgrams Word2Vec is likely to work better than CBOW Word2Vec with small training corpus size [Mikolov et al., 2013b].

Impact of training corpus size. From Table 2.2 we observe that when 10% of the corpus was used for training, the resulting word embeddings perform poorly. Unsurprisingly, the performance continues to improve as training size increases.

2.3.5 Variance Study on Speech2Vec Embeddings

At the end of Section 2.2, we mention that in Speech2Vec, every instance of a spoken word will produce a different embedding vector. Here we try to understand how the vectors for a given word vary, i.e., are they similar, or is there considerable variance that the averaging operation we adopted smooths out?

To study this, we partitioned all words into four sub-groups based on the number of times, N, that they appeared in the corpus, ranging from $5 \sim 99, 100 \sim 999, 1000 \sim 9999,$ and $\geq 10k$. Then, for all vector representations $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^N\}$ of a given word w that appeared N times, we computed the mean of the standard deviations of each dimensions $m_w = \frac{1}{d}\sum_{i=1}^{d} \texttt{std}(\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^N)$, where d denotes the embedding size. Finally, we averaged m_w for every word w that belongs to the same sub-group

and reported the results in Figure 2-3.

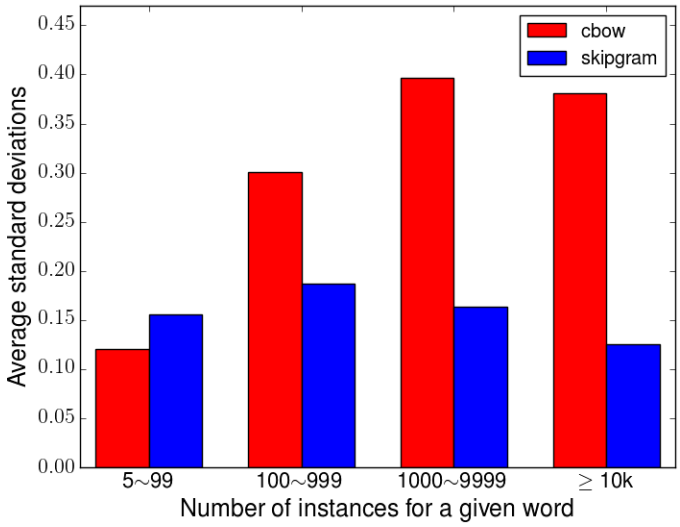

Figure 2-3: How the vector representations for a given word vary with respect to the times it appears in the corpus.

From Figure 2-3 we observe that when N falls in $5 \sim 99$, the variances of the vectors generated by CBOW Speech2Vec are smaller than those generated by skipgrams Speech2Vec. However, when N becomes bigger, variances of the vectors generated by skipgrams Speech2Vec become smaller than those generated by CBOW Speech2Vec, and the gap continues to grow as N increases. We suspect the lower variation of the skipgrams model relative to the CBOW model is related to the overall superior performance of the skipgrams Speech2Vec model. We are encouraged that the deviation of the skipgrams model gets smaller as N increases, as it suggests stability in the model. We conclude that the vectors produced by skipgrams Speech2Vec for a given word are relatively invariant with respect to the frequency of the word and the averaging operation does not have a large impact in an either positive or negative way. While for CBOW Speech2Vec, the averaging operation is unable to smooth out the variance as CBOW Speech2Vec consistently performs worse than skipgrams Speech2Vec according to Table 2.1, and thus calls for better methods for mapping several vectors into a single one.

2.3.6 Visualizing Speech2Vec Embeddings

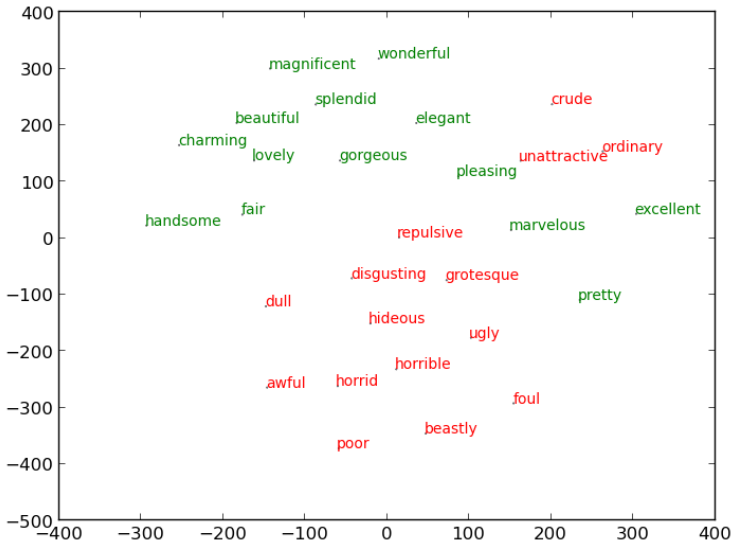

Figure 2-4: t-SNE projection of the word embeddings learned by skipgrams Speech2Vec. Words with positive and negative meanings were colored in green and red, respectively.

We visualized the word embeddings learned by skipgrams Speech2Vec with t-SNE [Maaten and Hinton, 2008] in Figure 2-4. We see that words with positive meanings (colored in green) are mainly located at the upper part of the figure, while words with negative meanings (colored in red) are mostly located at the bottom. Such distribution suggests that the learned word embeddings do capture notions of antonym and synonyms to some degree.

2.4 Conclusions

In this chapter, we propose Speech2Vec, a neural architecture that integrates the RNN Encoder-Decoder framework with skipgrams or CBOW for training and extends the text-based Word2Vec [Mikolov et al., 2013b] model to learn word embeddings directly from speech. Speech2Vec has access to richer information in the speech signal that does not exist in plain text, which is one of the possible reasons why in our experiments in Section 2.3, the learned word embeddings outperform those produced by Word2Vec from the transcriptions.

We are fully aware of the fact that using word similarity tasks as the only way to measure the quality of word vectors is imperfect and can sometimes lead to incorrect inferences [Faruqui et al., 2016, Schnabel et al., 2015]. In this chapter, we used these word similarity benchmarks for faster validation of the effectiveness of the proposed model for learning meaningful vector representations from speech. The usefulness of the Speech2Vec embeddings in downstream tasks, which are what we truly care about, will be investigated more in the rest of the thesis.

Chapter 3

Aligning Speech and Text Embeddings without Parallel Data

Recent research has shown that word embedding spaces learned from text corpora of different languages can be aligned without any parallel data supervision. Inspired by the success in unsupervised cross-lingual word embeddings, in this chapter we target learning a *cross-modal* alignment between the embedding spaces of speech and text learned from corpora of their respective modalities in an unsupervised fashion. We propose a framework that first respectively learns the individual speech and text embedding spaces using Speech2Vec and Word2Vec [Mikolov et al., 2013b], and then attempts to align the two spaces via adversarial training, followed by a refinement procedure. We show how our framework could be used to perform spoken word recognition and translation, and the experimental results on these two tasks demonstrate that the performance of our unsupervised alignment approach is comparable to its supervised counterpart. Our framework is especially useful for developing speech-to-text sequence transduction systems such as automatic speech recognition (ASR) and speech-to-text translation for low- or zero-resource languages, which have little parallel audio-text data for training modern supervised ASR and speech-to-text translation models, but account for the majority of the languages spoken across the world.

This chapter is organized as follows. We start with a brief introduction to cross-lingual word embeddings and our motivation in Section 3.1. Section 3.2 describes how

we obtain the speech embedding space in a completely unsupervised manner using Speech2Vec. Next, we present our unsupervised cross-modal alignment approach in Section 3.3. In Section 3.4, we describe the tasks of spoken word recognition and translation, which are similar to ASR and speech-to-text translation, respectively, except that now the input are speech segments corresponding to spoken words. Finally, we evaluate the performance of our unsupervised alignment on the two tasks and analyze our results in Section 3.5. The content of this chapter was published in Chung et al. [2018b].

3.1 Introduction

3.1.1 Cross-Lingual Word Embeddings

Most successful word embedding models [Mikolov et al., 2013b, Pennington et al., 2014, Bojanowski et al., 2017] rely on the distributional hypothesis [Harris, 1954], i.e., words occurring in similar contexts tend to have similar meanings. Exploiting word co-occurrence statistics in a text corpus leads to word vectors that reflect semantic similarities and dissimilarities: similar words are geometrically close in the embedding space, and conversely, dissimilar words are far apart.

In addition, word embedding spaces have been shown to exhibit similar structures across languages [Mikolov et al., 2013a]. The intuition is that most languages share similar expressive power and are used to describe similar human experiences across cultures; hence, they should share similar statistical properties. Inspired by the notion, several studies have focused on designing algorithms that exploit this similarity to learn a cross-lingual alignment between the embedding spaces of two languages, where the two embedding spaces are trained from independent text corpora [Faruqui and Dyer, 2014b, Xing et al., 2015, Artetxe et al., 2016, Smith et al., 2016, Artetxe et al., 2017, Cao et al., 2016, Duong et al., 2016]. In particular, recent research has shown that such cross-lingual alignments can be learned without relying on any form of bilingual supervision [Zhang et al., 2017a,b, Conneau et al., 2018, Artetxe

et al., 2018a], and has been applied to training machine translation systems in a completely unsupervised fashion [Lample et al., 2018b, Artetxe et al., 2018b, Lample et al., 2018a]. This eliminates the need for a large parallel training corpus to train machine translation systems.

3.1.2 Motivation

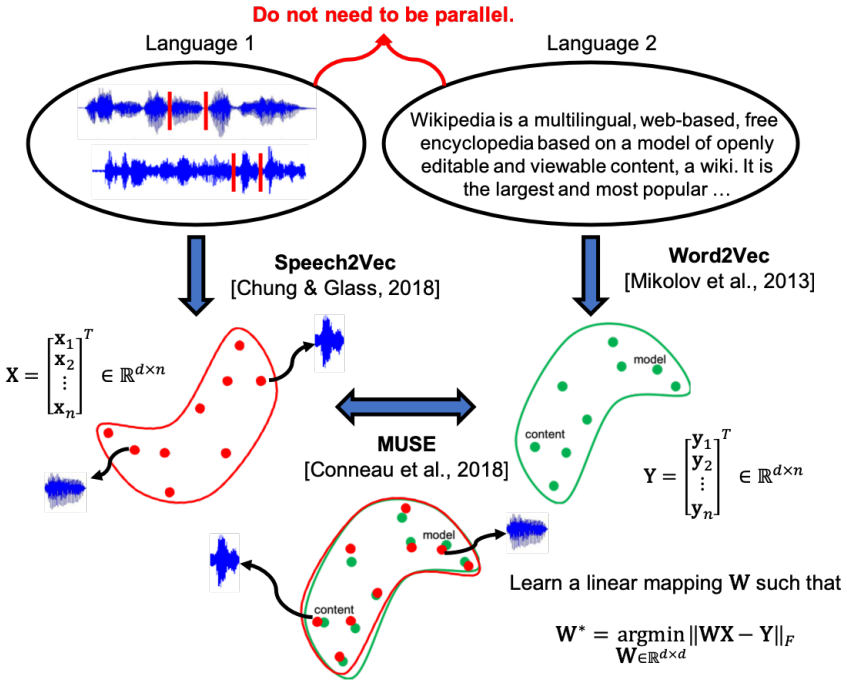

Figure 3-1: Overview of the proposed framework. Given two independent corpora of speech and text that do not need to be parallel, the framework individually learns speech and text embeddings using Speech2Vec and Word2Vec. Next, it leverages an algorithm that is originally proposed for unsupervised cross-lingual word embeddings to learn a *cross-modal* linear mapping from the speech embedding space to the text embedding space. The entire framework is unsupervised.

In Chapter 2, we develop Speech2Vec, which is capable of representing speech segments excised from a speech corpus as fixed dimensional vectors that contain semantic information of the underlying spoken words. The design of Speech2Vec is based on the RNN Encoder-Decoder framework [Sutskever et al., 2014, Cho et al., 2014], and borrows the methodology of skipgrams or continuous bag-of-words from Word2Vec for training. Since Speech2Vec and Word2Vec share the same training methodology

and speech and text are similar media for communicating, it is reasonable to assume that the two embedding spaces learned respectively by Speech2Vec from speech and Word2Vec from text exhibit similar structure.

Motivated by the recent success in unsupervised cross-lingual alignment [Zhang et al., 2017a,b, Conneau et al., 2018, Artetxe et al., 2018a] and the assumption that the embedding spaces of the two modalities (speech and text) share similar structure, we are interested in learning an unsupervised *cross-modal* alignment between the two spaces. Such an alignment would be useful for developing automatic speech recognition (ASR) and speech-to-text translation systems for low- or zero-resource languages that lack parallel corpora of speech and text for training. In this chapter, we propose a framework for unsupervised cross-modal alignment, borrowing the methodology from unsupervised cross-lingual alignment presented in Conneau et al. [2018]. The framework consists of two steps. First, it uses Speech2Vec and Word2Vec to learn the individual embedding spaces of speech and text. Next, it leverages adversarial training to learn a linear mapping from the speech embedding space to the text embedding space, followed by a refinement procedure. The proposed framework is illustrated in Figure 3-1.

3.2 Unsupervised Learning of the Speech Embedding Space

Both Speech2Vec and Word2Vec [Mikolov et al., 2013b] learn the semantics of words by making use of the co-occurrence information in their respective modalities, and are both intrinsically unsupervised. However, unlike text where the content can be easily segmented into word-like units, speech has a continuous form by nature, making the word boundaries challenging to locate. In Chapter 2, we assumed that utterances in the speech corpus are already pre-segmented into speech segments corresponding to words using word boundaries obtained by forced alignment. Such an assumption, however, makes the process of learning word embeddings from speech not truly un-

supervised and hence defeats our goal. To eliminate the need of forced alignment, here we propose a simple pipeline for training Speech2Vec in a totally unsupervised manner.

3.2.1 Unsupervised Speech Segmentation

Unsupervised speech segmentation is a core problem in zero-resource speech processing in the absence of transcriptions, lexicons, or language modeling text. Early work mainly focused on unsupervised term discovery, where the aim is to find word- or phrase-like patterns in a collection of speech [Park and Glass, 2008, Jansen and Van Durme, 2011]. While useful, the discovered patterns are typically isolated segments spread out over the data, leaving much speech as background. This has prompted several studies on *full-coverage* approaches, where the entire speech input is segmented into word-like units [Kamper et al., 2016a, Lee et al., 2015, Sun and Van hamme, 2013, Walter et al., 2013].

3.2.2 Unsupervised Speech2Vec

We propose to use an off-the-shelf, full-coverage, unsupervised segmentation system for segmenting our data into word-like units. Three representative systems are explored in this paper. The first one, referred to as Bayesian embedded segmental Gaussian mixture model (BES-GMM) [Kamper et al., 2017a], is a probabilistic model that represents potential word segments as fixed-dimensional acoustic word embeddings [Levin et al., 2013], and builds a whole-word acoustic model in this embedding space while jointly doing segmentation. The second one, called embedded segmental K-means model (ES-KMeans) [Kamper et al., 2017b], is an approximation to BES-GMM that uses hard clustering and segmentation, rather than full Bayesian inference. The third one is the recurring syllable-unit segmenter called SylSeg [Räsänen et al., 2015], a fast and heuristic method that applies unsupervised syllable segmentation and clustering, to predict recurring syllable sequences as words.

After training the Speech2Vec model using the speech segments obtained by an

unsupervised segmentation method, each speech segment is then transformed into an embedding that contains the semantic information about the segment. Since we do not know the identity of the embeddings, we use the k-means algorithm to cluster them into K clusters, potentially corresponding to K different word types. We then average all embeddings that belong to the same cluster (potentially the instances of the same underlying word) to obtain a single embedding. Note that by doing so, it is possible that we group the embeddings corresponding to different words that are semantically similar into one cluster.

3.3 The Embedding Spaces Alignment Framework

Suppose we have speech and text embedding spaces trained on independent speech and text corpora. Our goal is to learn a mapping, without using any form of cross-modal supervision, between them such that the two spaces are aligned.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_m\} \subseteq \mathbb{R}^{d_1}$ and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\} \subseteq \mathbb{R}^{d_2}$ be two sets of m and n word embeddings of dimensionality d_1 and d_2 from the speech and text embedding spaces, respectively. Ideally, if we have a known dictionary that specifies which $s_i \in \mathcal{S}$ corresponds to which $t_j \in \mathcal{T}$, we can learn a linear mapping W between the two embedding spaces such that

$$W^* = \underset{W \in \mathbb{R}^{d_2 \times d_1}}{\operatorname{argmin}} \|WX - Y\|^2, \tag{3.1}$$

where X and Y are two aligned matrices of size $d_1 \times k$ and $d_2 \times k$ formed by k word embeddings selected from $\mathcal{S}$ and $\mathcal{T}$, respectively. At test time, the transformation result of any speech segment a in the speech domain can be defined as $\operatorname{argmax}_{t_j \in \mathcal{T}} \cos(Ws_a, t_j)$. Our goal here is to learn this mapping W without using any cross-modal supervision. The proposed framework, inspired by Conneau et al. [2018], consists of two steps: domain-adversarial training for learning an initial proxy of W, followed by a refinement procedure which uses the words that match the best to create a synthetic parallel dictionary for applying Equation 3.1.

3.3.1 Domain-Adversarial Training

The intuition behind this step is to make the mapped $\mathcal{S}$ and $\mathcal{T}$ indistinguishable. We define a discriminator, whose goal is to discriminate between elements randomly sampled from $W\mathcal{S} = \{Ws_1, Ws_2, \ldots, Ws_m\}$ and $\mathcal{T}$. The mapping W, which can be viewed as the generator, is trained to prevent the discriminator from making accurate predictions. This is a two-player game, where the discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making $W\mathcal{S}$ and $\mathcal{T}$ as *similar* as possible. Given the mapping W, the discriminator, parameterized by θ_D, is optimized by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_D(\theta_D|W) = &-\frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}(\text{speech} = 1|Ws_i) \\ &-\frac{1}{n}\sum_{j=1}^{n}\log P_{\theta_D}(\text{speech} = 0|t_j),\end{aligned} \tag{3.2}$$

where $P_{\theta_D}(\text{speech} = 1|v)$ is the probability that vector v originates from the speech embedding space (as opposed to an embedding from the text embedding space). Given the discriminator, the mapping W aims to fool the discriminator's ability to accurately predict the original domain of the embeddings by minimizing the following objective function:

$$\begin{aligned}\mathcal{L}_W(W|\theta_D) = &-\frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}(\text{speech} = 0|Ws_i) \\ &-\frac{1}{n}\sum_{j=1}^{n}\log P_{\theta_D}(\text{speech} = 1|t_j)\end{aligned} \tag{3.3}$$

The discriminator θ_D and the mapping W are optimized iteratively to respectively minimize $\mathcal{L}_D$ and $\mathcal{L}_W$ following the standard training procedure of adversarial networks [Goodfellow et al., 2014].

3.3.2 Refinement Procedure

The domain-adversarial training step learns a rotation matrix W that aligns the speech and text embedding spaces. To further improve the alignment, we use the W learned in the domain-adversarial training step as an initial proxy and build a synthetic parallel dictionary that specifies which $s_i \in \mathcal{S}$ corresponds to which $t_j \in \mathcal{T}$.

To ensure a high-quality dictionary, we consider the most frequent words from $\mathcal{S}$ and $\mathcal{T}$, since more frequent words are expected to have better quality of embedding vectors, and only retain their mutual nearest neighbors. For deciding mutual nearest neighbors, we use the Cross-Domain Similarity Local Scaling proposed in Conneau et al. [2018] to mitigate the so-called hubness problem [Dinu et al., 2015] (points tending to be nearest neighbors of many points in high-dimensional spaces). Subsequently, we apply Equation 3.1 on this generated dictionary to refine W.

3.4 Defining Tasks for Evaluating the Alignment Quality

Conventional hybrid ASR systems [Graves et al., 2013] and recent end-to-end ASR models [Graves and Jaitly, 2014, Chorowski et al., 2015, Chan et al., 2016, Amodei et al., 2016, Chiu et al., 2018] rely on a large amount of parallel audio-text data for training. However, most languages spoken across the world lack parallel data, so it is no surprise that only very few languages support ASR. It is the same story for speech-to-text translation [Waibel and Fugen, 2008], which typically pipelines ASR and machine translation, and could be even more challenging to develop as it requires both components to be well trained. Compared to parallel audio-text data, the cost of accumulating independent corpora of speech and text is significantly lower. With our unsupervised cross-modal alignment approach, it becomes feasible to build ASR and speech-to-text translation systems using independent corpora of speech and text only, a setting suitable for low- or zero-resource languages.

Since a cross-modal alignment is learned to link the *word* embedding spaces of

speech and text, we perform the tasks of spoken word recognition and translation to directly evaluate the effectiveness of the alignment. The two tasks are similar to standard ASR and speech-to-text translation, respectively, except that now the input is a speech segment corresponding to a spoken word.

3.4.1 Spoken Word Recognition

The goal of this task is to recognize the underlying spoken word of an input speech segment. Suppose we have two independent corpora of speech and text that belong to the same language. The speech and text embedding spaces, denoted by $\mathcal{S}$ and $\mathcal{T}$, can be obtained by training Speech2Vec and Word2Vec on their respective corpora. The alignment W between $\mathcal{S}$ and $\mathcal{T}$ can be learned in an either supervised or unsupervised way. At test time, given an input speech segment, it is first transformed into an embedding vector s in the speech embedding space $\mathcal{S}$ by Speech2Vec. The vector s is then mapped to the text embedding space as $t_s = Ws \in \mathcal{T}$. In $\mathcal{T}$, the word that has embedding vector $t^* = \text{argmax}_{t \in \mathcal{T}} \cos(t, t_s)$ closest to t_s will be taken as the recognition result. The performance is measured by accuracy.

3.4.2 Spoken Word Translation

This task is similar to the one in the text domain that considers the problem of retrieving the translation of given source words, except that the source words are in the form of speech segments. Spoken word translation can be performed in the exact same way as spoken word recognition, but the speech and text corpora belong to different languages. At test time, we follow the standard practice of word translation and measure how many times one of the correct translations (in text) of the input speech segment is retrieved, and report precision@ k for $k = 1$ and 5. We use the bilingual dictionaries provided by Conneau et al. [2018] to obtain the correct translations of a given source word.

3.5 Experiments

In this section, we empirically demonstrate the effectiveness of our unsupervised cross-modal alignment approach on spoken word recognition and translation introduced in Section 3.4.

3.5.1 Data and Preprocessing

For our experiments, we used English and French LibriSpeech [Panayotov et al., 2015, Kocabiyikoglu et al., 2018], and English and German Spoken Wikipedia Corpora (SWC) [Köhn et al., 2016]. All corpora are read speech, and come with a collection of utterances and the corresponding transcriptions. For convenience, we denote the speech and text data of a corpus in uppercase and lowercase, respectively. For example, EN_{swc} and en_{swc} represent the speech and text data, respectively, of English SWC.

Table 3.1: Detailed statistics of the corpora.

Corpus	Train	Test	Words	Segments
English LibriSpeech	420 hr	50 hr	37K	468K
French LibriSpeech	200 hr	30 hr	26K	260K
English SWC	355 hr	40 hr	25K	284K
German SWC	346 hr	40 hr	31K	223K

In Table 3.1, column Train is the size of the speech data used for training the speech embeddings; column Test is the size of the speech data used for testing, where the corresponding number of speech segments (i.e., spoken word tokens) is specified in column Segments; column Words provides the number of distinct words in that corpus. Train and test sets are split in a way so that there are no overlapping speakers.

3.5.2 Model Implementation and Setup

The speech embeddings were trained using Speech2Vec with skipgrams by setting the window size k to three. The Encoder is a single-layer bidirectional LSTM, and the Decoder is a single-layer unidirectional LSTM. The model was trained by SGD with a

fixed learning rate of 10^{-3}. The text embeddings were obtained by training Word2Vec on the transcriptions using the fastText implementation without subword information [Bojanowski et al., 2017]. The dimension of both speech and text embeddings is 50. During our hyperparameter search, we tried window size $k \in \{1, 2, 3, 4, 5\}$ and embedding dimension $d \in \{50, 100, 200, 300\}$ and found that the reported k and d yield the best performance.

For the adversarial training, the discriminator was a two-layer neural network of size 512 with ReLU as the activation function. Both the discriminator and W were trained by SGD with a fixed learning rate of 10^{-3}. For the refinement procedure, we used the default setting specified in Conneau et al. [2018]. We also tried multi-layer neural network to model W. However, we did not observe any improvement on our evaluation tasks when using it compared to a linear W. This discovery aligns with Mikolov et al. [2013a].

3.5.3 Comparing Methods

Table 3.2: Different configurations for training Speech2Vec to obtain the speech embeddings with decreasing level of supervision. The last column specifies whether the configuration is unsupervised.

Configuration	Speech2Vec training		Unsupervised
	How word segments were obtained	How embeddings were grouped together	
A & A^*	Forced alignment	Use word identity	✗
B	Forced alignment	k-means	✗
C	BES-GMM [Kamper et al., 2017a]	k-means	✓
D	ES-KMeans [Kamper et al., 2017b]	k-means	✓
E	SylSeg [Räsänen et al., 2015]	k-means	✓
F	Equally sized chunks	k-means	✓

Alignment-Based Approaches. Given the speech and text embeddings, alignment-based approaches learn the alignment between them in an either supervised or unsupervised way; for an input speech segment, they perform spoken word recognition and translation as described in Section 3.4.

By varying how word segments were obtained before being fed to Speech2Vec

and how the embeddings were grouped together, the level of supervision is gradually decreased towards a fully unsupervised configuration. In configuration A, the speech training data was segmented into words using forced alignment with respect to the reference transcription, and the embeddings of the same word were grouped together using their word identities. In configuration B, the word segments were also obtained by forced alignment, but the embeddings were grouped together by performing k-means clustering. In configurations C, D, and E, the speech training data was segmented into word-like units using different unsupervised segmentation algorithms described in Section 3.2. Configuration F serves as a baseline by naively segmenting the speech training data into equally sized chunks. Unlike configurations A and B, configurations C, D, E, and F did not require the reference transcriptions to do forced alignment and the embeddings were grouped together by performing k-means clustering, and are thus unsupervised. Configurations A to F all used our unsupervised alignment approach to align the speech and text embedding spaces.

We also implemented configuration A^*, which trained Speech2Vec in the same way as configuration A, but learned the alignment using a parallel dictionary as cross-modal data supervision. The different configurations are summarized in Table 3.2.

Word Classifier. We established an upper bound by using the fully-supervised Word Classifier that was trained to map speech segments directly to their corresponding word identities. The Word Classifier was composed of a single-layer bidirectional LSTM with a softmax layer appended at the output of its last time step. This approach is specific to spoken word recognition.

Majority Word Baseline. For both spoken word recognition and translation tasks, we implemented a straightforward baseline dubbed Major-Word, where for recognition, it always predicts the most frequent word, and for translation, it always predicts the most commonly paired word. Results of the Major-Word offer us insight into the word distribution of the test set.

3.5.4 Results and Discussions

Table 3.3: Accuracy on spoken word recognition. $\text{EN}_{\text{ls}} - \text{en}_{\text{swc}}$ means that the speech and text embeddings were learned from the speech training data of English LibriSpeech and text training data of English SWC, respectively, and the testing speech segments came from English LibriSpeech. The same rule applies to Table 3.5 and Table 3.6. For the Word Classifier, $\text{EN}_{\text{ls}} - \text{en}_{\text{swc}}$ and $\text{EN}_{\text{swc}} - \text{en}_{\text{ls}}$ could not be obtained since it requires parallel audio-text data for training.

Corpora	$\text{EN}_{\text{ls}} - \text{en}_{\text{ls}}$	$\text{FR}_{\text{ls}} - \text{fr}_{\text{ls}}$	$\text{EN}_{\text{swc}} - \text{en}_{\text{swc}}$	$\text{DE}_{\text{swc}} - \text{de}_{\text{swc}}$	$\text{EN}_{\text{ls}} - \text{en}_{\text{swc}}$	$\text{EN}_{\text{swc}} - \text{en}_{\text{ls}}$
Nonalignment-based approach						
Word Classifier	89.3	83.6	86.9	80.4	–	–
Alignment-based approach with cross-modal supervision (parallel dictionary)						
A^*	25.4	27.1	29.1	26.9	21.8	23.9
Alignment-based approaches without cross-modal supervision (our approach)						
A	23.7	24.9	25.3	25.8	18.3	21.6
B	19.4	20.7	22.6	21.5	15.9	17.4
C	10.9	12.6	14.4	13.1	6.9	8.0
D	11.5	12.3	14.2	12.4	7.5	8.3
E	6.5	7.2	8.9	7.4	4.5	5.9
F	0.8	1.4	2.8	1.2	0.2	0.5
Majority Word Baseline						
Major-Word	0.3	0.2	0.3	0.4	0.3	0.3

Spoken Word Recognition. Table 3.3 presents our results on spoken word recognition. We observe that the accuracy decreases as the level of supervision decreases, as expected. We also note that although the Word Classifier significantly outperforms all the other approaches under all corpora settings, the prerequisite for training such a fully-supervised approach is unrealistic—it requires the utterances to be perfectly segmented into speech segments corresponding to words with the word identity of each segment *known*. We emphasize that the Word Classifier is just used to establish an upper bound performance that gives us an idea on how good the recognition results could be.

For alignment-based approaches, configuration A^* achieves the highest accuracies under all corpora settings by using a parallel dictionary as cross-modal supervision for learning the alignment. However, we see that configuration A using our unsupervised alignment approach only suffers a slight decrease in performance, which demonstrates that our unsupervised alignment approach is almost as effective as it

supervised counterpart A^*. As we move towards unsupervised methods (k-means clustering) for grouping embeddings, in configuration B, a decrease in performance is observed.

The performance of using unsupervised segmentation algorithms is behind using exact word segments for training Speech2Vec, shown in configurations C, D, and E versus B. We hypothesize that word segmentation is a critical step, since incorrectly separated words lack a logical embedding, which in turn hinders the clustering process. The importance of proper segmentation is evident in configuration F as it performs the worst.

The aforementioned analysis applies to different corpora settings. We also observe that the performance of the embeddings learned from different corpora is inferior to the ones learned from the same corpus (refer to columns 1 and 3, versus 5 and 6, in Table 3.3). We think this is because the embedding spaces learned from the same corpora (e.g., both embeddings were learned from LibriSpeech) exhibit higher similarity than those learned from different corpora, making the alignment more accurate.

Spoken Word Synonyms Retrieval. Word recognition does not display the full potential of our alignment approach. In Table 3.4 we show a list of retrieved results of example input speech segments. The words were ranked according to the cosine similarity between their embeddings and that of the speech segment mapped from the speech embedding space.

Table 3.4: Retrieved results of example speech segments that are considered incorrect in word recognition. The match for each speech segment is marked in bold.

Rank	Input speech segments			
	beautiful	clever	destroy	suitcase
1	lovely	cunning	destroyed	bags
2	pretty	smart	**destroy**	suitcases
3	gorgeous	**clever**	annihilate	luggage
4	**beautiful**	crafty	destroying	briefcase
5	nice	wisely	destruct	**suitcase**

From the table we observe that the list actually contains both synonyms and

different lexical forms of the speech segment. This provides an explanation of why the performance of alignment-based approaches on word recognition is poor: the top ranked word may not match the underlying word of the input speech segment, and would be considered incorrect for word recognition, despite that the top ranked word has high chance of being semantically similar to the underlying word.

Table 3.5: Results on spoken word synonyms retrieval. We measure how many times one of the synonyms of the input speech segment is retrieved, and report precision@k for $k = 1, 5$.

Corpora	$\text{EN}_{\text{ls}} - \text{en}_{\text{ls}}$		$\text{FR}_{\text{ls}} - \text{fr}_{\text{ls}}$		$\text{EN}_{\text{swc}} - \text{en}_{\text{swc}}$		$\text{DE}_{\text{swc}} - \text{de}_{\text{swc}}$		$\text{EN}_{\text{ls}} - \text{en}_{\text{swc}}$		$\text{EN}_{\text{swc}} - \text{en}_{\text{ls}}$	
Average P@k	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5
Alignment-based approach with cross-modal supervision (parallel dictionary)												
A^*	52.6	66.9	46.6	69.4	47.4	62.5	49.2	63.7	41.3	54.2	39.0	49.4
Alignment-based approaches without cross-modal supervision (our approach)												
A	43.2	57.0	42.4	58.0	36.3	50.4	32.6	48.8	33.9	47.5	33.4	45.7
B	35.0	48.2	35.4	50.4	33.8	44.6	29.3	45.4	30.0	42.9	31.1	40.7
C	27.7	37.3	26.4	35.7	21.1	30.3	26.2	34.5	22.4	28.9	17.1	26.3
D	26.7	35.2	27.2	36.3	21.1	28.2	25.3	33.2	21.2	29.3	18.7	25.1
E	17.7	24.2	20.8	28.4	17.3	21.8	18.3	23.0	15.2	21.1	11.2	17.8
F	3.5	5.7	5.2	6.9	3.8	5.8	2.7	4.9	3.2	5.7	2.9	4.4

We define word synonyms retrieval to also consider synonyms as valid results, as opposed to the word recognition. The synonyms were derived using another language as a pivot. Using the cross-lingual dictionaries provided by Conneau et al. [2018], we looked up the acceptable word translations, and for each of those translations, we took the union of their translations back to the original language. For example, in English, each word has 3.3 synonyms on average. Table 3.5 shows the results of word synonyms retrieval. We see that our approach performs better at retrieving synonyms than classifying words, an evidence that the system is learning the semantics rather than the identities of words. This showcases the strength of our semantics-focused approach.

Spoken Word Translation. Table 3.6 presents the results on spoken word translation. Similar to spoken word recognition, configurations with more supervision yield better performance than those with less supervision. Furthermore, we observe that translating using the same corpus outperforms those using different corpora (refer

Table 3.6: Results on spoken word translation. We measure how many times one of the correct translations of the input speech segment is retrieved, and report precision@k for $k = 1, 5$.

Corpora	$EN_{ls} - fr_{ls}$		$FR_{ls} - en_{ls}$		$EN_{swc} - de_{swc}$		$DE_{swc} - en_{swc}$		$EN_{ls} - de_{swc}$		$FR_{ls} - de_{swc}$	
Average P@k	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5
Alignment-based approach with cross-modal supervision (parallel dictionary)												
A^*	47.9	56.4	49.1	60.1	40.2	51.9	43.3	55.8	34.9	46.3	33.8	44.9
Alignment-based approaches without cross-modal supervision (our approach)												
A	40.5	50.3	39.9	50.9	32.8	43.8	33.1	43.4	31.9	42.2	30.1	42.1
B	36.0	44.9	35.5	44.5	27.9	38.3	30.9	40.9	26.6	35.3	25.4	38.2
C	24.7	35.4	23.9	37.3	22.0	30.3	20.5	29.1	19.2	26.1	14.8	23.1
D	25.4	33.1	24.4	34.6	23.5	29.1	20.7	31.3	20.8	25.9	14.5	22.4
E	15.4	20.6	16.7	19.9	14.1	15.9	16.6	17.0	14.8	16.7	9.7	11.8
F	4.3	5.6	6.9	7.5	4.9	6.5	5.3	6.6	4.2	5.9	1.8	2.6
Majority Word Baseline												
Major-Word	1.1	1.5	1.6	2.2	1.2	1.5	2.0	2.7	1.1	1.5	1.6	2.2

to $EN_{swc} - de_{swc}$ versus $EN_{ls} - de_{swc}$). We attribute this to the higher structural similarity between the embedding spaces learned from the same corpora.

3.6 Conclusions

In this chapter, we proposed a framework capable of aligning speech and text embedding spaces in a completely unsupervised manner. The method learns the alignment from independent corpora of speech and text, without requiring any cross-modal supervision, which is especially important for low- or zero-resource languages that lack parallel data with both audio and text. We demonstrate the effectiveness of our unsupervised alignment by showing comparable results to its supervised alignment counterpart that uses full cross-modal supervision (see A vs. A^* in Tables 3.3, 3.5, and 3.6) on the tasks of spoken word recognition and translation.

In the next chapter, we describe how our cross-modal alignment framework could be used to develop real-world speech-to-text sequence transduction systems. Specifically, we take the task of speech-to-text translation as an example application and build a completely unsupervised system of it using the alignment framework as a fundamental building block.

Chapter 4

Unsupervised Speech-to-Text Translation

In Chapter 3, we proposed a completely unsupervised approach capable of learning an alignment between speech and text embedding spaces inferred from monolingual corpora of speech and text without relying on any forms of supervision. In this chapter, we apply this unsupervised cross-modal alignment approach in a real-world downstream task as an example application. Specifically, we present a framework for building speech-to-text translation (ST) systems using only monolingual speech and text corpora, in other words, speech utterances from a source language and independent text from a target language. As opposed to traditional cascaded systems and end-to-end architectures, our system does not require any labeled data (i.e., transcribed source audio or parallel source and target text corpora) during training, making it especially applicable to language pairs with very few or even zero bilingual resources. The framework initializes the ST system with a cross-modal bilingual dictionary inferred from the monolingual corpora, that maps every source speech segment corresponding to a spoken word to its target text translation. For unseen source speech utterances, the system first performs word-by-word translation on each speech segment in the utterance. The translation is further improved by leveraging a language model and a sequence denoising autoencoder to provide prior knowledge about the target language. Experimental results show that our unsupervised system

achieves comparable BLEU scores to supervised end-to-end models despite the lack of supervision. We also provide an ablation analysis to examine the utility of each component in our system. The content of this chapter was published in Chung et al. [2019c].

4.1 Background

4.1.1 Speech-to-Text Translation

Conventional speech-to-text translation (ST) systems typically cascade automatic speech recognition (ASR) and machine translation (MT), and therefore impose significant requirements on training data [Waibel and Fugen, 2008]. They usually require hundreds of hours of transcribed audio and millions of words of parallel text from the source and target languages to train individual components, which makes it difficult to use this approach on low-resource languages. Although recent works have shown the feasibility of building end-to-end systems that directly translate source speech to target text without using any intermediate source language transcriptions, they still require data in the form of source audio paired with target text translations for end-to-end training [Weiss et al., 2017, Bansal et al., 2018, Bérard et al., 2018, 2016].

4.1.2 Unsupervised Machine Translation

In contrast to ST, which requires paired data for training, recent research in MT has explored fully unsupervised settings—relying only on monolingual corpora from each language. They have shown that unsupervised MT models can achieve comparable (sometimes even superior) results to supervised ones [Lample et al., 2018b, Artetxe et al., 2018b]. A key principle behind these unsupervised MT approaches is to initialize a MT model with a bilingual dictionary inferred from monolingual corpora, without using cross-lingual signals [Conneau et al., 2018, Artetxe et al., 2018a]. Given a source word, the initial MT model is able to perform word-by-word translation by looking up the dictionary, and can be further improved by leveraging other

techniques such as back translation [Sennrich et al., 2016].

4.1.3 Towards Unsupervised Speech-to-Text Translation

In Chapter 3, we showed that the unsupervised bilingual dictionary induction algorithms originally proposed for unsupervised MT could also be applied to scenarios where the source and target corpora are of different modalities, namely speech and text. The learned *cross-modal* bilingual dictionary, as we will show here, is capable of performing word-by-word translation, with the difference being that the input, instead of text, is a speech segment corresponding to a spoken word in the source language. In this chapter we propose a framework for building a ST system using only independent monolingual corpora of speech and text. The two corpora can be collected independently which greatly reduces human labeling efforts. Our framework starts by initializing a ST system with a cross-modal bilingual dictionary inferred from the monolingual corpora to perform word-by-word translation. To further improve the quality of the translations, we incorporate a pre-trained language model (LM) and sequence denoising autoencoder (DAE) [Sutskever et al., 2014, Vincent et al., 2008] that contain prior knowledge about the target language; their primary function is to consider context in lexical choices and handle local reordering and multi-aligned words. To the best of our knowledge, this is the first work that tackles ST in an unsupervised setting. More importantly, experiments show that our unsupervised system achieves comparable results to supervised end-to-end models [Bérard et al., 2018] despite the lack of supervision.

4.2 Proposed Framework

Our framework builds on several recently developed techniques for unsupervised speech processing and MT. We first derive a ST system that can perform simple word-by-word translation. Next, we integrate a language model into the framework to introduce contextual information during the translation process. Finally, we post-process the translated results using a DAE to handle local reordering and

multi-aligned words. Below we describe each step in detail.

4.2.1 Word-by-Word Translation

In our framework, a speech corpus from the source language is first pre-processed using an unsupervised speech segmentation algorithm [Kamper et al., 2017b] to generate speech segments corresponding to spoken words. We then apply Speech2Vec to learn a speech embedding space from the set of speech segments such that each vector corresponds to a word whose semantics has been captured. A text embedding space that captures word semantics can be learned by training Word2Vec [Mikolov et al., 2013b] on a text corpus from the target language. Based on the assumption that monolingual word embedding spaces are approximately isomorphic, since languages are used to convey thematically similar information in similar contexts [Barone, 2016], it is theoretically possible to align these two spaces.

To achieve this, one can use an unsupervised bilingual dictionary induction (BDI) algorithm to learn a cross-lingual mapping from the source embedding space to the target embedding space. Two of the most representative BDI algorithms are MUSE [Conneau et al., 2018] and VecMap [Artetxe et al., 2018a], neither of which rely on cross-lingual signals. Note that both these BDI algorithms were originally proposed for aligning two embedding spaces learned from text. In Chapter 3, we show that MUSE can also be applied to learn a *cross-modal* alignment between embedding spaces learned from speech and text. In our experiments, we include the results of both algorithms for comparison.

We obtain a rudimentary ST system after deriving a cross-modal and cross-lingual mapping from speech to the text corpora, which is essentially a linear transformation W. Given an unseen speech utterance, we first segment it into several speech segments using the speech segmentation algorithm previously mentioned. Then, for each speech segment that potentially corresponds to a spoken word, we map it from the speech embedding space to the text embedding space via W and apply nearest neighbor search to decide its text translation. However, the translations generated by this preliminary system are far from acceptable since nearest neighbor search does

not consider the context of the current word. In many cases, the correct translation is not the nearest target word but synonyms or other close words with morphological variations, prompting us to incorporate further improvements.

4.2.2 Language Model for Context-Aware Beam Search

We incorporate contextual information into word-by-word translation by introducing a LM during the decoding process [Kim et al., 2018]. Let w_s be the word vector mapped from speech to the text embedding space and w_t the word vector of a possible target word. Given a history h of target words before w_t, the score of w_t being the translation of w_s is computed as:

$$LM(w_t; w_s, h) = \log \frac{f(w_s, w_t) + 1}{2} + \lambda_{\text{LM}} \log p(w_t|h), \tag{4.1}$$

where λ_{LM} is the weight parameter that decides how *context-aware* the system is, and $f(w_s, w_t) \in [-1, 1]$ is the cosine similarity between w_s and w_t, linearly scaled to the range $[0, 1]$ to make it comparable with the output probability of the LM. Empirically, we found that setting λ_{LM} to 0.1 yields the best performance. Accumulating the scores per position, we perform a beam search to allow only reasonable translation hypotheses.

4.2.3 Sequence Denoising Autoencoder

We may achieve semantic correctness through learning an appropriate cross-modal bilingual dictionary and using a LM. However, to further improve the quality of the translations, it is also necessary to consider syntactic correctness. To this end, we apply a sequence DAE to correct the translated outputs. By injecting noise to the input sequence during the training process, the DAE learns to output the original (clean) sequence given a corrupted, noisy input. In our framework, we adopt three noise simulation techniques proposed in Kim et al. [2018]: word insertion, deletion and permutation. We seek to simulate the noise introduced during the word-by-word translation process with these three techniques. Readers can refer to Kim et al. [2018]

for more details. Along with the context-aware LM, we found that adopting a DAE further boosts translation performance.

4.3 Experiments

4.3.1 Data and Preprocessing

We used an English-to-French speech translation dataset [Kocabiyikoglu et al., 2018] augmented from the LibriSpeech ASR corpus [Panayotov et al., 2015]. The dataset is split into train, dev, and test sets; all come with a collection of English speech utterances and their corresponding French text translations. The train set contains 100 hours of speech, which was used to train Speech2Vec to obtain the speech embedding space. For the text embedding space, we trained Word2Vec on two different corpora—the parallel corpus that contains the text translations, and an independent corpus crawled from French Wikipedia. For evaluation, we merged the dev and test sets, resulting in speech data of about 6 hours. BLEU scores [Papineni et al., 2002] were used as the evaluation metric.

4.3.2 Model Implementation and Setup

We trained Speech2Vec following the same procedure used in Chapter 3. The text embedding space was trained by Word2Vec using the fastText implementation [Bojanowski et al., 2017] with default settings without subword information. The dimension of both speech and text embeddings is 100. For both VecMap [Artetxe et al., 2018a] and MUSE [Conneau et al., 2018], we followed the default settings of the implementations released by their original authors. For the LM, we trained a 5-gram count-based LM using KenLM [Heafield, 2011] with its default settings. Finally, we implemented the DAE, structured as a 6-layer Transformer [Vaswani et al., 2017], with embedding and hidden layer size of 512, a feedforward sublayer size of 2,048, and 8 attention heads.

4.3.3 Results and Discussions

We first study the similarities between different pairs of embedding spaces to be aligned. We then present the main ST results.

Table 4.1: Embedding similarity of different speech and text embeddings pair evaluated by eigenvector similarity. We denote the embedding training method and corpus name in upper and lower case, respectively. For the pair, we denote the speech and text embedding space at the left and right side, respectively. For example, A_{libri} - T_{wiki} represents the speech embedding space trained on the LibriSpeech corpus using Audio2Vec and the text embedding space trained on Wikipedia corpus. A, S, T indicates Audio2Vec, Speech2Vec and text (Word2Vec) embedding.

Speech & text embedding spaces pair	Eigenvector similarity
A_{libri} - T_{libri}	14.74
A_{libri} - T_{wiki}	15.02
S_{libri} - T_{libri}	6.43
S_{libri} - T_{wiki}	7.17

Having approximately isomorphic embedding spaces is important for BDI. To quantify whether the embedding spaces are isomorphic, or similar in structure, we computed the eigenvector similarity, which is derived from Laplacian eigenvalues. Both our study and Søgaard et al. [2018] demonstrate that the eigenvector similarity metric is correlated to the performance of the translation task, which implies that the metric reflects the distance between embedding spaces in a meaningful way. The similarity is computed as follows. Let L_1 and L_2 be the Laplacians of two nearest neighbor embedding graphs. We search for the smallest value of k for each graph such that the sum of largest k Laplacian eigenvalues is smaller than 90% of the Laplacian eigenvalues. Then, we select the smallest k across two graphs and compute the squared differences between the largest k Laplacian eigenvalues in two graphs. The differences is the eigenvector similarity we use to measure the similarity between embedding spaces. Note that a *higher* value of the eigenvector similarity metric indicates that the given two embedding spaces are *less* similar.

Table 4.1 presents the eigenvector similarity of different speech-text pairs. The eigenvector similarity of speech and text embedding space pairs is smaller when we trained the speech embedding using the Speech2Vec algorithm than the Au-

dio2Vec [Chung et al., 2016] algorithm. These results are expected since Speech2Vec utilizes semantic context of the speech corpus, similarly to how Word2Vec uses that of the text corpus. Furthermore, we applied skipgrams as a training methodology for both algorithms, resulting in isomorphic embedding spaces. In contrast, Audio2Vec focuses on similarities in acoustics rather than semantics, thus the learned embedding space differs fundamentally. Embedding space pairs learned from comparable corpora also yield higher similarity, since the word distributions are more similar; for example, the distribution of English LibriSpeech speech embeddings is more similar to that of the French LibriSpeech text embeddings than French Wikipedia text embeddings.

Table 4.2: Different configurations for speech-to-text translation and their performance. The numbers in the section of unsupervised methods denoted as BLEU score (%) of VecMap / BLEU score (%) of MUSE. The notation used in the Table is the same as Table 4.1. For cascaded systems, we followed the ASR and MT pipeline in Bérard et al. [2018]. E2E stands for end-to-end.

	System	Best	Average
Cascaded and end-to-end ST systems (supervised)			
(a)	Cascaded + greedy	13.7	13.0
(b)	Cascaded + beam	14.2	13.2
(c)	E2E + greedy	12.3	11.6
(d)	E2E + beam	12.7	12.1
Our alignment-based ST systems (unsupervised)			
(e)	A_{libri} - T_{libri}	0.0 / 0.0	0.0 / 0.0
(f)	A_{libri} - T_{wiki}	0.0 / 0.0	0.0 / 0.0
(g)	S_{libri} - T_{libri}	4.5 / 4.6	4.2 / 2.7
(h)	S_{libri} - T_{wiki}	3.7 / 2.1	3.0 / 0.9
(i)	(g) + LM_{libri}	5.2 / 5.0	4.7 / 2.9
(j)	(g) + LM_{wiki}	9.5 / 8.8	9.0 / 5.7
(k)	(g) + LM_{wiki} + DAE_{wiki}	12.2 / 11.8	11.3 / 7.3
(l)	(h) + LM_{wiki} + DAE_{wiki}	11.5 / 9.1	10.8 / 6.2

We present the results of our unsupervised approach as well as supervised baselines in Table 4.2. We trained every system 10 times and report both the best and average performance. In configurations (a-d), we replicate state-of-the-art supervised algorithms and arrived at the conclusion that cascaded systems perform better than their end-to-end counterparts and beam search performs better than greedy search. Note

that cascaded systems require more supervision than end-to-end systems, whereas our approach makes no assumptions of having speech-text or language pairs of the comparable corpora.

In configurations (e-l), we showcase the performance of our unsupervised approach, denoted as (BLEU score of VecMap / BLEU score of MUSE) in the columns of Table 4.2.

Alignment Quality Configurations (e-h) demonstrate that eigenvector similarity of speech and text embedding space pairs have strong positive correlation, namely comparing the relative performances to those shown in Table 4.1, with the BLEU score of alignment-based ST tasks. The results, from configurations (g) and (h), illustrates that using comparable corpora, and thus better alignment, affects the quality of ST. It also hints that there may exist a threshold of usefulness in alignment performances. Since configurations (e) and (f) lie underneath that threshold, they achieve scores of zero. These findings indicate that eigenvector similarity of embedding spaces could serve as an indicator of unsupervised ST performance.

Unsupervised BDI In all of our unsupervised experiments, we compared the performance between two unsupervised BDI algorithms, VecMap and MUSE. VecMap outperforms MUSE in all but one experiment, demonstrating that VecMap can be applied to more difficult scenarios through weak, fully unsupervised initialization with iterative mapping improvements, whereas MUSE, which maps embeddings to the shared space through adversarial training, could only succeed on a more limited set of conditions. Additionally, VecMap trains more stably and faster than MUSE, which has a similar best performance but much lower average performance.

Language Model Integration Integrating a LM improves the performance of ST in all experimental configurations, regardless of the selection of corpus, configurations (g) versus (i) and (j); configurations (h) versus (l) generalize this result to different embedding spaces. By comparing configurations (i) and (j), we discover that the text corpus used to train the LM does not need to be the same as the one used for

Word2Vec text embedding space training. In fact, adopting the LM trained on the Wikipedia corpus (LM_{wiki}) produces better performance than using that trained on the LibriSpeech corpus (LM_{libri}). Since introducing the LM grounds words into a context based on the previous word, the much larger LM_{wiki}, containing more words, topic contexts, and sentence structures, serves as a better approximation of the French language than LM_{libri}.

Sequence DAE In configurations (j) versus (k), we show that applying DAE on top of the baseline alignment architecture and LM can further enhance performance in unsupervised ST; the performance is now comparable to end-to-end supervised systems. This also justifies our alignment and post-processing approach since configuration (k) essentially has the same degree of supervision as configurations (c) and (d) and performs similarly well while employing a completely different approach. We attribute this to the DAE's ability to reconstruct corrupted data after translation. Since the semantic alignment method we used may retrieve synonyms based on context, rather than the exact syntactically correct word, it is possible that the output even when taking the LM into account is still syntactically incorrect. Moreover, one of the key obstacles in training Speech2Vec lies in the limited performance of unsupervised speech segmentation methods. By incorporating a DAE, we could limit these negative effects after translation. Last but not least, the DAE was trained on LM_{wiki} rather than LM_{libri}. This design decision follows from the observation of the LM corpus choice: since the DAE should learn the French language, a larger, more diverse dataset would perform better than the same dataset used for Word2Vec text embeddings.

Scenario of Real-World ST In configuration (l), we conducted experiments modeling a real-world setting where there exists no comparable speech and text corpora. Instead, we need to collect them independently from different sources. Text data exists in more abundance than speech data and thus we usually adopt the text embedding learned from larger corpus such as Wikipedia, which configuration (h) replicates

to our best efforts. By comparing configurations (k) and (l), we demonstrate that the performance of our proposed framework under no supervision is only slightly inferior to the best performance achieved using unsupervised alignment, which requires comparable corpora for speech and text embedding spaces and should be considered supervised. The proposed unsupervised ST framework is thus promising for low language resource ST.

4.4 Conclusions

In this chapter, we propose a framework capable of performing speech-to-text translation in a completely unsupervised manner. Since the system translates using an inferred cross-modal bilingual dictionary trained without parallel data between speech and text, it could be applied to low or zero-resource languages. By incorporating knowledge of the target language, through adding a LM and a DAE, both are intrinsically unsupervised, our system greatly enhances the translation performance: We achieve comparable performance with state-of-the-art end-to-end systems using parallel corpora and only slightly lower scores without it. These results indicate that our approach could serve as a promising first step towards fully unsupervised speech-to-text translation.

Chapter 5

Conclusions and Future Work

5.1 Summary of Contributions

In this thesis, we explore unsupervised learning of automatic sequence transduction between speech and text. The framework relies only on monolingual corpora of speech and text that do not need to be parallel, and is hence applicable to low- and zero-resource languages. Specifically, the framework consists of the following three steps where each step is by itself unsupervised:

1. Individually learn two embedding spaces of speech and text that both reveal word semantics and relationships of languages.

2. Exploit the geometrical similarity exhibited in the two embedding spaces and learn a cross-modal alignment between them via adversarial training followed by a refinement procedure.

3. With the alignment learned in the previous step, one can map a spoken word in the speech domain into the text domain (and theoretically, vice versa) and retrieve its recognized result or translation in another language.

For the first step, in Chapter 2, we draw inspiration from Word2Vec [Mikolov et al., 2013b], which learns word embeddings from text, and design a novel Speech2Vec model that shares the same training methodologies with Word2Vec but with archi-

tecture adapted to handle speech data. One may think that speech and text are just two different ways for expressing languages and thus underestimate the power of the proposed Speech2Vec. In fact, factors like vocal tract differences across speakers, speaking styles, contextual differences, and environmental conditions all make speech signal much more complicated than pure text. It is therefore surprising and exciting to see our proposed Speech2Vec is able to, at least to some extent, factor out these inherent variability in speech production and preserve the semantic information of spoken words in a latent space [Chrupała et al., 2019], as shown by the results on word similarity benchmarks and visualization of the learned embeddings.

For the second and third steps, in Chapter 3, we propose a framework for learning a linear transformation that maps a vector corresponding to a spoken word from the speech embedding space to its correspondence in the text embedding space. The core idea is to use adversarial training—a two-player game between a generator and a discriminator—so as to make the two embedding spaces indistinguishable. The learned alignment were used to perform spoken word recognition and translation as example applications. From the experimental results, it is noteworthy that word embedding spaces not only exhibit similar structures across languages [Mikolov et al., 2013a], but also across different modalities (speech and text).

To show how we could use the proposed cross-modal alignment framework for real-world applications, in Chapter 4, we present a completely unsupervised speech-to-text translation system developed using only monolingual speech and text corpora. We combine the alignment framework, which can already perform speech-to-text translation at a word level, with a language model pre-trained on large corpora of text in the source language to generate full sentences. To produce more robust results, we further incorporate a sequence denoising autoencoder that is pre-trained to denoise three types of artificial noises. We achieve performance only slightly worse than the state-of-the-art supervised end-to-end systems. The results indicate that our approach could serve as a promising first step towards fully unsupervised speech-to-text translation.

5.2 Future Work

This thesis work can be extended in several directions.

First of all, speech embeddings learned by Speech2Vec contain semantic information about the underlying spoken words, making them potentially useful for downstream tasks that require language understanding from speech input such as spoken question answering [Lee et al., 2018a,b] and machine comprehension of spoken content [Tseng et al., 2016, Chung et al., 2018a]. The pre-trained speech embeddings, which we have already released online along with our source code, can be used in a similar fashion as how we use pre-trained word vectors (e.g., those learned by the Word2Vec model) in NLP tasks.

The Speech2Vec model itself also requires more studies. As pointed out in Chapter 2, unlike pure text, speech signals inherently contain plenty of complex variabilities caused by speakers and environmental conditions. It is still not clear to us how Speech2Vec is able to remove those variabilities (at least to some degree) and preserve only the word semantics. Furthermore, an interesting aspect of Speech2Vec we did not investigate in this thesis is to use the Speech2Vec decoder as a generative model. Given a speech embedding, being able to reconstruct meaningful acoustic feature sequence (e.g., spectrogram) would make Speech2Vec an invertible function (encoding and decoding), allowing Speech2Vec to be applied to an even wider range of applications.

Regarding our unsupervised cross-modal alignment framework, as indicated in our experimental results in Chapter 3, an essential step to obtain a high-quality mapping is to devise unsupervised speech segmentation approaches that produce more accurate word segments. However, although unsupervised speech segmentation has attracted quite a few attention recently [Godard et al., 2018, Kamper et al., 2017b, 2016a], it remains one of the most challenging tasks in the field of zero-resource speech processing and requires more future effort. Additionally, in this thesis, we obtained the speech embeddings in two steps: we first segmented all the speech utterances in a speech corpus into word segments—either by referring to word boundaries obtained

by forced alignment with respect to the reference transcriptions, or by applying off-the-shelf unsupervised speech segmentation algorithms—and Speech2Vec was trained on those pre-segmented speech segments. We are currently working on designing a model that jointly optimizes the estimation on word boundaries and Speech2Vec training.

Last but not least, we seek to explore other applications of our unsupervised cross-modal alignment framework besides speech-to-text translation presented in this thesis. Theoretically, the framework can be applied to any task whose goal is to transcribe a sequence of tokens from a source domain to another in the target domain, where one domain belongs to speech and the other belongs to text. With the proposed framework, one only needs to collect non-parallel corpora that are sufficiently large from the two domains to build the sequence transduction system. We are currently interested in unsupervised speech recognition and text-to-speech synthesis for low- or zero-resource languages.

Appendix A

Word Similarity

A.1 Basic Idea

The method of measuring word semantic similarity is based on the idea that the distances between words in an embedding space can be evaluated through human judgments on the actual semantic distances between these words. For instance, the distance between "cup" and "mug" defined in an continuous interval $[0, 1]$ would be 0.8 since these words are synonymous, but not really the same thing. When collecting a word similarity dataset, the human assessor is given a set of pairs of words and asked to assess the degree of similarity for each pair by assigning it a real value within a certain interval. The distances between these pairs are also collected in a word embeddings space, where the similarity between a given pair of words is measured by the cosine similarity[1] between their corresponding word embeddings. A rank correlation (e.g., Spearman's rank correlation coefficient [Myers and Well, 1995]) between the two obtained distance sets is calculated. The higher the rank is, the more similar the two distance sets are, and thus the better the embeddings are at capturing word semantics.

[1] $\cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$, where $\mathbf{u}$ and $\mathbf{v}$ are the word embeddings of the two words.

A.2 Benchmarks

The 13 word similarity benchmark datasets used in Chapter 2 are listed here with their size (number of word pairs) and scale (the range of the similarity value assigned by human assessor when collecting the dataset). There are quite a few publicly available toolkits that include all these benchmarks. The one we used in this thesis is `https://github.com/mfaruqui/eval-word-vectors`.

1. **WordSim-353**: 353 pairs of words, scale $\in [0, 10]$
2. **WordSim-353-REL**: 252 pairs, a subset of WordSim-353
3. **WordSim-353-SIM**: 203 pairs, a subset of WordSim-353
4. **RG-65**: 65 pairs of words, scale $\in [0, 4]$
5. **MC-30**: 30 pairs, a subset of RG-65
6. **Rare-Word**: 2034 pairs of words with low occurrences, scale $\in [0, 10]$
7. **MEN**: 3000 pairs of words, scale $\in \{0, 1, 2, ..., 50\}$
8. **MTurk-287**: 287 pairs of words, scale $\in [0, 5]$
9. **MTurk-771**: 771 pairs of words, scale $\in [0, 5]$
10. **YP-130**: 130 pairs of *verbs*, scale $\in [0, 4]$
11. **SimLex-999**: 999 pairs of words, scale $\in [0, 10]$
12. **Verb-143**: 143 pairs of *verbs*, scale $\in [0, 4]$
13. **SimVerb-3500**: 3500 pairs of *verbs*, scale $\in [0, 4]$

Bibliography

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *ACL*, 2017.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*, 2018a.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *ICLR*, 2018b.

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

Miguel Ballesteros, Chris Dyer, and Noah Smith. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *EMNLP*, 2015.

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

Antonio Valerio Miceli Barone. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. In *RepL4NLP*, 2016.

Samy Bengio and Georg Heigold. Word embeddings for speech recognition. In *Interspeech*, 2014.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan. 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.

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

David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Hailong Cao, Tiejun Zhao, Shu Zhang, and Yao Meng. A distribution-based model to learn bilingual word embeddings. In *COLING*, 2016.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*, 2016.

Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li. Parallel inference of dirichlet process gaussian mixture models for unsupervised acoustic modeling: A feasibility study. In *Interspeech*, 2015.

Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung-Yi Lee, and Lin-shan Lee. Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval. In *SLT*, 2018.

Chung-Cheng Chiu, Tara Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art speech recognition with sequence-to-sequence models. In *ICASSP*, 2018.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NIPS*, 2015.

Jan Chorowski, Ron Weiss, Samy Bengio, and Aäron van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *arXiv preprint arXiv:1901.08810*, 2019.

Grzegorz Chrupała, Lieke Gelderloos, and Afra Alishahi. Representations of language in a model of visually grounded speech signal. In *ACL*, 2017.

Grzegorz Chrupała, Lieke Gelderloos, Ákos Kádár, and Afra Alishahi. On the difficulty of a distributional semantics of spoken language. In *SCiL*, 2019.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*, 2014.

Yu-An Chung and James Glass. Learning word embeddings from speech. In *NIPS Workshop on Machine Learning for Audio Signal Processing*, 2017.

Yu-An Chung and James Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In *Interspeech*, 2018.

Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee. Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. In *Interspeech*, 2016.

Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. In *NAACL-HLT*, 2018a.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. Unsupervised cross-modal alignment of speech and text embedding spaces. In *NeurIPS*, 2018b.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240*, 2019a.

Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and RJ Skerry-Ryan. Semi-supervised training for improving data efficiency in end-to-end speech synthesis. In *ICASSP*, 2019b.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. Towards unsupervised speech-to-text translation. In *ICASSP*, 2019c.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *ICLR*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. In *ICLR Workshop Track*, 2015.

Jennifer Drexler and James Glass. Combining end-to-end and adversarial training for low-resource speech recognition. In *SLT*, 2018.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. In *EMNLP*, 2016.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4):112, 2018.

Manaal Faruqui and Chris Dyer. Community evaluation and exchange of word vectors at wordvectors.org. In *ACL System Demonstrations*, 2014a.

Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *EACL*, 2014b.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *RepEval*, 2016.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.

James Glass. Towards unsupervised speech processing. In *ISSPA*, 2012.

Pierre Godard, Marcely Zanon Boito, Lucas Ondel, Alexandre Berard, François Yvon, Aline Villavicencio, and Laurent Besacier. Unsupervised word segmentation from speech with attention. In *Interspeech*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

David Harwath and James Glass. Deep multimodal semantic embeddings for speech and images. In *ASRU*, 2015.

David Harwath and James Glass. Learning word-like units from joint audio-visual analysis. In *ACL*, 2017.

David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *NIPS*, 2016.

Wanjia He, Weiran Wang, and Karen Livescu. Multi-view recurrent neural acoustic word embeddings. In *ICLR*, 2017.

Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *WMT*, 2011.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Nils Holzenberger, Mingxing Du, Julien Karadayi, Rachid Riad, and Emmanuel Dupoux. Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments. In *Interspeech*, 2018.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *ACL*, 2018.

Wei-Ning Hsu, Yu Zhang, and James Glass. Learning latent representations for speech generation and transformation. In *Interspeech*, 2017a.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *NIPS*, 2017b.

Wei-Ning Hsu, Yu Zhang, Ron Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. In *ICASSP*, 2019.

Jui-Ting Huang and Mark Hasegawa-Johnson. Semi-supervised training of gaussian mixture models by conditional entropy minimization. In *Interspeech*, 2010.

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

Herman Kamper. Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models. In *ICASSP*, 2019.

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

Herman Kamper, Weiran Wang, and Karen Livescu. Deep convolutional acoustic word embeddings using word-pair side information. In *ICASSP*, 2016b.

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

Herman Kamper, Karen Livescu, and Sharon Goldwater. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. In *ASRU*, 2017b.

Herman Kamper, Shane Settle, Gregory Shakhnarovich, and Karen Livescu. Visually grounded learning of keyword prediction from untranscribed speech. In *Interspeech*, 2017c.

Herman Kamper, Gregory Shakhnarovich, and Karen Livescu. Semantic speech retrieval with a visually grounded model of untranscribed speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):89–98, 2019.

Shigeki Karita, Shinji Watanabe, Tomoharu Iwata, Atsunori Ogawa, and Marc Delcroix. Semi-supervised end-to-end speech recognition. In *Interspeech*, 2018.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. Character-aware neural language models. In *AAAI*, 2016.

Yunsu Kim, Jiahui Gend, and Hermann Ney. Improving unsupervised word-by-word translation with language model and denoising autoencoder. In *EMNLP*, 2018.

Ali Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. Augmenting librispeech with french translations: A multimodal corpus for direct speech translation evaluation. In *LREC*, 2018.

Arne Köhn, Florian Stegen, and Timo Baumann. Mining the spoken wikipedia for speech data and beyond. In *LREC*, 2016.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural amr: Sequence-to-sequence models for parsing and generation. In *ACL*, 2017.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL-HLT*, 2016.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018a.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *EMNLP*, 2018b.

Thomas Landauer, Peter Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, 1998.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. ODSQA: Open-domain spoken question answering dataset. In *SLT*, 2018a.

Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung-Yi Lee. Spoken SQuAD: A study of mitigating the impact of speech recognition errors on listening comprehension. In *Interspeech*, 2018b.

Chia-Ying Lee and James Glass. A nonparametric bayesian approach to acoustic model discovery. In *ACL*, 2012.

Chia-Ying Lee, Timothy O'Donnell, and James Glass. Unsupervised lexicon discovery from acoustic input. *Transactions of the Association for Computational Linguistics*, 3:389–403, 2015.

Keith Levin, Katharine Henry, Aren Jansen, and Karen Livescu. Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings. In *ASRU*, 2013.

Bo Li, Tara Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean Chin, Khe Chai Sim, Ron Weiss, Kevin Wilson, Ehsan Variani, Chanwoo Kim, Olivier Siohan, Mitchel Weintraub, Erik McDermott, Richard Rose, and Matt Shannon. Acoustic modeling for google home. In *Interspeech*, 2017.

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

Vince Lyzinski, Gregory Sell, and Aren Jansen. An evaluation of graph clustering methods for unsupervised term discovery. In *Interspeech*, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Tomas Mikolov, Quoc Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

Benjamin Milde and Chris Biemann. Unspeech: Unsupervised speech context embeddings. In *Interspeech*, 2018.

Jerome Myers and Arnold Well. *Research design and statistical analysis*. Routledge, 1 edition, 6 1995.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP*, 2015.

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

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

Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *arXiv preprint arXiv:1904.03416*, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, 2018.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *ACL*, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

Okko Räsänen, Gabriel Doyle, and Michael Frank. Unsupervised word discovery from speech using automatic segmentation into syllable-like units. In *Interspeech*, 2015.

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

George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall. English conversational telephone speech recognition by humans and machines. In *Interspeech*, 2017.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *EMNLP*, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2016.

Shane Settle and Karen Livescu. Discriminative acoustic word embeddings: Recurrent neural network-based approaches. In *SLT*, 2016.

Jonathan Shen, Ruoming Pang, Ron Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerrv-Ryan, Rif Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, 2018.

Samuel Smith, David Turban, Steven Hamblin, and Nils Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *ICLR*, 2016.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the limitations of unsupervised bilingual dictionary induction. In *ACL*, 2018.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *ICLR*, 2018.

Amarnag Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. *Journal of Machine Learning Research*, 12(Nov):3311–3370, 2011.

Meng Sun and Hugo Van hamme. Joint training of non-negative tucker decomposition and discrete density hidden markov models. *Computer Speech and Language*, 27 (4):969–988, 2013.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Bo-Hsiang Tseng, Sheng-Syun Shen, Hung-Yi Lee, and Lin-Shan Lee. Towards machine comprehension of spoken content: Initial toefl listening comprehension test by machine. In *Interspeech*, 2016.

Balakrishnan Varadarajan, Sanjeev Khudanpur, and Emmanuel Dupoux. Unsupervised learning of acoustic sub-word units. In *ACL*, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *CVPR*, 2015.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

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

Oliver Walter, Timo Korthals, Reinhold Haeb-Umbach, and Bhiksha Raj. A hierarchical system for word discovery exploiting dtw-based initialization. In *ASRU*, 2013.

Yu-Hsuan Wang, Hung-Yi Lee, and Lin-Shan Lee. Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection. In *ICASSP*, 2018.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif Saurous. Tacotron: Towards end-to-end speech synthesis. In *Interspeech*, 2017.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Kaiser ukasz Liu, Xiaobing, 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, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *NAACL-HLT*, 2015.

Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Michael Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423, 2017.

Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex Acero. Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Computer Speech and Language*, 24(3):433–444, 2010.

Xiang Yu and Ngoc Thang Vu. Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages. In *ACL*, 2017.

Neil Zeghidour, Gabriel Synnaeve, Maarten Versteegh, and Emmanuel Dupoux. A deep scattering spectrum-deep siamese network pipeline for unsupervised acoustic modeling. In *ICASSP*, 2016.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*, 2017a.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *EMNLP*, 2017b.

Yaodong Zhang and James Glass. Towards multi-speaker unsupervised speech pattern discovery. In *ICASSP*, 2010.