Learning Spoken Language Through Vision

by

David Frank Harwath

B.S., University of Illinois at Urbana-Champaign (2010)
S.M., Massachusetts Institute of Technology (2013)

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
February 28, 2018

Certified by . . . .
James R. Glass
Senior Research Scientist, CSAIL
Thesis Supervisor

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

Learning Spoken Language Through Vision

by

David Frank Harwath

Submitted to the Department of Electrical Engineering and Computer Science
on February 28, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Humans learn language at an early age by simply observing the world around them. Why can't computers do the same? Conventional automatic speech recognition systems have a long history and have recently made great strides thanks to the revival of deep neural networks. However, their reliance on highly supervised (and therefore expensive) training paradigms has restricted their application to the major languages of the world, accounting for a small fraction of the more than 7,000 human languages spoken worldwide. This thesis introduces datasets, models, and methodologies for grounding continuous speech signals at the raw waveform level to natural image scenes. The context and constraint provided by the visual information enables our models to efficiently learn linguistic units, such as words, along with their visual semantics. For example, our models are able to recognize instances of the spoken word "water" within spoken captions and associate them with image regions containing bodies of water. Further, we demonstrate that our models are capable of learning cross-lingual semantics by using the visual space as an interlingua to perform speech-to-speech retrieval between English and Hindi. In all cases, this learning is done without linguistic transcriptions or conventional speech recognition - yet we show that our methods achieve retrieval scores close to what is possible when transcriptions are available. This offers a promising new direction for speech processing that only requires speakers to provide narrations of what they see.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist, CSAIL

Acknowledgments

I am immensely grateful to my doctoral advisor, Jim Glass, for providing me with the opportunity, resources, and mentorship that made this thesis possible. Jim has been an ideal advisor. He is patient and kind when offering guidance, and always manages to see research problems from a unique angle. When you bring a single seed of an idea into a discussion with him, he is always able to help you multiply it into a myriad of fruitful ideas. He makes time for his students whenever they need it, and when he isn't meeting with his students he is doing everything he can to keep their research paths clear of roadblocks. He leads the Spoken Language Systems group in a way that cultivates collaboration, mutual support, and friendship among its members.

I am indebted to the other members of my thesis committee, Victor Zue and Antonio Torralba, for their razor-sharp insights that helped me to align my thinking to the big picture when conducting, presenting, and writing about my research. Thank you so much to Victor for all of the time you have taken to discuss my work, edit my writing, and impart sage advice as I continue forward in my career. As a speech researcher, venturing into the realm of vision was stepping outside the bounds of my expertise; Antonio has been a superb collaborator and guide for me to this new world. I also want to acknowledge Antonio for helping to craft several of the figures in Chapters 5 and 6.

Thanks to TJ Hazen for hosting me as an intern at Lincoln Laboratory, for continuing on to co-supervise my Master's work, and for helping to teach me the fundamentals of being a good researcher. Thanks to Scott Cyphers for the technical assistance over the years, and for all of the history lessons. Thanks to Najim Dehak for teaching me about speaker recognition, and for the energy that he brought to the SLS offices as well as our social events. Thank you to Marcia Davidson for making the administrative aspects of my career as a graduate student as painless as possible, and for the many thought-provoking (and entertaining) conversations we've shared over the years. Thank you to Rahul Yargop and Mark Hasegawa-Johnson for first introducing me to, and sparking my interest in research in speech recognition while I was still an undergraduate at UIUC.

Thank you to my academic siblings in SLS . I learned an enormous amount from you, and greatly enjoyed our time together. Thanks to Ekapol Chuangsuwanich for being the first person to welcome me to MIT, for teaching me so much about speech recognition, and for being a great friend. Thank you to Yaodong Zhang and Hung-An Chang for helping me to get oriented with the SLS computing environment and software as a first-year graduate student, and for being a great sounding board for ideas during my Master's work. Thanks to Michael Price, Yonatan Belinkov, Yu Zhang, Mandy Korpusik, Daniel Li, Wei-Ning Hsu, Stephen Shum, Ian McGraw, Timo Mertens, and Ann Lee for so many great conversations both in lab as well as over our many lunches, dinners, bowls of ice cream, and pints of beer together. Thank you to Mandy, Wei-Ning, and Hao Tang for proofreading this thesis. Their comments were invaluable, and I hope to be able to return the favor to Mandy and Wei-Ning when the time comes for them to write their dissertations. I want to acknowledge Tuka Al Hanai for being my longest running officemate (through 4 different offices!), and for the many productive research conversations that we shared as a result of that. Thank you to Ian and Timo for mentoring me when I was a new graduate student, for the many conversations about research (among plenty of other things), and for helping to organize a superb reading group. I am thankful for the opportunity to have mentored, worked with, and published with exceptionally talented undergraduate and MEng students during my PhD, including Felix Sun, Galen Chuang, Karan Kashyap, Matt McEachern, and Ken Leidal.

A special thanks goes out to Stephen Shum, not only for introducing me to my now-wife, but also for being a true friend and great research collaborator.

Thank you to my parents, Frank and Nancy, and my sister, Amy. It has been difficult to live so far away from you, but our telephone conversations kept me grounded and made that distance feel a bit smaller. Thank you for your endless love, support, and counsel.

Thank you to my wife, Sharon. You have loved and supported me through thick and thin. You are the light of my life, and the best friend I've ever had.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

The ability of human beings to communicate complex ideas to one another through spoken language is one of our defining characteristics. While animals such as bees and dolphins are capable of using simple forms of communication, human language is unique in terms of the range and complexity of ideas that it can be used to convey (Pinker, 1994). Language is what allows us to cooperate. It allows us to share new discoveries, so that we can collectively benefit from our individual insights. Language is how we express our thoughts to one another. By putting our mental state into words, which are then received by another person, we can induce a new mental state in that person. Through language, we can connect our brains together *across space and time*, forming networks of thought capable of solving problems far more complex than any one person could on their own.

Throughout our history, we have exploited our mental faculties to compensate for our physical deficiencies. We crafted tools from stone and metal to withstand the wear that our hands could not. We domesticated dogs and horses for their speed and strength, enabling us to hunt, travel, and farm more effectively than we otherwise could. But for many generations, our only mental collaborators were one another. The computer is the culmination of our efforts to build a new kind of tool - one that can help us *think*. Algorithmic computation has removed the tedium from calculations that once had to be done by hand, and to such a great extent the realm of what problems are "solvable" expanded dramatically (for example, see Appel and Haken (1977)). But the computer is still an impoverished intellectual partner. Its entire range of behavior must be completely specified a priori by a human mind, using a language far less expressive and far more esoteric than our own. Over the past 75 years, a new scientific field - artificial intelligence (AI) - has coalesced, with the aim of endowing computers with the capability to learn, think and act in a manner similar to humans. But in order to realize the computer's full potential as an intellectual partner in this regard, humans must be able to communicate with computers just as richly as we communicate with one another.

A prophetic vision of this level of communicative depth can be found in Stanley Kubrick's 1968 film *2001: A Space Odyssey.* HAL 9000 couldn't just hold a conversation with Dr. Dave Bowman and Dr. Frank Poole. HAL could sense their body language to infer their emotional state. He understood his environmental surroundings enough to know that when Dave and Frank entered an EVA pod to have a private conversation, it was likely because they didn't want him to hear what they were saying. Of course, that didn't matter much, because HAL could read their lips anyway. HAL's ultimately nefarious intentions aside, his ability to observe and communicate in a holistic way is what made him so compelling.

It is not hard to see the influence of science fiction AIs like HAL on the devices that dominate the consumer computing market today. You would be hard-pressed to find a person who hasn't asked Siri about the weather, told Alexa to turn on some music, or followed the spoken directions from Google Maps while driving. The technologies of automatic speech recognition (ASR), spoken language understanding (SLU), and test-to-speech synthesis (TTS) have progressed far enough to become a part of our everyday lives. In May of 2016, Apple's Siri had over 48.7 million unique users in the U.S. alone (Perez, 2017). Speech and language technology is no doubt one of the great success stories of artificial intelligence and machine learning.

That said, the technology is far from mature. While your favorite computerized personal assistant can tell you the weather, perform a web search, dictate a text message, or give you directions, it can't yet hold an open-ended conversation with you. You can't teach it new words, or to speak a new language just by talking to it - despite the fact that nearly every one of us learned language in that manner when we were babies. It can't grasp the significance of a joking wink, an indifferent shrug, or an eyebrow raised in skepticism. It can't understand the signing of a deaf user. It can't appreciate the universal truths conveyed by a piece of music or art. It can't watch you bake cookies and let you know that you've forgotten to add the chocolate chips, or help you decipher the assembly diagrams that accompany your IKEA furniture.

The reality is that our current technologies have barely modeled the tip of the iceberg that is holistic communication. Consider the pipeline depicted in Figure 1-1:

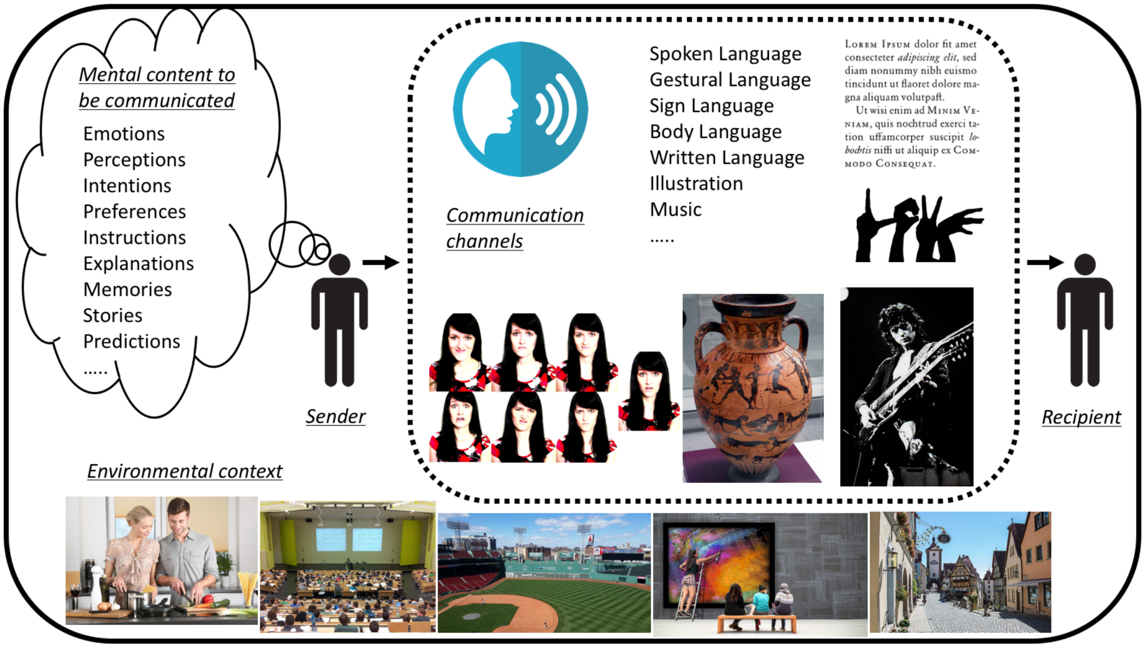

Mental content to be communicated
Emotions
Perceptions
Intentions
Preferences
Instructions
Explanations
Memories
Stories
Predictions
.....
Sender
Communication channels
Spoken Language
Gestural Language
Sign Language
Body Language
Written Language
Illustration
Music
.....
Lorem Ipsum dolor fit amet consecteter adipiscing elit, sed diam nonummy nibh euismo tincidunt ut flaoret dolore magna aliquam volutpaft.
Ut wisi enim ad Minim Veniam, quis nochtrud exerci tation uffamcorper suscipit lobodtis niffi ut aliquip ex Commodo Consequat.
Recipient
Environmental context

Figure 1-1: A schematic diagram of communication between people.

The many modes of communication such as speech, writing, illustration, etc. serve as the channels to transmit information from the mind of one person to another. No one mode is capable of capturing the full range of the human consciousness by itself. Each mode constitutes a lossy projection that can reflect only certain aspects of our thinking. It is by employing multiple modes of communication *in conjunction with one another* that we are able to infer another person's thoughts with the highest possible fidelity. And of course, neither thought nor communication ever exist within a vacuum; both are immoderately influenced by our surrounding environment.

To appreciate how tightly coupled these modes are to one another and their environmental context, think about the last time you watched a how-to video on YouTube to learn a new skill - such as cooking a meal or changing your car's oil. Would you have been able to learn that skill without the benefit of *both* the spoken narration and the visual demonstration? Why are Skype, Google Hangouts, Apple FaceTime, and other means of teleconferencing so popular when the telephone has been in use since the 19th century? Why haven't we all adopted texting as our sole means of communication and stopped speaking to one another altogether? Why do we still appreciate illustration, music, and all other forms of artwork if written language is sufficient to express our deepest meditations? It speaks volumes that the most popular artistic medium of our time - cinema - attempts to reflect the human experience as fully as possible by weaving together drama, language, and music into a greater whole.

Historically, modality has served as one of the primary dividing lines between the sub-disciplines of machine learning: speech processing, natural language processing, computer vision, robotic locomotion, and so on. This is partly because each of these problem domains requires domain-specific knowledge to effectively model. It is also due to the historic limitations of computational processing power, data storage capacity, data collection capability, and our mathematical models themselves. Within the last decade, these technological barriers have all but disappeared. The latest GPU architectures offer over 100 teraflops of computational power on a single card (Carbotte, 2017), and the cost of disk storage on modern hard drives is pennies per gigabyte (Klein, 2017). Wireless internet access is ubiquitous worldwide, and as of 2018 more than a third of the world's population owns a smartphone (Statista, 2018). Deep neural networks have proven themselves capable of seamlessly integrating inputs from disparate modalities into unified models. As the Internet of Things continues to expand and integrate a greater variety of sensory capabilities, so do the opportunities for large-scale, distributed, multimodal data collection. We need new ways of making sense of this ever-growing flood of data, not only to gain new insights today but also to preserve our knowledge for future generations. The fruit is ripe for the picking; the time to bridge the gaps between the various sub-disciplines of artificial intelligence is now.

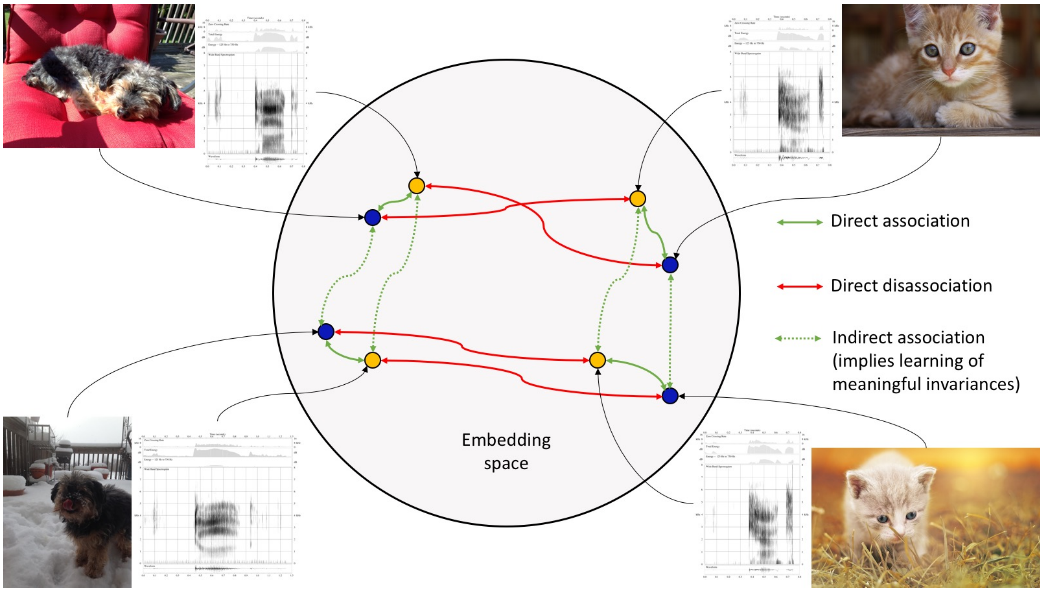

Direct association
Direct disassociation
Indirect association (implies learning of meaningful invariances)
Embedding space

Figure 1-2: Depiction of jointly learning audio-visual embedded representations.

Because the intrinsic nature of the world is multimodal - because as humans we *absolutely rely* on many communicative modes to express ourselves - the next generation of machine learning methodologies will need to treat *all* modalities as first-class citizens. This means moving beyond cascaded, isolated, unimodal system blocks that filter out all information that they cannot individually model. It instead means unified models that draw inferences according to the complex and interdependent relationships of its inputs as well as its context, because there is much to be gained by doing so. Multimodality offers robustness to noise via information *redundancy*; it reduces the sample complexity of learning problems via information *complementarity*; and it enables completely new learning paradigms by allowing parallel modes to co-supervise one another, offering solutions to problems which previously had none.

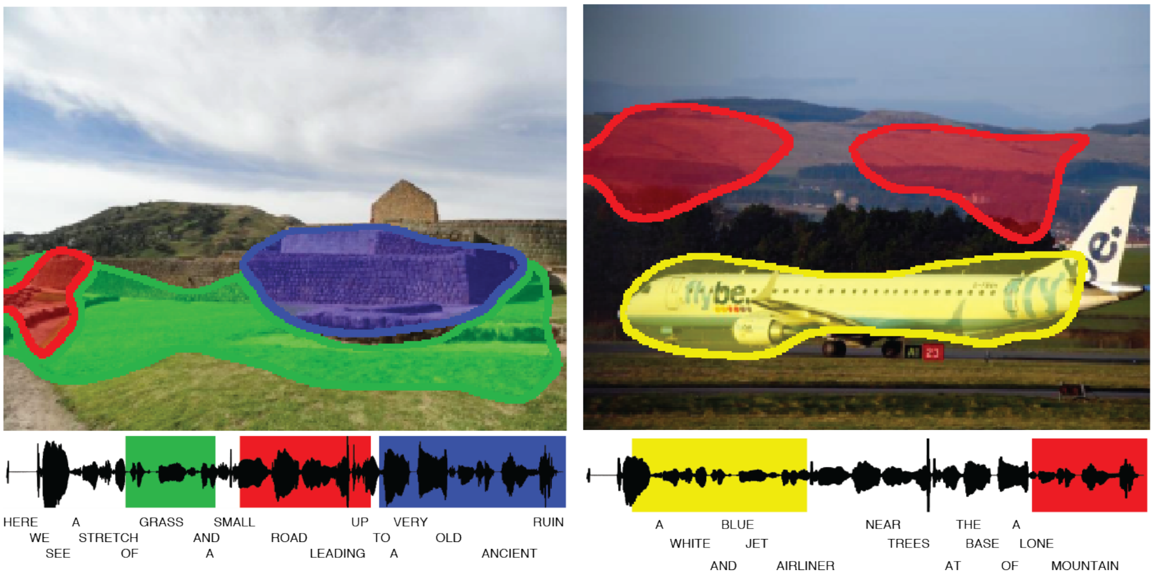


Figure 1-3: Automatically inferred semantic alignments between speech signals and visual images. No supervised speech recognition nor any text transcripts were used in the training of these models; the text is shown solely for the purpose of analysis.

This thesis defines - and offers solutions for - several of these new problems, serving as a small step towards the lofty goal of a holistic learning agent. As a starting point, we consider the joint learning of semantic representations for speech audio and visual images. Our conceptual underpinning is the notion of a multimodal embedding space (Figure 1-2). We employ this idea to construct end-to-end models that directly associate what they *hear* with what they *see*, without the need for conventional speech recognition or text transcripts. These models are not only able to learn to pick out word-like patterns in continuous speech, but also to semantically associate these patterns with objects, colors, and textures in natural image scenes (Figure 1-3). This association allows our models to perform tasks such as semantic image search from spoken queries, without the need for supervised speech recognition.

We go on to demonstrate that our approach is language-agnostic by successfully applying it to Hindi as well as English, and hypothesize that these multimodal correspondence learning techniques could therefore serve as the keystone for building spoken language systems across a multitude of languages that do not rely on expert annotation. Such systems would be a boon for the overwhelming majority of human languages, 98% of which simply do not support supervised speech processing technology due to the lack of expensive, expert-annotated corpora (Lewis et al., 2016; Google, 2018).

Finally, we consider the problem of learning semantic correspondences directly between Hindi and English speech. By learning the Hindi and English words that refer to the same underlying visual objects, we show that our models can leverage the visual domain as an interlingua, enabling them to learn translations between Hindi and English speech without the need for text transcriptions (Figure 1-4). These techniques may be able to serve as a basis for speech translation systems that can accommodate resource-poor languages, and even languages which lack a formal writing system.

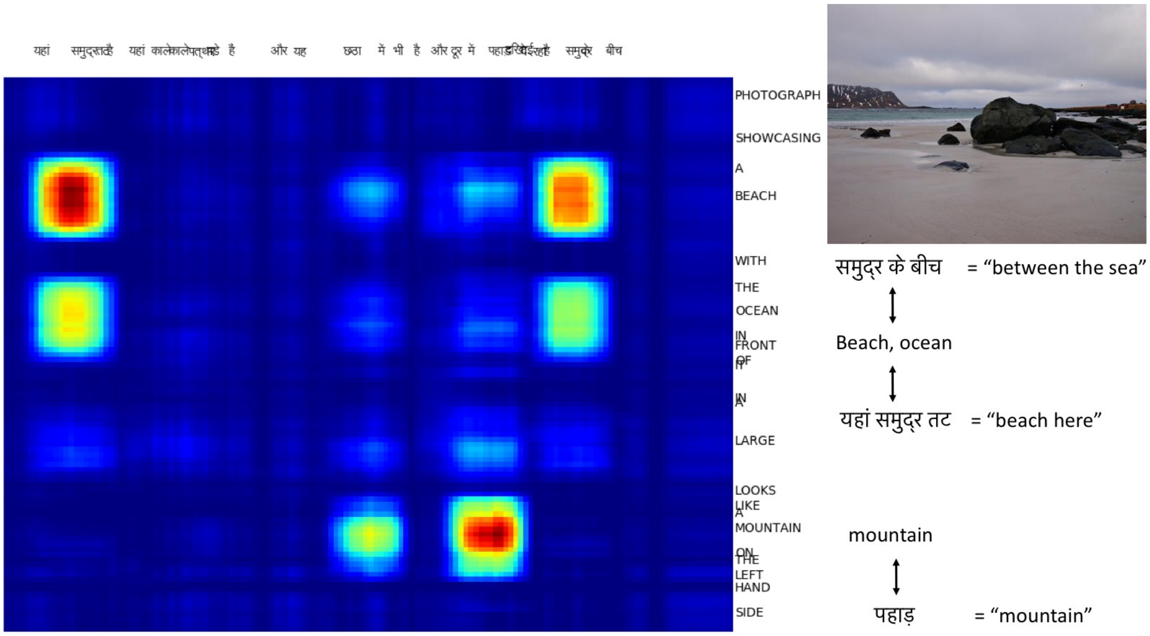


Figure 1-4: Semantic similarity matrix between independent spoken captions in Hindi and in English describing the same underlying image of a beach. Red regions indicate alignments between the speech signals which are inferred by the model to have similar meaning. All similarity scores were computed directly between speech signals with no knowledge of the underlying text (and no conventional speech recognition). The underlying words in Hindi and English are displayed time-aligned along the axes, allowing us to verify that the model is identifying meaningful Hindi-English translations directly at the speech waveform level (Shown at bottom right).

Our society is increasingly flooded with multimodal data streams as smart devices, home automation systems, and mobile/wearable computing devices continue to become more ubiquitous and richer in their sensing capability. This thesis demonstrates that the synergistic modeling of just two modes of communication - speech audio and still-frame natural images - gives rise to compelling new learning problems, as well as their solutions. In doing so, it lights a path towards a future in which these multimodal data streams could be leveraged to build intelligent computing systems that are able to communicate with humans in much more varied, and therefore much richer ways than they do today.

1.2 Contributions of This Thesis

This thesis makes the following specific contributions:

1. **Introduction of models capable of mapping complex visual images and unsegmented, continuous speech into a shared, semantic vector space.** We introduce a more advanced modeling framework based on deep convolutional neural networks that is capable of learning the semantic association between unsegmented images and their spoken captions. We show that these models can embed entire image frames and entire spoken captions as fixed points in a high dimensional, multimodal vector space. In this space, semantic relationships are preserved via vector operations such as the inner product. This enables high-level semantic similarity between image scenes and their captions to be computed via vector operations in the embedding space, which we utilize to perform semantic image search from spoken queries.

2. **Demonstration that the internal representations learned by the models recognize and associate individual words and objects.** We explore two distinct ways of extracting localized segments containing word-like units and object-like image regions: 1) using coupled sliding windows imposed upon the input, and 2) extracting connected components from 3-dimensional spatial-temporal association maps derived from the neural model's internal feature maps. We demonstrate that in both cases, the extracted patterns can be grouped into

very pure clusters using simple algorithms (such as k-means), suggesting that the representations learned by the networks capture a significant amount of high-level linguistic abstraction.

3. **Demonstration of the language-agnostic nature of the models.** Using an additional spoken caption dataset collected in Hindi, we train a set of audio-visual association networks. We show that the caption-to-image (and vice versa) retrieval scores achieved by the Hindi model are close to those achieved with a similarly sized English dataset, suggesting that our approach is indeed language agnostic.

4. **Demonstration of the models' ability to learn cross-lingual semantics.** In addition to training Hindi-language variants of the audio-visual association models originally trained on English, we train a *triplet* model that utilizes a shared image model in conjunction with an English speech model and a Hindi speech model. We demonstrate that such a network can not only perform image/caption retrieval in either language alone, but also can retrieve the Hindi caption associated with the image associated with an English query caption (and vice versa). While the cross-lingual speech-to-speech retrieval scores we achieve are lower than the speech-to-image and image-to-speech scores, they are many times better than chance and suggest a promising new direction for speech-to-speech translation research.

5. **Collection of a very large, multilingual spoken caption dataset.** Over the course of this thesis work, we collected 40,000 read captions for the Flickr 8k dataset (Rashtchian et al., 2010), over 400,000 spontaneous spoken captions for the Places 205 dataset (Zhou et al., 2014), and nearly 10,000 spoken captions for the ADE20k dataset (Zhou et al., 2017) (all in English). We additionally collected nearly 100,000 spoken captions for the Places 205 data in Hindi.

1.3 Chapter Guide

The remaining chapters of this thesis cover the following material:

- Chapter 2 contains a literature review, as well as relevant algorithmic background.

- Chapter 3 Gives an overview of the datasets used throughout this thesis, including a detailed account of how we went about collecting them.

- Chapter 4 introduces the problem of learning an alignment model between visual objects and individual words within a spoken caption. The setting is constrained by the availability of an oracle word segementation and an off-the-shelf visual object detection system.

- Chapter 5 removes the proverbial "training wheels" from the problem setting of Chapter 4 by doing away with the segmentation in both the speech and visual input modalities. In this chapter, models for learning a high dimensional, multimodal vector embedding space are introduced. Arbitrary images and acoustic waveforms can be mapped to points in the embedding space, and vector operations can recover their semantic similarities and differences.

- Chapter 6 describes methodologies for using the models introduced in Chapter 5 to localize visual objects within a larger image frame, as well as individual linguistic units (approximately at the word-level) within a spoken caption. The representations learned for individual object-like and word-like units are shown to be useful not only for clustering together distinct instances of the same underlying word/object, but also for capturing semantic relatedness within and across the modalities.

- Chapter 7 applies the models and methods presented in Chapter 5 to a second language, Hindi. We not only demonstrate the language-agnostic property of our models, but also show that the representations learned can be used infer semantic similarity between English and Hindi speech.

Bibliographic Note: This thesis represents the body of work completed over the course of a Ph.D. program. Therefore, much of the work presented in this thesis has previously appeared in peer-reviewed scientific publications. The content of Chapter 4 was largely published in Harwath and Glass (2015), as was the description of the Flickr8k spoken caption dataset found in Chapter 3. The content relating to the NIPS16 and ACL17 models in Chapters 5 and 6 first appeared in Harwath et al. (2016) and Harwath and Glass (2017), as did the Places English audio caption dataset description found in Chapter 3. The Matchmap model content of those chapters is yet to be published. The content of Chapter 7 as well as the description of the Places Hindi dataset found in Chapter 3 has been accepted to appear in the proceedings of ICASSP 2018.

Chapter 2

Background and Related Work

In this chapter, we describe relevant background material related to acoustic speech signal representations, automatic speech recognition, unsupervised speech processing, artificial neural networks, visual object discovery, machine translation, and multimodal modeling of vision and language.

2.1 Acoustic Signal Representation

All of the models introduced in this thesis function on input data from two modalities: visual images, represented as 2-D arrays of RGB pixels, and audio waveforms of human speech, represented as 1-D arrays of discrete samples. This section details the data pre-processing techniques that we apply to the audio waveforms, before any modeling is performed.

Human speech is produced by a complex process which at its core involves a sound source and an adjustable filter. The sound source derives its energy from airflow out of the lungs, which either creates a periodic signal by vibrating the vocal folds (in the case of voiced speech, such as vowels) or flows freely as turbulent white noise (in the case of unvoiced speech, which includes most consonants). The vocal tract - comprised of the throat, oral cavity, and nasal cavity - forms a cascade of acoustic tubes which filter the signal produced by the sound source. This filter is adjustable in the sense that our muscles can move our tongue, jaw, and lips about, open or close our velum to couple or decouple our nasal cavity to our oral cavity, etc. Depending upon the configuration of these articulators, the acoustic filter formed by our overall vocal tract shape produces perceptually different sounds when excited by a sound source. A full treatment of the acoustic theory of speech production is beyond the scope of this thesis, so we direct interested readers to Stevens's *Acoustic Phonetics* (Stevens, 2000). What is of immediate relevance to us is the fact that the speech signal can be represented as a continuous time waveform $x(t)$, whose properties at a particular moment in time (such as sustained resonances at particular frequencies, moments of silence, quick bursts of wideband energy, etc) are reflective of the underlying speech sound being articulated.

Because digital computers cannot perform computations directly on continuous time signals, we convert $x(t)$ into a digital signal by first sampling its amplitude at discrete time intervals nT, and then quantizing this amplitude, resulting in the discrete sequence $x[n]$. T represents the sampling period, the length of which determines the maximum bandwidth which can be captured by $x[n]$ according to the Nyquist sampling theorem. Most of the salient information in the speech signal is contained in the band below 8 kHz, and so oftentimes a sampling frequency $f_s = \frac{1}{T} = 16$kHz is used (which is the case for all experiments contained in this thesis).

Because a single sample cannot convey enough information by itself to indicate what the vocal tract is doing at a particular point in time, short-time windows of consecutive samples are extracted from $x[n]$ in order to perform a short-time Fourier analysis. In Automatic Speech Recognition (ASR) systems, these windows typically span between 10 and 50 milliseconds in length, with 25 milliseconds being the most commonly used duration. Some overlap is allowed between consecutive windows so as to produce smoothly varying frames - the most common value for 25 ms windows is 15 ms, giving rise to a 10 ms time shift between consecutive windows. Because the speech signal is relatively periodic and stationary at short timescales (such as within the duration of a single frame), mapping each frame into the frequency domain results in a simpler basis for analysis purposes. This operation of windowing the signal followed by applying a Fourier transform to each window individually is known as the

Short Time Fourier Transform (STFT), and is discussed in depth in Oppenheim and Schafer's *Discrete Time Signal Processing* (Oppenheim and Schafer, 2009).

In a practical ASR system, before we apply the STFT, it is advantageous to remove the DC component of the signal,

$$x_0[n] = x[n] - \frac{1}{N}\sum_{n'=0}^{N} x[n'], \tag{2.1}$$

and then to apply pre-emphasis filtering in order to flatten the spectrum, counteracting the lowpass response of the glottal excitation:

$$x_p[n] = x_0[n] - 0.97x_0[n-1] \tag{2.2}$$

After pre-emphasis, the STFT is computed according to:

$$X[m,\omega] = \sum_{n=-\infty}^{\infty} x[n]w[n-mR]e^{-j\omega n} \tag{2.3}$$

where $-\pi \leq \omega \leq \pi$ indexes the frequency axis (with π corresponding to half the sampling rate), the integer variable m indexes the STFT frames, R is the shift between frames (in samples), and w is the window function which performs the selection of the samples to include in the window. An important part of computing the STFT is the particular choice of w. The multiplication of the window h with the signal x in the time domain manifests itself as a convolution in the frequency domain, in effect "smearing" the true spectrum of x with the Fourier representation of h. The shape of h in the time domain determines its shape in the Frequency domain, and so a large number of window functions have been proposed over the years, each with distinct properties and trade-offs (such as frequency resolution vs. the height of the spectral "noise floor" introduced by the window's smearing effect, known as spectral leakage). The most typical window function used in ASR feature extraction front ends is the Hamming window.

$X[m,\omega]$ is a complex-valued signal, capturing information regarding both the magnitude and phase of the frequency components of $x[n]$. It is generally accepted that phase information has little perceptual importance in audition as compared to the actual distribution of energy across the frequency axis. Therefore, we transform the complex spectrum into the power spectrum:

$$X_p[m, \omega] = |X[m, \omega]|^2 \tag{2.4}$$

The power spectrum is then "bucketed" by a set of L triangular, bandpass filters which are nonlinearly spaced along the frequency axis. This warping, known as the Mel scale, reduces the dimension of the power spectrum by grouping together different frequency components which are perceived to be nearly the same by humans (Moore, 1997). The so-called "Mel-frequency spectral coefficients" X_{mfsc} are computed as:

$$X_{mfsc}[m, l] = \sum_{k=-\infty}^{\infty} X_p[m, \omega] V_l[k], \tag{2.5}$$

where V_l denotes the l^{th} mel filter. Finally, the energies contained within each mel filter are converted to the dB scale. We later refer to this representation as log mel-filterbank features:

$$X_{lmf}[m, l] = 10 \log_{10} X_{mfsc}[m, l] \tag{2.6}$$

For ASR systems, X_{lmf} is generally taken one step further by applying the discrete cosine transform (DCT), resulting in mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980). In ASR systems, the MFCCs are typically truncated or "liftered" to the first 13 coefficients. However, for the experiments in this thesis, we take X_{lmf} to represent the speech audio signal. For all experiments in this thesis, we compute 40 MFSCs for every 10 milliseconds of speech audio.

2.2 Automatic Speech Recognition Overview

The methodological underpinning of modern speech recognition systems is the noisy channel model first put forth by Shannon and Weaver (1949). Shannon and Weaver's work focused on mathematically characterizing the information flow through noisy communication networks, but the same theory was later applied to the problem of speech recognition by Fred Jelinek's team at IBM in the 1970s and 80s (Jelinek, 1976). Under the noisy channel model, an acoustic waveform A is heard, and the listener's job is to recover the underlying word sequence $\hat{W}$ which gave rise to A (possibly being corrupted by noise in the process). Inference of $\hat{W}$ given A can be performed using Bayes' rule, resulting in the so-called "fundamental equation of speech recognition":

$$W^* = \operatorname*{argmax}_W P(W|A) = \operatorname*{argmax}_W \frac{P(A|W)P(W)}{P(A)} \tag{2.7}$$

Where W^* represents the recognition hypothesis - the best guess of $\hat{W}$ under the statistical models $P(A|W)$ and $P(W)$. The $P(A)$ term in the denominator is typically ignored, since it does not depend on W.

In ASR jargon, $P(A|W)$ is typically called the *acoustic model* because it tells us how likely an acoustic waveform A is given some word sequence W. If you had a recorded waveform of somebody speaking the word "cat," then a good acoustic model would assign a high likelihood score to the expression $P(A|\texttt{cat})$ and a low score to $P(A|\texttt{dog})$. The second term in the numerator, $P(W)$, is typically called the *language model* because it provides an *a priori* probability of how likely a person is to speak a word sequence W in the first place. Even a very strong acoustic model would have difficulty differentiating between the phrases "how to recognize speech" and "how to wreck a nice beach" due to the fact that the phrases are similar sounding and give rise to similar waveforms. Of course, any English speaker knows that the former phrase simply makes more sense than the latter. This knowledge is encoded by $P(W)$, whose job it is to assign a relatively higher probability to "how to recognize speech" and a lower probability to "how to wreck a nice beach".

Using statistical fitting techniques such as maximum likelihood (ML) to estimate $P(A|W)$ and $P(W)$ requires large collections of audio recordings of people speaking, along with parallel text transcriptions of the speech. The necessary size of the dataset depends on the task at hand. For recognizing read speech from the *Wall Street*

Journal, a standard task in the speech community, the core training set is comprised of approximately 80 hours of transcribed speech (Paul and Baker, 1992), and word error rates (WERs) of below 4% are now possible on this task. This is a relatively restricted task, however: the language is fluent, the audio clean and free of noise, and the vocabulary restricted (a few tens of thousands of words). Real-life speech is full of disfluencies, irregularities, background noise, and a wide-ranging vocabulary. The prevailing strategy for coping with these types of complexities is best captured by Robert Mercer's famous quote, "There's no data like more data," (Jelinek, 2004). In a recent paper, the team responsible for the Google Home speech recognition system (Li et al., 2017) used a core training dataset of 18,000 hours worth of speech audio, with an additional 4,000 hours of adaptation data. This allowed them to achieve a WER of approximately 5% - clearly, a far more difficult task than the *Wall Street Journal*!

There is a third model component that does not appear in Equation 2.7: the *pronunciation lexicon*. Of course, it is possible to model $P(A|W)$ at the word level, estimating a different density for every unique word in the recognizer's vocabulary. Doing so, however, fails to take advantage of one of the remarkable properties of spoken language: namely, that the spoken forms of all words in a language are themselves made up of strings of *phonemes* - elementary acoustic units such as the /r/ in the words *raft* and *car*. These phonemes are used across all words in a language, and a typical language possesses an inventory of merely dozens of different phonemes - as compared to many thousands or even millions of different words in its vocabulary. Modeling $P(A|W)$ where W represents strings of *phonemes* (or more commonly, *phones*; acoustic-phonetic realizations of the more abstracted phonemes) means that the acoustic model needs to estimate only a few dozen densities rather than millions (this is a slight oversimplification, as state-of-the-art recognizers compensate for the co-articulatory effects of neighboring phones by modeling tuples of phones, which brings the number of densities that must be modeled into the few thousands). In order for an acoustic model to function at the level of phones rather than words, a pronunciation dictionary - the lexicon - must provide a mapping between words and their phonetic spellings. This mapping can be represented as a conditional probability distribution $P(S|W)$, where a sequence of words W allocates a probability mass to a sequence of subword units S. Of course, in many cases words only have one or two pronunciations, making $P(S|W)$ a highly constrained distribution. Equation 2.7 can be rewritten to include the lexicon by marginalizing over all subword unit sequences consistent with a word hypothesis W (we also drop the unnecessary $P(A)$ term):

$$W^* = \underset{W}{\operatorname{argmax}} \sum_{S} P(A|S)P(S|W)P(W) \tag{2.8}$$

When it comes time to apply a trained ASR model to the task of recognizing a new speech utterance, the maximization is typically solved using a decoding algorithm such as Viterbi search, combined with an n-best search such as A^* when a set of likely hypotheses is desired (Soong and Huang, 1991). Finite state transducers (FSTs) are often employed to represent the search graph, because they enable each building block of the recognizer (acoustic model, lexicon, language model) to be specified individually, independent of one another (Pereira et al., 1994). The FST composition operation allows these individual FSTs to be combined into a composite FST search graph on the fly. This makes it very straightforward to unplug certain system blocks from a recognizer pipeline and replace them with a different block - for example, an ASR system designed to handle weather-related queries might use a language model that places a large amount of probability mass on weather-related words. In an FST framework, this recognizer could easily be adapted to a different domain, such as restaurant search, by replacing the language model FST with a restaurant-specific language model, and then re-composing the search graph.

Now we turn our attention to the specific types of statistical models which are typically employed in ASR systems. Given an acoustic waveform represented as a finite series of feature vectors (such as the previously described MFCCs or MFCSs) $A = x_1, x_2, \ldots, x_T$ and a subword unit sequence $S = s_1, s_2, \ldots, s_N$, $P(A|S)$ is generally modeled using a Hidden Markov Model (HMM) (Baker, 1975). Each subword unit is modeled by its own small HMM, typically consisting of 3 states to model the beginning, middle, and end of the unit. These individual HMMs are then concatenated to represent the unit sequence S. The acoustic frames are treated as the state emissions (or observations), and their densities are usually represented by either a set of Gaussian Mixture Models (GMM) (Bilmes, 1998) or a Deep Neural Network (DNN) (Mohamed et al., 2012). A front-end acoustic feature extraction scheme, such as the one detailed in Section 2.1, is necessary to first convert a recorded waveform to the vector series A.

A straightforward way to represent $P(S|W)$ is with a table of weighted key-value pairs, in which each key (a word, in this case) maps to a set of different values (the phone or subword unit sequence that spells the word), each with an associated weight or probability. For example, consider the trivial ARPABET lexicon below:

```
cat 1.0 /k/ /ae/ /t/
dog 1.0 /d/ /ao/ /g/
tomato 0.7 /t/ /ah/ /m/ /ey/ /t/ /ow/
tomato 0.3 /t/ /ah/ /m/ /aa/ /t/ /ow/
```

The lexicon is generally handcrafted by a linguist. Some alternative approaches are to use a graphemic lexicon (Killer et al., 2003) or to learn a set of pronunciations in a data-driven fashion (Badr et al., 2011; McGraw et al., 2013)

$P(W)$ is typically represented with a count-based n-gram model. Under this model, the probability of a word sequence is factorized into the product of the probabilities of each individual word in the sentence, each conditioned on all the words which appeared before it; in other words,

$$P(w_1, w_2, w_3, \dots, w_k) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_k|w_1, w_2, \dots, w_{k-1})$$

In practice, to cope with the combinatorially large number of possible word histories, an n-gram model truncates the history and considers only the previous $n-1$ words. For example, a bigram model would make the assumption that

$$P(w_1, w_2, w_3, \dots, w_k) = P(w_1)P(w_2|w_1)P(w_3|w_2) \dots P(w_k|w_{k-1})$$

Estimating n-gram probabilities is accomplished via simple counting, i.e.

$$P(w_2|w_1) = \frac{C_{w_1,w_2}}{C_{w_1}},$$

where C_{w_1,w_2} represents the number of times the two word sequence w_1, w_2 appears in the data, and C_{w_1} represents the number of times that w_1 appears in the data overall. Of course, raw count based estimation has several undesirable properties, such as zero probabilities assigned to unseen n-grams. To compensate for this, smoothing techniques are applied to the empirical count based distribution, with Kneser-Ney smoothing (Kneser and Ney, 1995) being a popular, time-tested technique. More recently, recurrent neural network (RNN) language models have been shown to outperform n-gram language models in a variety of tasks, including speech recognition (Schwenk and Gauvain, 2005). However, due to the difficult nature of incorporating them into the ASR decoder, a first pass recognition is usually performed with an n-gram model to produce an N-best list of recognition hypotheses (or a decoding lattice), which is then rescored with an RNN LM.

The preceding summary does not detail the great number of algorithmic bells and whistles that are necessary to achieve cutting edge performance in a real-world system; for the reader interested in an overview of those techniques, we recommend Yu and Deng's *Automatic Speech Recognition: A Deep Learning Approach* (Yu and Deng, 2014) or Huang et al's *Spoken Language Processing* (Huang et al., 2001) as further reading.

2.3 Unsupervised Speech Processing

While conventional ASR systems have a long history and have recently made great strides thanks to the revival of deep neural networks, their reliance on highly supervised training paradigms has essentially restricted their application to the major languages of the world, accounting for a small fraction of the more than 7,000 human languages spoken worldwide (Lewis et al., 2016). The main reason for this limitation is the fact that these supervised approaches require enormous amounts of very expensive human transcripts. Moreover, the use of the written word is a convenient but limiting convention, since there are many oral languages which do not even employ a writing system. In contrast, infants learn to communicate verbally before they are capable of reading and writing - so there is no inherent reason why spoken language systems need to be inseparably tied to text.

The completely supervised learning scenario described above is at one end of a spectrum defined by Glass (2012), shown in Figure 2-1. As the spectrum is traced from left to right, less and less supervision and annotation is available. Early speech recognition systems required very high levels of annotation, including time-aligned phonetic transcriptions of the speech, phonetic pronunciation dictionaries, etc. A respectable amount of effort has been made to fill in the "data-based" region of the spectrum, employing techniques such as grapheme-based acoustic models (Killer et al., 2003) or automatic sub-word unit learning (Rabiner et al., 1989). Recent end-to-end deep neural network (DNN) models for speech recognition such as the Listen, Attend, and Spell (LAS) sequence-to-sequence (seq2seq) model (Chan et al., 2016) or the EESEN model based on connectionist temporal classification (CTC) (Miao et al., 2015) also operate within this realm. These networks directly map the acoustic signal to letters or words, allowing the network's learned internal representations and recurrent structure to absorb the roles traditionally held by the lexicon and HMM acoustic models. Impressive as these results are, the data-based models are still unable to do away with the albatross of parallel text transcriptions. The most vexing challenge to the dominant, supervised ASR paradigm is the observation that nearly every human who has ever lived developed the ability to carry a simple conversation by the time they were approximately two years old. This is an enormously complex task; an infant must acquire a phoneme inventory and word lexicon, learn to segment continuous speech signals by identifying word boundaries, learn syntactic and grammatical structure, and learn how language semantically relates to the world at large (Jusczyk, 1997). Studies in developmental psychology have presented evidence that human infants do not learn to perform this multitude of abilities one-by-one in a sequential nature, but

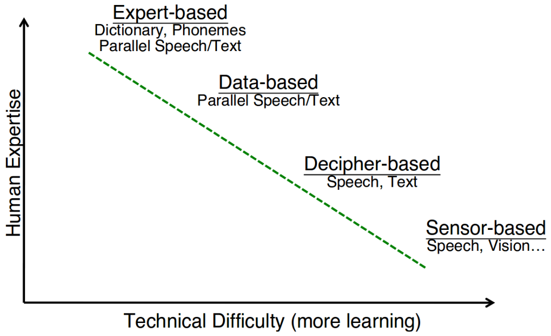


Figure 2-1: ASR learning scenarios as defined by Glass (2012).

rather simultaneously and gradually (Dupoux, 2016). This learning doesn't come to a halt once a child has learned to talk - the inherent ability to learn new words on the fly and even acquire additional languages stays with us for our entire lives (Bloomfield, 1933; Chomsky, 1986).

While the neurological mechanisms underlying human language abilities are not yet understood, the ability to learn so much with so little (in the way of annotation, transcription, etc. employed in computer ASR systems) can be seen as a "proof of concept" of a far more efficient, organic, and robust language learning framework than the current state-of-the-art ASR pipeline - represented by the lower-right end of the spectrum shown in Figure 2-1. This proof of concept - human beings - has served as the inspiration for many researchers in the field of spoken language processing, who have developed a growing body of work utilizing unsupervised or weakly/distantly supervised machine learning algorithms for speech processing (sometimes called "zero resource" speech processing). Much of the interest in this topic was sparked by Park and Glass' pioneering work on segmental dynamic time warping (S-DTW) (Park and Glass, 2005, 2008), an algorithm which is able to discover clusters of repeated speech patterns, typically corresponding to word-like units, directly from the acoustic waveform without any form of transcription. Many subsequent efforts extended these ideas (Jansen et al., 2010; Jansen and Van Durme, 2011; Dredze et al., 2010; Harwath et al., 2012; Zhang and Glass, 2009). Alternative approaches based on Bayesian nonparametric modeling (Lee and Glass, 2012; Ondel et al., 2016; Kamper et al.,

2016) employed a generative model to cluster acoustic segments into phoneme-like or word-like categories, and related works aimed to segment and cluster either reference or learned phoneme-like tokens into higher-level units (Johnson, 2008; Goldwater et al., 2009; Lee et al., 2015).

Although neural networks have dramatically pushed the state of the art forward in supervised ASR applications, their application in the unsupervised (or zero-resource) speech processing community has been more modest. Generally, they have been used to learn acoustic feature representations which better capture the underlying linguistic content and are (hopefully) more invariant to sources of undesirable variation, such as noise, speaker or channel characteristics, etc. Notable examples of this are Zhang et al. (2012); Renshaw et al. (2015); Kamper et al. (2015); Thiolliere et al. (2015). However, to the best of our knowledge there has so far been no other application of deep neural networks to directly discover high-level lingusitic units (such as at the word level) from within continuous speech at the raw signal level. This could be in part due to the difficult nature of defining a suitable neural network architecture and objective function for the task. Even the neural approaches for learning unsupervised frame-level representations rely upon targets derived from an S-DTW based pattern discovery step (Kamper et al., 2015).

2.4 Artificial Neural Networks

Although neural networks as a class of models have a very long history (McCulloch and Pitts, 1943), they fell out of favor with the advent of statistical machine learning models (Bishop, 2011) and kernel methods such as SVMs (Cortes and Vapnik, 1995). The basic formulation of a single hidden layer neural network is surprisingly simple. Given an input vector x, a parameter matrix A and bias vector b, a suitable nonlinear, scalar function $f()$, and an output vector $\hat{y}$, the relationship between x and $\hat{y}$ is expressed as:

$$\hat{y} = f(Ax + b) \tag{2.9}$$

Historically, the most iconic choice of $f()$ has been the sigmoid function:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.10}$$

Many other nonlinearities (or "activation functions") have been proposed, the most popular of which nowadays is arguably the rectified linear unit (ReLU):

$$f(z) = \max(0, z) \tag{2.11}$$

Training a neural network is framed as a numerical optimization problem, in which a dataset of (x, y) pairs is given. The task is to find the specific values of A and b which result in the predicted value $\hat{y} = NN(x)$ (where $NN()$ represents application of the network to input x) being "close" to the true value y according to a loss function $loss(y, \hat{y})$. Standard tasks in machine learning, such as classification and regression, are easily realized with this scheme. For example, a vector regression network can be trained when y is a real-valued, continuous vector and $loss(y, \hat{y}) = \frac{1}{2}||y - \hat{y}||^2$. Classification is most often performed by using a special "softmax" layer immediately before the output:

$$\hat{y}_i = \frac{\exp(v_i)}{\sum_j \exp(v_j)} \tag{2.12}$$

where v represents the output vector of the final hidden layer of the network. For classification, the most common loss function and label representation is to choose y as a 1-hot vector of length equal to the number of classes (where the element representing the target class is 1 and all other elements are 0) and use the cross entropy loss:

$$loss(y, \hat{y}) = \sum_i y_i \log(\hat{y}_i) \tag{2.13}$$

Gradient descent methods are the de-facto family of optimization algorithms used in training DNNs, with the backpropagation algorithm (Liannainmaa, 1970; Werbos, 1982; Rumelhart et al., 1986) providing an efficient way to calculate the gradient of the loss function with respect to the network parameters, given a particular set of

(x, y) pairs. When the set of (x, y) pairs at each update step is a randomly sampled subset of the full training set, the optimization is dubbed stochastic gradient descent (SGD) (Bottou, 2010).

Between 2006 and 2012, a special class of neural networks began to gain popularity. These networks came to be known as deep neural networks (DNNs), taking their name from the fact that they are formed by stacking multiple single layered neural networks on top of one another. For example, a two hidden layer network can be expressed as:

$$y = f(A_2 f(A_1 x + b_1) + b_2) \tag{2.14}$$

Of course, there is no limit to the number of layers which can be consecutively stacked, and many current architectures utilize far more than two layers! Arguably, one factor that prevented DNNs from finding widespread use before this time is the fact that training them is, in general, a challenging, unstable, and poorly understood optimization problem. Among the insights that are credited with the renewal of interest in DNNs is the layer-wise pretraining scheme introduced by Hinton et al. (2006). Although the layer-by-layer generative pretraining step is no longer widely used, it shed light on how some of the pitfalls which tend to befall DNNs can be avoided in practice. One example of this is the "vanishing gradient" (or the related "dying ReLU") problem, which arises when the weight vector associated with a particular neuron ("neuron" referring to a single dimension of the output of a particular hidden layer in the network) takes on a value for which nearly all inputs x (at least those seen in the training data) result in a very large or very small value, so that the neuron's nonlinearity becomes "stuck" very deeply in its saturation region. At this point, the output of the neuron is nearly flat, and thus the gradient of the output with respect to the neuron's weight vector (as well as its input) is close to zero. The subtext of Hinton et al. (2006) is that the initial setting of the weight vector of every neuron of the network must be set such that the expected value of the neuron's activation is outside the saturation region of the activation function. Today, this is often accomplished with careful parameter initialization (Glorot and Bengio, 2010) rather than generative layer-wise pretraining. The initialization issue is only one of many DNN training difficulties that have been addressed recently, others including faster training via massively parallel hardware architectures such as GPUs, and very large datasets such as Deng et al. (2009).

The basic fully-connected, feedforward network architecture described above is a flexible and general purpose model. However, a large number of more specialized architectures exist which are well-suited for certain problem domains. Two well-known variants are recurrent neural networks (RNNs) and convolutional neural networks (CNNs) (LeCun et al., 1998). In this thesis, we make extensive use of convolutional networks, which we describe here. Recall that in the case of a fully connected network, the i^{th} row of the A matrix for some given layer, a_i (and the corresponding element of its b vector, b_i), represents the weight vector of the i^{th} neuron in that layer. The output of that neuron is simply $f(a_i^T x + b_i)$ - the inner product of the weight vector with the layer's input x, offset by the bias b_i. Every single element of x is taken into account in computing the single scalar output of the neuron. In certain cases, the input data may be known to have special properties which can be exploited. Consider the case in which the input to a neural network is a natural image of size H pixels tall by W pixels wide, with D color channels (typically 3 in the case of an RGB image). We can write this input as a 3rd order tensor $x \in \mathcal{R}^{H \times W \times D}$ Intuitively, most images will obey the following two properties:

1. Individual pixels within images are often highly correlated with their neighboring pixels. Most objects, backgrounds, etc. appearing in natural images are comprised of only a few colors. Furthermore, discontinuities in the image space still exhibit local structure - for example, boundaries between objects tend to be locally linear.

2. Patterns in images are often translation invariant. Panning the image frame horizontally or vertically does not change the content of the image (unless it moves outside of the frame), only its position within the frame.

One way to take advantage of these properties is to restrict the input dependence of each neuron within a layer to a small patch of neighboring pixels (addressing property 1), and to re-apply the neuron to a large set of patches across the image (addressing property 2). Mathematically, this is expressed as a 2-dimensional convolution of a kernel image (the neuron's weight matrix) across the vertical and horizontal dimensions of the layer's input. When D_{out} such convolutional kernels are applied in parallel to the same input spanning D_{in} channels, they are said to form a convolutional layer producing an output feature map $v \in \mathcal{R}^{H_c \times W_c \times D}$. The exact values of H_c and W_c depend on how many kernel patches can fit within the bounds of the input for a given stride. It is common (though not without exception) to use a stride of 1 (consecutive applications of the kernel are spaced 1 pixel apart) and to add zero padding around the border of the input so that $H_c = H$ and $W_c = W$. In its basic form, a 2nd order convolution of D_{out} kernels each of height H_k by width W_k is expressed as follows:

$$v[h, w, d_{out}] = f\left(b_{d_{out}} + \sum_{d_{in}=0}^{D_{in}} \sum_{h_k=0}^{H_k-1} \sum_{w_k=0}^{W_k-1} C_{d_{out}}[h_k, w_k, d_{in}] x[s_h * h + h_k, s_w * w + w_k, d]\right) \tag{2.15}$$

where $C_{d_{out}}$ represents the weight matrix of the d_{out}^{th} kernel, $b_{d_{out}}$ is the bias of the d_{out}^{th} kernel, s_h is the vertical stride, s_w the horizontal stride, $f()$ represents the nonlinear activation function of the neuron. Because images contain many pixels, the raw data space is of very high dimension and thus difficult to model directly. For this reason, convolutions are usually applied in an alternating fashion with a form of spatial downsampling. Max pooling is the most commonly used variant, which uses a spatially strided window (with height H_k, width W_k, vertical stride s_h, and horizontal stride s_w) to extract the maximum activation of each channel within that window:

$$v_{pool}[h, w, d] = \max_{h_k=0}^{H_k-1} \max_{w_k=0}^{W_k-1} v[s_h * h + h_k, s_w * w + w_k, d] \tag{2.16}$$

Mean (or average) pooling is another variant, which replaces the within-window maximum with the within-window average. For a deeper discussion of neural network architectures, we direct interested readers to Goodfellow et al. (2016).

2.5 Visual Object Recognition and Discovery

Classification of visual objects (or other patterns) is a longstanding problem within the computer vision community, with the MNIST handwritten digit task being a classic and widely known example (LeCun et al., 1998). Recent progress in the field has been driven in part by recurring challenge competitions such as ISLVRC Russakovsky et al. (2015). Since 2012, the task has been dominated by deep convolutional neural networks (CNNs), popularized by Krizhevsky et al. (2012). Since that time, improved variants of the basic CNN architecture have continued to push the state of the art (Simonyan and Zisserman, 2014; He et al., 2015). While classification asks the question of "what", object detection and localization (also part of the ISLVRC suite of tasks) address the problem of "where". Generally these localization systems are trained using handcrafted bounding box annotations for the training data (Girshick et al., 2013; Redmon et al., 2016), however other works investigate weakly-supervised or unsupervised object localization (Bergamo et al., 2014; Cho et al., 2015; Zhou et al., 2015; Cinbis et al., 2016).

A large body of research has also focused on unsupervised visual object discovery, in which case there is no labeled training dataset available. One of the first works within this realm is Weber et al. (2010), which utilized an iterative clustering and classification algorithm to discover object categories. Further works borrowed ideas from textual topic models (Russell et al., 2006), assuming that certain sets of objects generally appear together in the same image scene. More recently, CNNs have been adapted to this task (Doersch et al., 2015; Guérin et al., 2017), for example by learning to associate image patches which commonly appear adjacent to one another.

2.6 Vision and Language

Multimodal modeling of images and text has been an extremely popular pursuit in the machine learning field during the past decade, with many approaches focusing on accurately annotating objects and regions within images. For example, Barnard et al. (2003) relied on pre-segmented and labelled images to estimate joint distributions over words and objects, while Socher and Li (2010) learned a latent meaning space covering images and words learned on non-parallel data. While these approaches focused on improving the identification of visual objects from a pool of predefined classes, other research has studied the problem of aligning text to the images or videos they describe. For example, Kong et al. (2014) took visual scenes with high level captions, parsed the text, detected visual objects, and then aligned the two modalities with a Markov random field. Lin et al. (2014) aligned semantic graphs over text queries to relational graphs over objects in videos to perform natural language video search. Matuszek et al. (2012) employed separate classifiers over text and visual objects that shared the same label sets.

A related problem is that of natural language caption generation. While a large number of papers have been published on this subject, recent efforts using deep neural networks (Karpathy and Li, 2015; Vinyals et al., 2015; Fang et al., 2015) have made tremendous progress and generated much interest in the field. In Karpathy and Li (2015), Karpathy uses a refined version of the alignment model presented in Karpathy et al. (2014) combined with an off-the-shelf RCNN object detection network (Girshick et al., 2013) to produce training exemplars for a caption-generating RNN language model that can be conditioned on visual features. Through the alignment process, a semantic embedding space containing both images and words is learned. Other works have also attempted to learn multimodal semantic embedding spaces, such as Frome et al. (2013) who trained separate deep neural networks for language modeling as well as visual object classification. They then embedded the object classes into a dense word vector space with the neural network language model, and fine-tuned the visual object network to predict the embedding vectors of the words corresponding to the object classes. New problems within the intersection of language and vision continue to be introduced, such as object discovery via multimodal dialog (de Vries et al., 2017), visual question answering (Antol et al., 2015), and text-to-image generation (Reed et al., 2016).

2.7 Machine translation

The final chapter of this thesis studies multilingual audio-visual models, and explores how the visual space can act as an interlingua for cross-lingual speech retrieval - highlighting potential connections to machine translation (MT). Machine translation is a well-established problem, and with the advent of neural models is currently undergoing a revolution comparable to those in speech recognition, computer vision, and natural language processing at large. At first dominated by statistical methods combining count-based translation and language models (Koehn et al., 2013), the current state-of-the-art paradigm relies upon neural sequence-to-sequence with attention models (Bahdanau et al., 2015) operating on dense lexical representations ("word vectors") (Mikolov et al., 2013; Pennington et al., 2014). However, new ideas continue to be introduced, including models which take advantage of shared visual context (Specia et al., 2016). The motivation behind these ideas has much in common with our own - that is, representations of language in machine learning models can be enriched with information from the visual domain. However, these approaches still operate on text data, whereas we propose to build models that can be applied directly to untranscribed acoustic waveforms.

Speech-to-speech translation has been a longstanding dream for researchers, world travelers, and the international business community. Current state-of-the-art approaches at their core rely on text-to-text translation models, with a speech-to-text preprocessing step and a text-to-speech postprocessing step (for example, Microsoft Translator). Recently, Weiss et al. (2017) published an effort to move beyond that paradigm, and achieved remarkable results in implementing translation between speech audio in the source language and written text in the target language. Weiss' model is completely end-to-end, and does not require the speech recognition preprocessing step; however, it still relies upon expert-crafted text transcriptions of the translations of the source speech into the target language, and would still require a text to speech postprocessing module in order to be capable of speech-to-speech translation. The approach that we propose would not require this translation and transcription, instead relying on the visual space to provide a shared anchor between speech audio in both the source and target languages.

2.8 Relation of Prior Work to this Thesis

Although all of the works cited above are in one way or another relevant to our own, the research contained within this thesis represents an entirely new direction that is distinct from any previously published work. While the speech and vision communities have both studied unsupervised pattern discovery within their respective modalities, these tasks have never been performed jointly together as we do in this thesis. Leveraging multiple modalities in this way allows us to perform speech pattern discovery far more efficiently than ever before. While current state-of-the-art, speech-only algorithms such as S-DTW run in $O(N^2)$ time, our model training and grounding can be completed in $O(N)$ time. We demonstrate this in Chapter 6 by performing pattern discovery on over 522 hours of audio data, by far the largest published speech pattern discovery experiment to date.

Not only do our methods scale linearly, but the representations learned are richer than unimodal representations by virtue of capturing cross-modal semantics. The visual semantics in turn serve as a bridge to learn lexical semantics across discovered patterns, exemplified by the fact that the pattern clusters corresponding to “lake” and “pond” neighbor one another in the embedding space because they tend to be used to describe similar visual patterns (see Chapter 6). Another byproduct of our learning procedure is that the representations learned by intermediate layers of our networks can themselves serve as frame-level acoustic representations for other tasks; Drexler and Glass (2017) showed that these representations are competitive with (and in several cases outperform) other popular unsupervised acoustic modeling approaches.

Finally, in Chapter 7, we present the first ever successful results for unsupervised cross-lingual semantic speech retrieval. We believe that the methods we introduce could be adapted for machine translation, potentially opening the door for MT systems that do not require directly parallel corpora, perhaps even in a direct speech-to-speech context.

This chapter has discussed relevant background work in automatic speech recognition, unsupervised speech processing, artificial neural networks, vision and language, visual object discovery, and machine translation. In the next chapter, we give an account of the datasets used throughout the remainder of this thesis.

Chapter 3

Datasets and Data Collection

In this chapter, we describe the datasets that will be used throughout this thesis. We describe in detail how we went about collecting data, and analyze the overall properties of our datasets. Specifically, we introduce four datasets of spoken audio captions. The Flickr8k audio caption dataset is a small-scale pilot dataset based on the previously published Flickr8k image corpus (Rashtchian et al., 2010) which we use to train proof-of-concept multimodal modals in Chapter 4. The Places English audio caption dataset is a much larger-scale dataset which forms the basis of Chapters 5 and 6. It is based upon the previously published Places 205 (Zhou et al., 2014) image datset. The ADE20k audio caption dataset is a second small-scale dataset that we collect, based upon the ADE20k image dataset (Zhou et al., 2017). These images contain dense pixel-level object annotations, allowing us to perform a fine-grained analysis of our models. Finally, we discuss the Places Hindi audio caption dataset, also based upon the Places 205 images, which we utilize to train cross-lingual models in Chapter 7.

3.1 The Flickr8k Audio Caption Dataset

3.1.1 Image Captioning Overview

Our first efforts towards data collection were inspired by contemporary research work being done on natural language image captioning. What makes image captioning datasets compelling is the fact that they make use of natural language, rather than contrived annotations of image regions using a fixed, closed set of label categories. These seminal works on neural image captioning (Karpathy and Li, 2015; Vinyals et al., 2015) utilized a number of datasets which contain images alongside human-generated text captions, such as Pascal, Flickr8k (Rashtchian et al., 2010), Flickr30k (Young et al., 2014), and MSCOCO (Lin et al., 2015). However, all of these datasets include text captions only, and no speech audio. Because of its manageable size and ubiquitousness in the previous literature, we choose to use the Flickr8k as the starting point for our data collection, soliciting human subjects to record themselves reading the Flickr8k captions out loud.

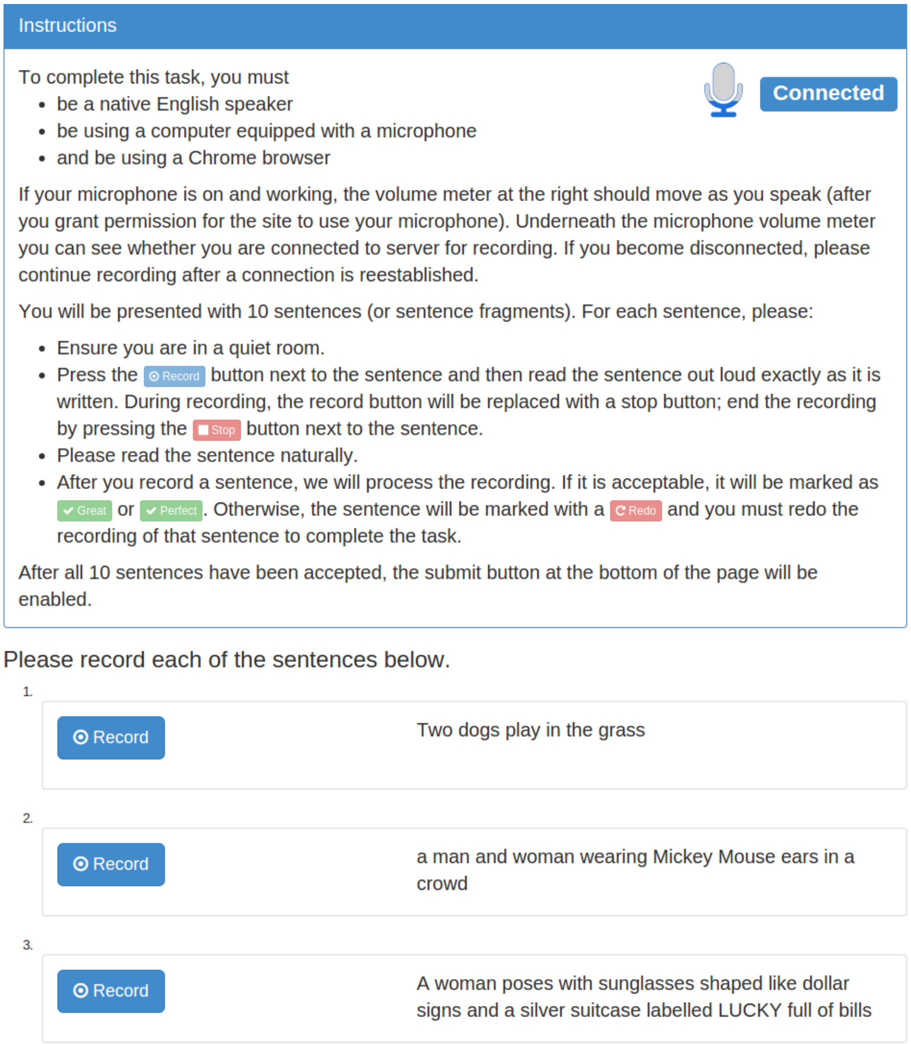
Collecting read speech in this manner offers two advantages. First, speech recognition can be used as an automatic quality control mechanism. When embedded in a computerized collection interface, it can provide instantaneous feedback to a user when their speech was too noisy, corrupted, or did not match the target caption text. Second, the text captions can function as a ground truth transcription for each spoken caption, enabling more meaningful analysis to be performed on the speech audio.

3.1.2 Collection of Read Captions via Amazon Mechanical Turk

Flickr8k contains approximately 8,000 images captured from the Flickr photo sharing website, each of which depicts actions involving people or animals. Each image was annotated with a text caption by five different people, resulting in a total of 40,000 captions. To collect these captions, Rashtchian et al. (2010) turned to Amazon's Mechanical Turk (AMT), an online service which allows requesters to post "Human

Intelligence Tasks" (HITs). These HITs are then made available to anonymous, non-expert workers, or "Turkers", who can choose to complete the tasks for a small amount of money. We utilized AMT to collect spoken audio recordings for each of the 40,000 captions from the Flickr8k dataset. We use the Spoke JavaScript framework (Saylor, 2015, Available at https://github.com/psaylor/spoke) as the basis of our audio collection HIT. Spoke is a flexible framework for creating speech-enabled websites, acting as a wrapper around the HTML5 getUserMedia API while also supporting streaming audio from the client to a backend server via the Socket.io library. The Spoke client-side framework also includes an interface to Google's SpeechRecognition service, which can be used to provide near-instantaneous feedback to the Turker.

Figure 3-1 displays a screenshot of the audio collection interface we used in our HITs. A set of 10 random captions are displayed to the user, who can click the start/stop button to record their speech while they read each caption out loud. A playback button allows the Turker to listen to their own recordings and diagnose any problems with their microphone or environment. Spoke pipes the audio to the Google recognizer, checks the recognition result against the prompt, and notifies the user if their speech could not be recognized accurately. The Turker is then given the option to re-record the errorful caption. The HIT cannot be submitted until all 10 captions have been successfully recorded. During collection, we utilized a very simple metric for verification - 60% or more of the caption words must appear in the recognition result, regardless of ordering. We found this to be both lenient and sufficient - users rarely complained about the system correctly recognizing their speech, and 95.7% of the collected utterances were easily aligned to their caption text using our Kaldi (Povey et al., 2011) forced alignment system. The majority of the utterances flagged as unalignable were either empty or cut short, which we believe may have been due to client-server connection issues; the problematic utterances were recollected by another round of HITs. We paid the Turkers 0.5 cents per spoken caption, resulting in at total cost of just over $200 including Amazon's service fee. We collected speech from 183 unique Turkers, with the average worker completing 218 captions. There were a handful of Turkers who completed far more than the average number of captions,



Instructions

To complete this task, you must

- be a native English speaker
- be using a computer equipped with a microphone
- and be using a Chrome browser

Connected

If your microphone is on and working, the volume meter at the right should move as you speak (after you grant permission for the site to use your microphone). Underneath the microphone volume meter you can see whether you are connected to server for recording. If you become disconnected, please continue recording after a connection is reestablished.

You will be presented with 10 sentences (or sentence fragments). For each sentence, please:

- Ensure you are in a quiet room.
- Press the Record button next to the sentence and then read the sentence out loud exactly as it is written. During recording, the record button will be replaced with a stop button; end the recording by pressing the Stop button next to the sentence.
- Please read the sentence naturally.
- After you record a sentence, we will process the recording. If it is acceptable, it will be marked as ✓ Great or ✓ Perfect. Otherwise, the sentence will be marked with a Redo and you must redo the recording of that sentence to complete the task.

After all 10 sentences have been accepted, the submit button at the bottom of the page will be enabled.

Please record each of the sentences below.

1. Record — Two dogs play in the grass
2. Record — a man and woman wearing Mickey Mouse ears in a crowd
3. Record — A woman poses with sunglasses shaped like dollar signs and a silver suitcase labelled LUCKY full of bills

Figure 3-1: Audio collection interface for capturing spoken captions on Amazon Mechanical Turk.

with the highest number collected from a single worker being 2,978.

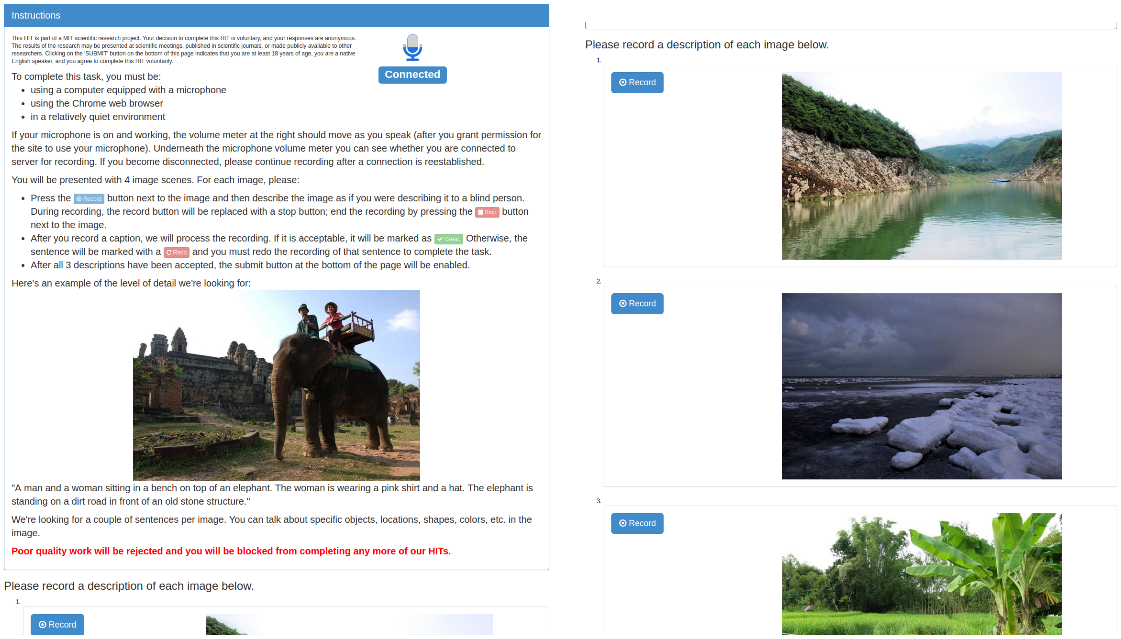
To further verify the integrity of our collected audio data, we split the 40,000 utterances into a 30,000 utterance training set, a 5,000 utterance development set, and a 5,000 utterance testing set, covering a 8,918 word vocabulary. Our splits correspond with the training, validation, and testing splits given by Rashtchian et al. (2010). We then used Kaldi to build a large vocabulary speech recognition system, adapting the standard Wall Street Journal recipe for a GMM/HMM + LDA + MLLT + SAT system for our data. We employed the training set to train the acoustic and language models, the CMU pronunciation lexicon, and the development set to tune the acoustic and language model weights. The final word error rate of our system on the test set was 11.67%, providing another indication that our data is relatively high quality. In order to preprocess the Flickr8k data for our CNN, we employ this recognizer to force align the audio to the ground truth text transcripts and segment the audio at the word level.

3.2 The Places 205 English Audio Caption Corpus

The experiments detailed in Chapter 4 that make use of the Flickr8k Audio Caption data demonstrate the feasibility of learning visually grounded representations of speech audio. However, they also highlight the fact that even with 40,000 captions, there is a very significant gap between the recall scores that can be achieved by using the speech audio models (0.179 image R@10, and 0.243 caption R@10) and models which utilize the textual representations of the captions (0.49.0 image R@10, and 0.567 caption R@10). This suggests that, not surprisingly, learning from speech audio is more difficult than learning from text, and so a much larger dataset of audio captions is needed in order to close this performance gap.

With this in mind, we set about collecting a second, much larger dataset of spoken image captions. Collecting captions for an existing dataset much larger than Flickr8k, such as MSCOCO, was a possibility we considered. However, MSCOCO has several drawbacks. Its captions are relatively short, averaging approximately 10 words per caption. Additionally, the number of object categories (80) is rather limited. Finally, we made a strategic decision to collect open-ended, spontaneous spoken captions instead of having users read text captions aloud. There are significant prosodic differences between read speech and spontaneous speech (Howell and Kadi-Hanifi, 1991), and we wanted to challenge our models to cope with the more realistic domain of spontaneous speech.

We ultimately decided that the Places 205 dataset (Zhou et al., 2014) would serve our purposes. Places205 contains over 2.5 million images categorized into 205 widely varied scene classes, such as beaches, stadiums, grocery stores, bedrooms, and city streets. This variety, combined with the scene-level focus of the images, provides an enormously rich taxonomy of visual object types appearing in many different contexts. We also hoped that the combination of rich visual scenes and free-form caption collection would provide us with much longer spoken captions compared to datasets like Flickr and MSCOCO. We were not disappointed, as on average the Places captions we collected contained 21 words - twice as many as MSCOCO.



Instructions

This HIT is part of a MIT scientific research project. Your decision to complete this HIT is voluntary, and your responses are anonymous. The results of the research may be presented at scientific meetings, published in scientific journals, or made publicly available to other researchers. Clicking on the 'SUBMIT' button on the bottom of this page indicates that you are at least 18 years of age, you are a native English speaker, and you agree to complete this HIT voluntarily.

To complete this task, you must be:

- using a computer equipped with a microphone
- using the Chrome web browser
- in a relatively quiet environment

If your microphone is on and working, the volume meter at the right should move as you speak (after you grant permission for the site to use your microphone). Underneath the microphone volume meter you can see whether you are connected to server for recording. If you become disconnected, please continue recording after a connection is reestablished.

You will be presented with 4 image scenes. For each image, please:

- Press the Record button next to the image and then describe the image as if you were describing it to a blind person. During recording, the record button will be replaced with a stop button; end the recording by pressing the Stop button next to the image.
- After you record a caption, we will process the recording. If it is acceptable, it will be marked as ✓ Great. Otherwise, the sentence will be marked with a Redo and you must redo the recording of that sentence to complete the task.
- After all 3 descriptions have been accepted, the submit button at the bottom of the page will be enabled.

Here's an example of the level of detail we're looking for:

"A man and a woman sitting in a bench on top of an elephant. The woman is wearing a pink shirt and a hat. The elephant is standing on a dirt road in front of an old stone structure."

We're looking for a couple of sentences per image. You can talk about specific objects, locations, shapes, colors, etc. in the image.

Poor quality work will be rejected and you will be blocked from completing any more of our HITs.

Please record a description of each image below.

1.

Please record a description of each image below.

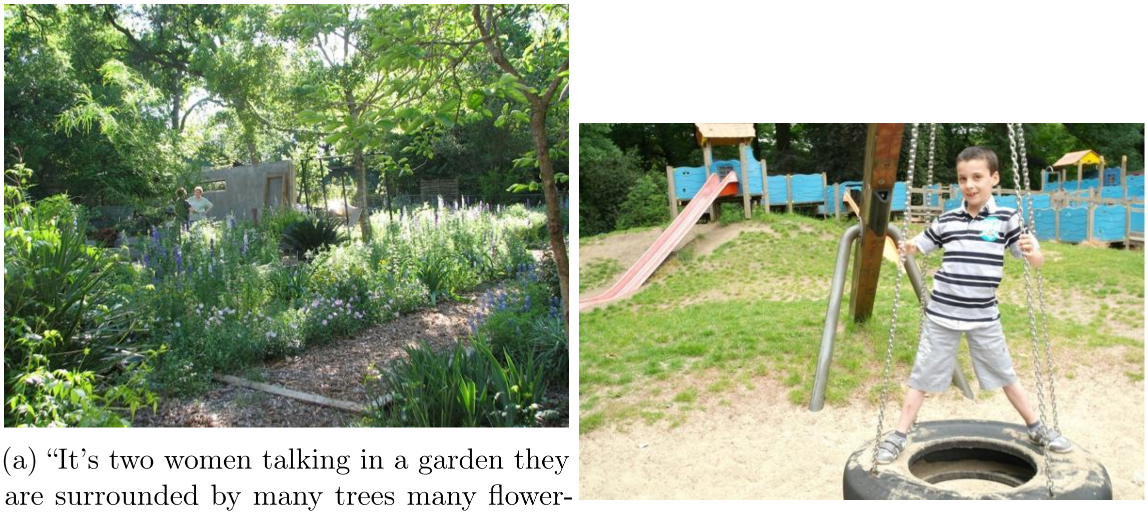
1.

2.

3.

Figure 3-2: Screenshots of the Places English audio caption collection interface.

To collect audio captions, we again turned to Amazon's Mechanical Turk service. We used a modified version of the Flickr8k audio collection interface based on the Spoke JavaScript framework (Saylor, 2015, Available at https://github.com/psaylor/spoke) as the basis of our HITs. Instead of displaying text to a Turker and asking them to read it aloud into their microphone, four randomly selected images are shown to the user, and a start/stop record button is paired with each image. The user is instructed to record a free-form spoken caption for each image, focusing on describing the salient objects in the scene. The backend sends the audio off to the Google speech recognition service, which returns a text hypothesis of the words spoken. Because we do not have a ground truth transcription to check against, we use the number of recognized words as a means of quality control. If the Google recognizer was able to recognize a minimum number of words (8 to 12 worked well for us), we accept the caption. If not, the Turker is notified in real-time that their caption cannot be accepted, and is given the option to re-record their caption. Each HIT cannot be submitted until all 4 captions have been successfully recorded. Two screenshots of our collection interface are displayed in Figure 3-2. Turkers were paid $0.03 per caption. At the time of writing, we have collected approximately 415,000 captions, equally sampled across the 205 different scene categories from the Places 205 dataset. Approximately 10,000 of these captions were drawn from the ADE20K dataset (Zhou et al., 2017). These images are drawn from the same 205 Places scenes, but are also accompanied by object segmentation and annotation masks. These annotations enable richer experimental analysis to be performed. To give the reader a concrete idea of what our data looks like, two example image/caption pairs are shown in Figure 3-3.



(a) "It's two women talking in a garden they are surrounded by many trees many flowering perennials and there's a pathway in the garden."

(b) "Young boy standing on a tire swing he's wearing a black and white striped shirt."

Figure 3-3: Two example Places images and the text transcripts of their associated captions.

3.2.1 Dataset Statistics and Analysis

We analyzed some basic properties of the Places audio caption data, which we present here. The Places 205 image scene database has a relatively even balance across all 205 categories, and we were careful to maintain this property when collecting audio captions. Figure 3-4 displays the number of captions collected for each scene category. The overwhelming majority of scene categories have approximately 2,000 audio captions each, with the scene category posessing the smallest number of captions still totaling at over 1,500.

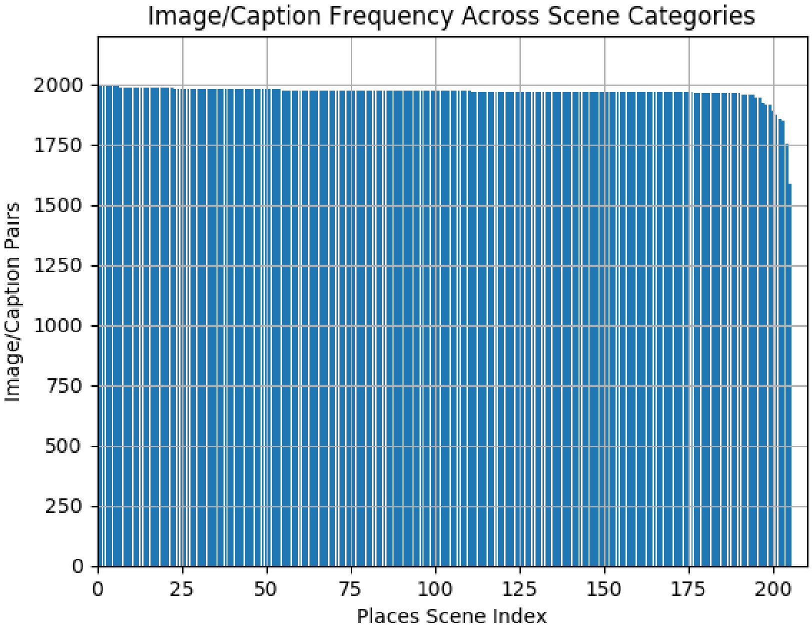
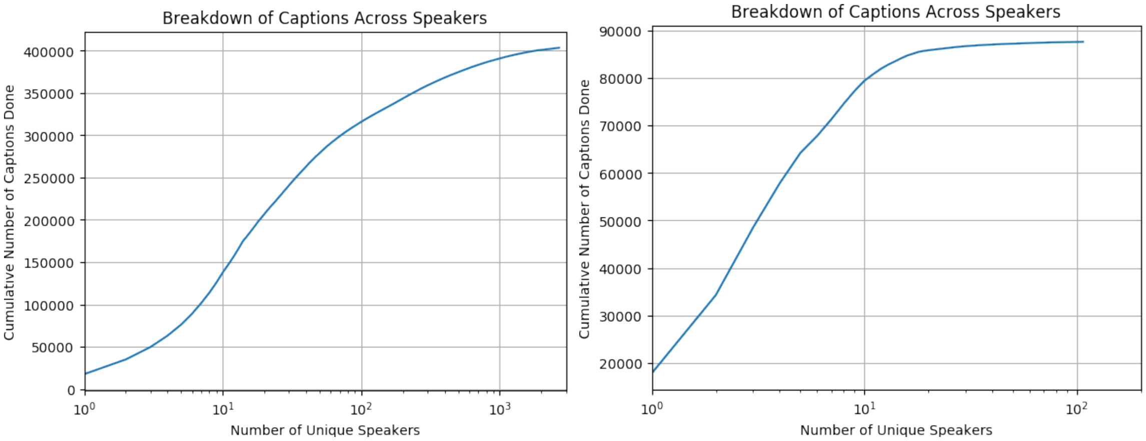


Figure 3-4: Number of English audio captions collected for each Places 205 scene category

We next analyzed the breakdown of the captions across unique speakers. While we do not have a ground truth speaker identity for each caption, Amazon Mechanical Turk logs a unique string tag for each worker, allowing us to deduce which captions were submitted by the same worker. We make the reasonable assumption that each worker constitutes a unique speaker (it is possible the multiple users would share the same account, but anecdotal listening to a sampling of the captions indicated to us that this is not likely to be common), and then we total the number of captions completed by each worker. Figure 3-5a displays these totals in the form of a cumulative distribution of captions across speakers. We can see that while we collected speech from approximately 3,000 unique Turkers, the overwhelming majority of the captions were recorded by a small fraction of those Turkers; the 100 most prolific workers account for approximately 80% of the captions, and the 10 most prolific workers



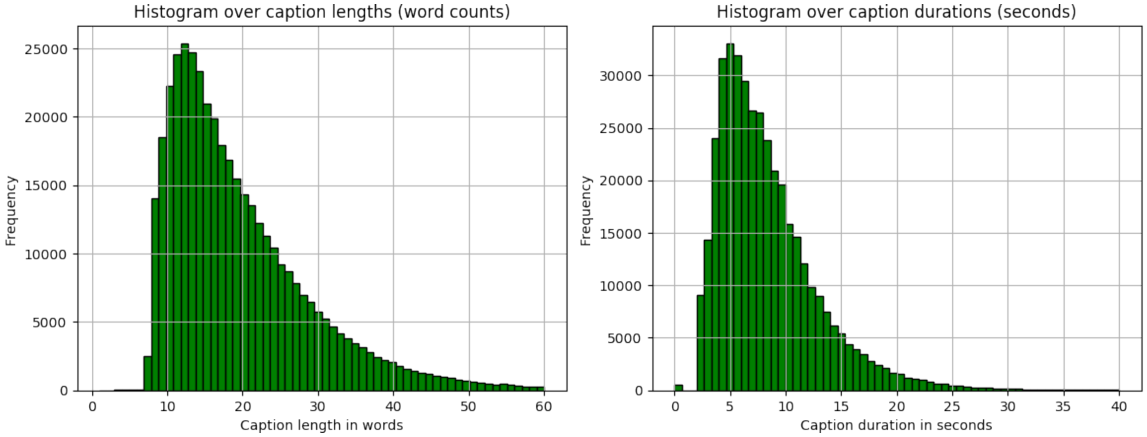
(a) English

(b) Hindi

Figure 3-5: Breakdown of audio captions recorded across speakers for the Places English and Hindi data. For each (x, y) point on the curves, y represents the sum total number of captions completed by the x most prolific speakers.

account for approximately one-third of the captions. While in an ideal world we would prefer a more uniform distribution across speakers, the tendency for a majority of HITs to be completed by a small group of “power users” is a widely-known property of the Mechanical Turk platform (Adda and Mariani, 2010; Fort et al., 2011). Imposing a cap on the maximum number of HITs a single user can complete therefore makes collecting large datasets far less feasible.

We also examined the lexical properties of the captions in several ways. Overall, the vocabulary size of the entire caption dataset was found to be 43,953 unique words. Figure 3-6 displays a histogram of the number of words per caption, as estimated by the Google ASR transcriptions. The mode of the distribution is around 12 words, however there are a long tail of captions with significantly longer lengths, which pulls the average number of words per caption to approximately 20. Because our experiments in subsequent chapters do not leverage knowledge of these transcriptions but require truncation or padding of the waveforms themselves to a uniform size (for computational efficiency reasons inherent to training neural networks on existing GPU hardware with existing toolkits), we are also interested in assessing the distribution over caption durations. A marginal histogram over caption durations in seconds is displayed in Figure 3-6. Nearly all captions are under 20 seconds in duration, while the majority are under 10 seconds. The practical effects of truncation to these durations will be explored later in Chapter 5.


Histogram over caption lengths (word counts)

Histogram over caption durations (seconds)

Figure 3-6: Histograms over Places English caption lengths in words and durations in seconds.

3.2.2 Spoken Captions for the ADE20k Dataset

The ADE20k (Zhou et al., 2017) dataset is comprised of approximately 20,000 image scenes accompanied by dense (pixel-level) object annotation. Approximately 10,000 of the images belong to the same scene categories found in the Places 205 dataset; the remainder are drawn from a broader set of scene labels. We utilized our AMT interface to collect spoken captions for the 10,000 images belonging to the Places 205 categories in order to form a dataset with both word-level (in the case of the speech) and object level (in the case of the images) annotation for the purpose of pattern analysis. These experiments are detailed in Chapter 6.

3.3 Multilingual Extensions to the Places 205 Audio Caption Corpus

A central claim of this thesis is that our models are language-agnostic, and should therefore work equally well on non-English languages. To provide evidence in support of this claim, we ported our collection interface to additional languages including

Hindi, Spanish, and Arabic. At the time of writing, the Spanish and Arabic caption collection efforts are still in their infancy; however, we have collected over 100,000 Hindi captions. In initial attempts at soliciting captions from Hindi-speaking Turkers, we translated the whole of the HIT instructions into Hindi and posted them on AMT. However, we found that very few workers completed these tasks, possibly because Hindi-speaking Turkers are accustomed to using the AMT website (which includes searching for HITs to complete) in English. Once we changed the instructions for the HIT back to English, but included additional verbage instructing the Turkers to speak their captions in Hindi, we found that workers started to complete the HITs at a much more rapid pace.

We performed a similar analysis of the Hindi data as we did for the English data, with histograms over utterance lengths in words and durations in seconds shown in Figure 3-7, and a breakdown of the captions across speakers shown in Figure 3-5b. We see similar trends with respect to caption lengths and durations, although the Hindi caption vocabulary size is considerably smaller at 19,226 unique words. This is not unexpected, as we have collected over four times as many English captions as Hindi captions. Notably, the breakdown of captions across Hindi Turkers is even more skewed, with 10 Turkers responsible for contributing the vast majority of the captions.

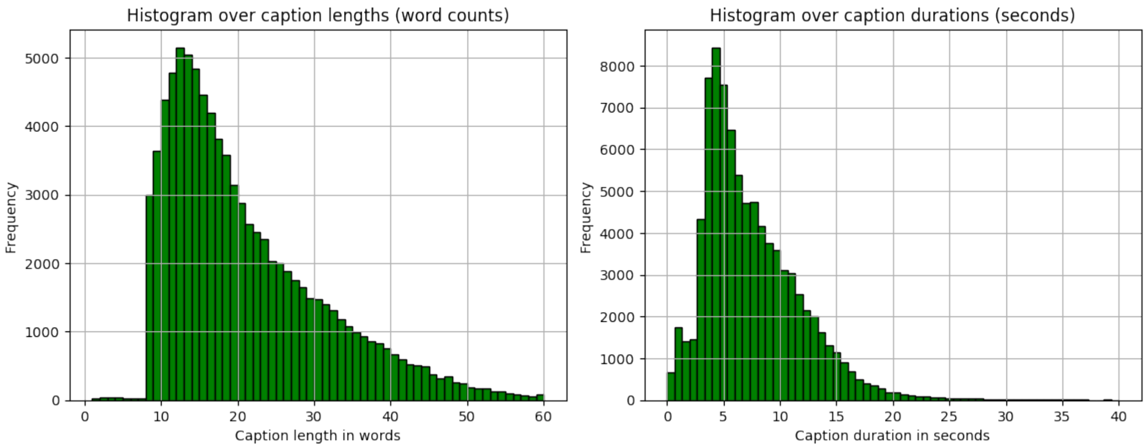


Figure 3-7: Histograms over Places Hindi caption lengths in words and durations in seconds.

3.4 Chapter Summary

In this chapter, we provided an overview of the datasets studied throughout this thesis. Specifically, we presented our English audio caption corpora to accompany the Flickr8k, Places 205, and ADE20k datasets, and a Hindi audio caption corpora for the Places 205 dataset. In the next chapter, we present our first exploratory efforts into joint modeling of vision and speech audio.

Chapter 4

Grounding Speech to Images: The Pre-Segmented Case

In Chapter 3, we described in detail the datasets used throughout this thesis. This chapter describes our primordial foray into joint audio-visual modeling - a “proof-of-concept” using the Flickr8k audio caption dataset. We first enumerate the modeling assumptions we make, and then present an audio-visual alignment model that makes use of pre-segmented inputs. The model is trained to discriminate which image is described by a specific caption, and vice versa. We finish with experimental results for image and caption retrieval, and also provide some visualizations. These experimental results were promising enough for us to embark on a much more ambitious quest to collect data, which culminated in the Places English, Hindi, and ADE20k English datasets. These datasets, as well as the lessons learned from this chapter, form the foundation for the models at the heart of this thesis, found in Chapter 5.

The contents of this chapter were first published in Harwath and Glass (2015).

4.1 Problem setting

Conventional automatic speech recognition (ASR) systems utilize training data in the form of speech audio with parallel text transcriptions. The text transcriptions provide an extremely strong supervisory signal that enables the recognizer to learn which invariant features of the acoustic signal distinguish one word from another. Of course, transcriptions are expensive to create, as well as restrictive - an ASR system cannot recognize a word that is out of its vocabulary. We are interested in investigating to what degree it is possible to replace those transcriptions with contextually relevant *images.* Such data could conceivably be far easier to collect, while still offering enough constraint for a model to learn to recognize words. This training paradigm also opens the door for richer linguistic representations to be learned by endowing words with visual semantics.

Given a dataset comprised of image scenes with accompanying spoken audio captions, we propose the first of several models developed in this thesis that are capable of learning to associate images with their spoken descriptions. This model relies on a pair of convolutional neural networks (CNNs), one for images and another for speech, along with an alignment and embedding model. The outputs of the networks provide fixed-dimensional representations of variable-sized visual regions and spoken words, which are then mapped into a shared semantic embedding space. This allows us to align the words in the captions to the objects and regions they refer to in the image scene. This is an exceptionally difficult problem, and thus we make several simplifying assumptions:

1. The spectrograms of the spoken audio captions are pre-segmented at the word level via forced alignment to a ground truth transcription using a supervised ASR system.

2. A pre-trained neural network capable of providing embedding vectors of spectrogram segments is also available. This network was pre-trained to perform isolated word classification on a separate speech corpus, but here we remove the final classification layer.

3. A pre-trained Region-Convolutional Neural Network (R-CNN) (Girshick et al., 2013) based upon the VGG architecture (Simonyan and Zisserman, 2014) trained on the ILSVRC12 (Deng et al., 2009) corpus is available to produce object proposal regions. This network is also able to produce embedding vectors

capturing the visual semantics of those regions; however, its final classification layer has also been removed.

We will revisit these assumptions in Chapter 5 and onwards, where they will be removed or largely weakened.

4.2 Model Description

Our goal here is to be able to represent examples of spoken words, alongside examples of visual objects, as points in a high dimensional vector space. For example, in this vector space, we would like different spoken examples of the word “dog” to neighbor one another, and also to neighbor image crops containing dogs. In order to do this, we require some means to transform variable sized image crops as well as variable duration audio waveforms into fixed dimensional vector representations. Further, we also require some way of coaxing these vectors into taking on the property that semantically similar images and words neighbor one another. To achieve this, we employ two separate neural network architectures, one for images and one for audio, which we then marry together with an embedding alignment model.

4.2.1 Region Convolutional Neural Networks

In order to detect a set of candidate regions in an image which are likely to contain meaningful objects, we use the Region Convolutional Neural Network (RCNN) model (Girshick et al., 2013). The RCNN object detector works by first using selective search (Uijlings et al., 2013) to build a large list of proposal regions, typically numbering in the thousands for a given image. Each proposal region is then fed into a CNN object classifier, which is used to extract the activations of the penultimate layer of neurons in the network. These activations form a fixed-dimensional (4096 in Girshick et al. (2013), as well as our work) feature vector representation of each proposal region. A set of one-versus-all support vector machines are then used to calculate detection scores over some set of classes for each region, and highly overlapping regions with similar classification scores are merged. Finally, the remaining set of regions can be ranked in order of their maximum classification score across all classes. In our work, we follow Karpathy and Li (2015) and take the top 19 detected regions along with the entire image frame, resulting in 20 regions per image. We use the $d_I = 4096$ dimensional RCNN feature vectors to represent each region, which we will refer to as $\mathcal{V} = \{v_i | i = 1 \ldots 20\}$

4.2.2 Spectrogram Convolutional Neural Network

Previous efforts (Karpathy and Li, 2015; Vinyals et al., 2015) to perform semantic alignment of text to objects in image scenes have benefited from the fact that text is naturally segmented into words, and all instances of the same word share the same orthography. On the other hand, segmenting continuous speech into words is nontrivial, and different spoken instances of the same underlying word will inevitably differ in not only their duration, but also in their acoustic feature representations as influenced by factors such as the microphone and speaker characteristics and the context in which the word was spoken.

While a speech recognition system is a reasonable solution for building a spoken interface for natural language image retrieval systems such as the one described in Karpathy and Li (2015), in this chapter we are more interested in investigating the potential of neural networks to learn meaningful semantic representations which operate directly on the feature level. However, tasking our system with also performing word segmentation on the audio stream significantly complicates the problem at hand. We choose to take a step back from the text-based framework by pre-segmenting each spoken caption into a sequence of audio waveforms, each containing a single ground-truth word, and then throwing away the word identity of each segment.

In Bengio and Heigold (2014), the authors trained a CNN isolated word recognizer and utilized it for N-best recognition hypothesis re-ranking; here, we propose to use a similar CNN to model the spectrogram of each isolated word in the image captions. Standard CNNs expect their inputs to be of a fixed size, so in order to accommodate our variable duration words we follow Bengio and Heigold (2014) and choose to embed their spectrograms in a fixed duration window, applying zero-padding and truncation when necessary. While Bengio and Heigold (2014) found that a 2 second window was sufficient to capture the duration of 97% of the words in their corpus, in our case a 1 second long window is long enough to capture 99.9% of the words appearing in the dataset used for the experiments in this chapter.

To create the spectrogram representing each word, we begin by performing forced-alignment of the audio to its ground truth text transcription in order to determine word boundary information. Next, we apply a standard 25 millisecond window with a 10 millisecond shift to each word utterance, extracting log energy filterbank features for each window using 40 filterbanks spaced along the Mel scale, as described in Section 2.1. Finally, we either pad with zeros or truncate equally on both sides to force the spectrogram to have a width of 100 frames, or 1 second. Figure 4-1 shows an example of what the input data to the network looks like for an instance of the word "strategists". From this point onwards, we treat our spectrograms as 40 pixel-tall by 100 pixel-wide grayscale images.

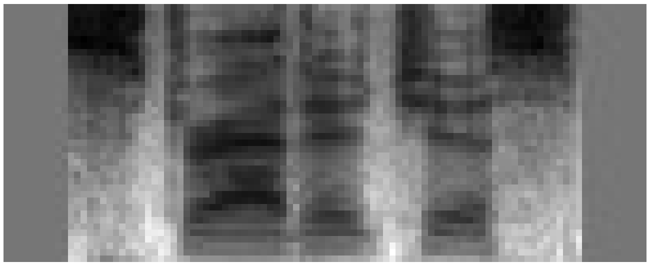


Figure 4-1: Log mel filterbank spectrogram of the word "strategists".

We rely on the Caffe (Jia et al., 2014) toolkit to train our networks and extract the word spectrogram features. Our CNN architecture is as follows:

1. Pixel-by-pixel mean image spectrogram subtraction, with the mean spectrogram estimated over the entire training set;
2. Convolutional layer with filters sized 5 frames by 40 features with a stride of 1, vertical padding of 1 pixel on both the top and bottom, and 64 output channels with a ReLU nonlinearity;
3. Local response normalization of width 5, $\alpha = 0.0001$, and $\beta = 0.75$;

4. Max pooling layer of height 3, width 4, vertical stride 1, and horizontal stride 2;
5. Two fully connected layers of 1024 units each, with a dropout ratio of 0.5 and ReLU nonlinearities;
6. A softmax classification layer

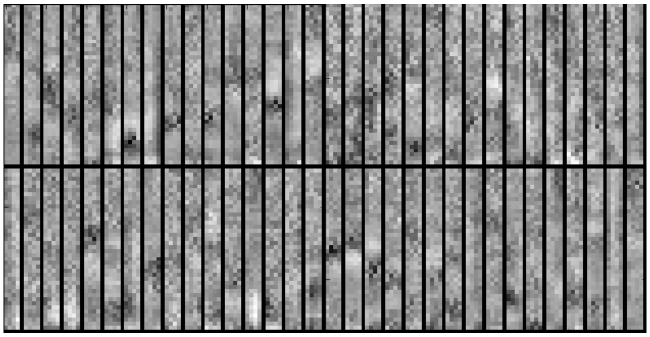


Figure 4-2: 64 learned filters for the spectrogram CNN.

To extract vector representations for each word in some image caption, we feed the word's spectrogram through the network and discard the softmax outputs, retaining only the activations of the $d_W = 1024$ dimensional fully connected layer immediately before the classification layer. For a given caption, we will refer to these vectors as $\mathcal{W} = \{w_j | j \dots N_w\}$, where N_w is the number of words appearing in the caption.

4.2.3 Embedding Alignment Model

Given an image-caption pair and their corresponding object detection boxes and word spectrograms, our task is to align each word with one of the detection boxes found in the image. Note that these detection boxes are far from error-free, but we use them anyway with the hopes that the most salient objects in each image will be captured by a few of the detection boxes. Additionally, this model makes the assumption that each word is independently associated with one (and only one) detection box in the image. Obviously, this reflects an impoverished model of language which limits what can be learned. Nevertheless, we hope to at least capture linkages between salient objects and the words which reference them. To perform the matching, we adopt the transform model from Karpathy et al. (2014) but with the objective function presented by Karpathy and Li (2015). However, we replace the text modelling side of Karpathy's models with our word spectrogram CNN, enabling us to align the image fragments directly to segments of speech audio. We provide a brief overview of the alignment model and objective here.

Let $\mathcal{V} = \{v_i | i = 1 \ldots 20\}$ be the set of d_I-dimensional vectors representing the activations of the penultimate layer of the RCNN for each detected image region. Also let $\mathcal{W} = \{w_j | j \ldots N_w\}$ be the d_W-dimensional vectors representing the similar activations of the spectrogram CNN on each of the N_w words in the spoken caption. The job of the alignment model is to map all of the $v \in \mathcal{V}$ and $w \in \mathcal{W}$ vectors into a shared, h-dimensional space where semantically related words and images have a high similarity.

The alignment model is two-faceted, with separate transforms applied to the image vectors as well as the word spectrogram vectors. We use an affine transform, $y = W_{image}v + b_{image}$ to map an image vector v into the h-dimensional semantic embedding space. To map a word spectrogram vector w into that same embedding space, we use a nonlinear transform, $x = f(W_{audio}w + b_{audio})$ where $f(z)$ is some element-wise nonlinear function. For the experiments in this chapter, we set $f(z) = \max(0, z)$.

Motivated by the assumption that the spoken caption l for a given image k should contain words which directly reference objects in the image, Karpathy's objective function tries to assign a high similarity to matching image-caption pairs by "grounding" each word vector to one or more image fragment vectors. The inner product similarity between a given word embedding and an image fragment embedding is used to measure the degree of grounding, and each word in caption l is given a score according to its maximum similarity across all image fragments from image k. An overall image-caption similarity score is then computed by summing the scores of all words in the caption, thresholded below at 0:

$$S_{kl} = \sum_{t \in g_l} \max_{i \in g_k}(0, y_i^T x_t), \tag{4.1}$$

where g_l denotes the set of image fragments in image l, and g_k is the set of word spectrograms in caption k.

In Karpathy and Li (2015), the authors use a max margin objective function which forces matching image-caption pairs to have a higher similarity score than mismatched pairs, by a margin. Given that S_{kk} denotes the similarity between a matching image-sentence pair, the cost is defined as:

$$\begin{aligned}\mathcal{C}(\theta) = \sum_k \Big[\sum_l & \max(0, S_{kl} - S_{kk} + 1) \\ + \sum_l & \max(0, S_{lk} - S_{kk} + 1) \Big].\end{aligned} \tag{4.2}$$

In practice, we use stochastic gradient descent to optimize this cost function in terms of the parameters $\theta = \{W_m, b_m, W_d, b_d\}$. Figure 4-3 illustrates our full model.

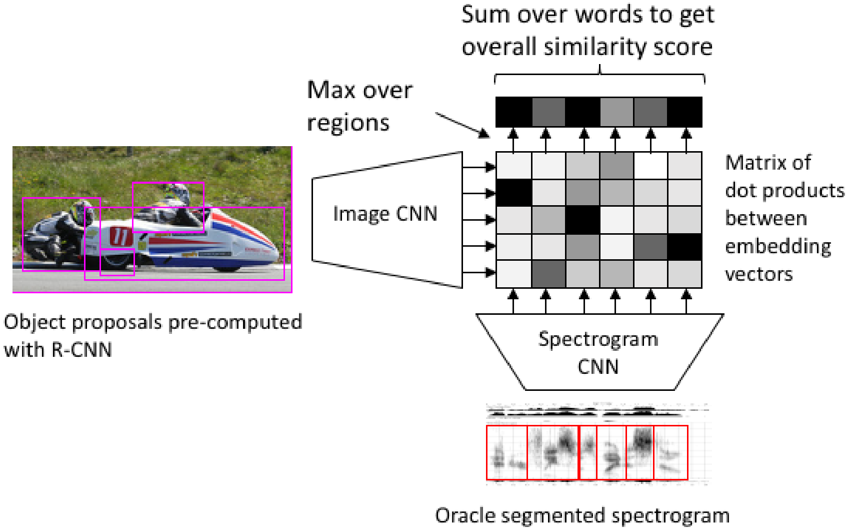

Sum over words to get overall similarity score
Max over regions
Image CNN
Matrix of dot products between embedding vectors
Object proposals pre-computed with R-CNN
Spectrogram CNN
Oracle segmented spectrogram

Figure 4-3: Illustration of the audio-visual alignment model. The inputs are pre-segmented, and the similarity between each unique pair of (image crop, spectrogram segment) is reflected in the matrix of dot products. The overall similarity score between an image and a caption is computed by taking the column-wise maximum over the dot product matrix (i.e. over image regions) and then summing the resulting vector (i.e. over the spectrogram segments)

4.3 Experimental Data

The experiments in this chapter make use of the 40,000 spoken captions collected for the Flickr8k Audio Caption corpus, described previously in Chapter 3. Because the

Flickr8k corpus contains a small number of images and captions relative to datasets such as Imagenet (Deng et al., 2009), we follow the example of Karpathy and Li (2015) and use the off-the-shelf RCNN provided by Girshick et al. (2013) trained on ImageNet to extract the 4096-dimensional visual object embeddings. Similarly, we employ supervised pretraining for the word spectrogram CNN using the Wall Street Journal SI-284 split (Paul and Baker, 1992). This set contains roughly 82 hours of speech, from which we extracted all instances of words occuring at least 10 times in the data. This gave us a total of 612,108 words covering a vocabulary of size 6,010, which we split 80/20 into training and testing sets. We used this data to train our word spectrogram CNN using the 6,010 word vocabulary as our output targets. Even though this training is supervised, 6,749 of the unique words appearing in the Flickr8k transcriptions (75% of the vocabulary) do not appear in the training set for the spectrogram CNN.

4.4 Experiments

We use stochastic gradient descent with a learning rate of 1e-6 and momentum of 0.9 across batches of 40 images to train our embedding and alignment model, and run our training for 20 epochs. Training is performed using the standard 6,000 image train set from the Flickr8k data, using the accompanying 30,000 captions. At each batch, we randomly choose to use only one of the five captions associated with each image. We tried several different settings for h, the dimension of the semantic embedding space, and found that values between 512 and 1024 seemed to work well, in line with Karpathy et al. (2014). We also found that it was necessary to normalize the w vectors to unit magnitude in order to prevent exploding gradients.

To evaluate the alignment and embedding model, we follow the example of Karpathy and Li (2015); Karpathy et al. (2014); Socher et al. (2014) and use our model to perform image retrieval and annotation. Image search is defined as choosing a caption from the test set and then asking the system to find which image belongs with the caption. Image annotation is the opposite problem: choosing an image from the test set without its caption, and then asking the system to search over all the captions in the test set and find one of the five which belongs with the image. We report recall@10 as our evaluation metric, or the probability that the correct result is found in the top 10 returned hits. Table 4.1 details the results of our system ("Spectrogram CNN"), as well as a comparison to replacing the word spectrogram embeddings with 200-dimensional word vectors taken from Huang et al. (2012). We also compare to Socher et al. (2014) and Karpathy et al. (2014). While our text + word vector system outperforms Karpathy et al. (2014), the model is more similar to the refinements made in Karpathy and Li (2015) but with a single layer word embedding network rather than a bidirectional recurrent neural network. Karpathy and Li (2015) reports high recalls on the Flickr30k data (50.5 search and 61.4 annotation), but does not include any results on the Flickr8k data. Although our spectrogram CNN does not perform nearly as well as any of the systems with access to the ground truth text, it massively outperforms a random ranking scheme. This is in spite of the fact that not only does the spectrogram CNN system not have direct access to the ground truth word identity of the caption words, but also that the CNN word embedding vectors are of dimension 1024 rather than 200. We believe that these results are quite promising, and with more training data we expect to see substantial improvements. Figure 4-4 displays several alignments of Flickr8k images to their captions inferred by our system. While by no means perfect, our system reliably aligns salient objects in the images with their associated caption words.

We also trained several different word spectrogram CNNs with varying configurations. Table 4.2 displays the top-1 and top-5 accuracies of a few of these networks. A two-layer conventional DNN with 1024 units per layer and ReLU nonlinearities achieved a classification accuracy of 75.5%, while adding a third layer brought that number even lower to 69.5%. We speculate that our training set is not large enough to train such a network. However, replacing the first fully connected layer with a 64-unit convolutional layer boosted the accuracy to 84.2%. We also trained a network with two convolutional layers and one fully connected layer and achieved similar results to the network with only a single convolutional layer. We also explored varying

Model	Search R@10	Annotation R@10
Socher et al. (2014)	28.6	29.0
Karpathy et al. (2014)	42.5	44.0
Text + word vec	49.0	56.7
Spectrogram CNN	17.9	24.3

Table 4.1: Image search and annotation results on the Flickr8k test images (1000 images with 5 captions each).

Model	Top-1 Acc.	Top-5 Acc.
DNN, 2x1024 FC	75.5	93.9
DNN, 3x1024 FC	69.5	91.4
CNN, 1x64 Conv + 2x1024 FC	84.2	97.4

Table 4.2: Isolated word recognition accuracies on our WSJ test set. "FC" stands for "fully connected".

the size and shapes of the convolutional filters, pooling layers, and dimension of the fully connected layers, but the network achieving 84.2% accuracy reflects our best performance. Although these networks show a wide range of top-1 accuracies, it is interesting to note that their top-5 accuracies are all in excess of 90%. Figure 4-2 displays the 64 filter responses from the first layer of our network.

4.5 Chapter Summary

In this chapter, we have presented our first efforts to construct a model which can learn a joint semantic representation over spoken words as well as visual objects. At training time, the model only requires weak labels in the form of paired images and natural language spoken captions. Our system aligns salient visual objects in the images with their associated caption words, in the process building a semantic representation across both modalities. We evaluate our model on the Flickr8k image search and annotation tasks, and compare it to several systems with access to the ground truth text. Encouraged by our findings, in the next chapter we "remove the training wheels," so to speak, by intoducing improved models which are able to cope with unsegmented audio and visual inputs.

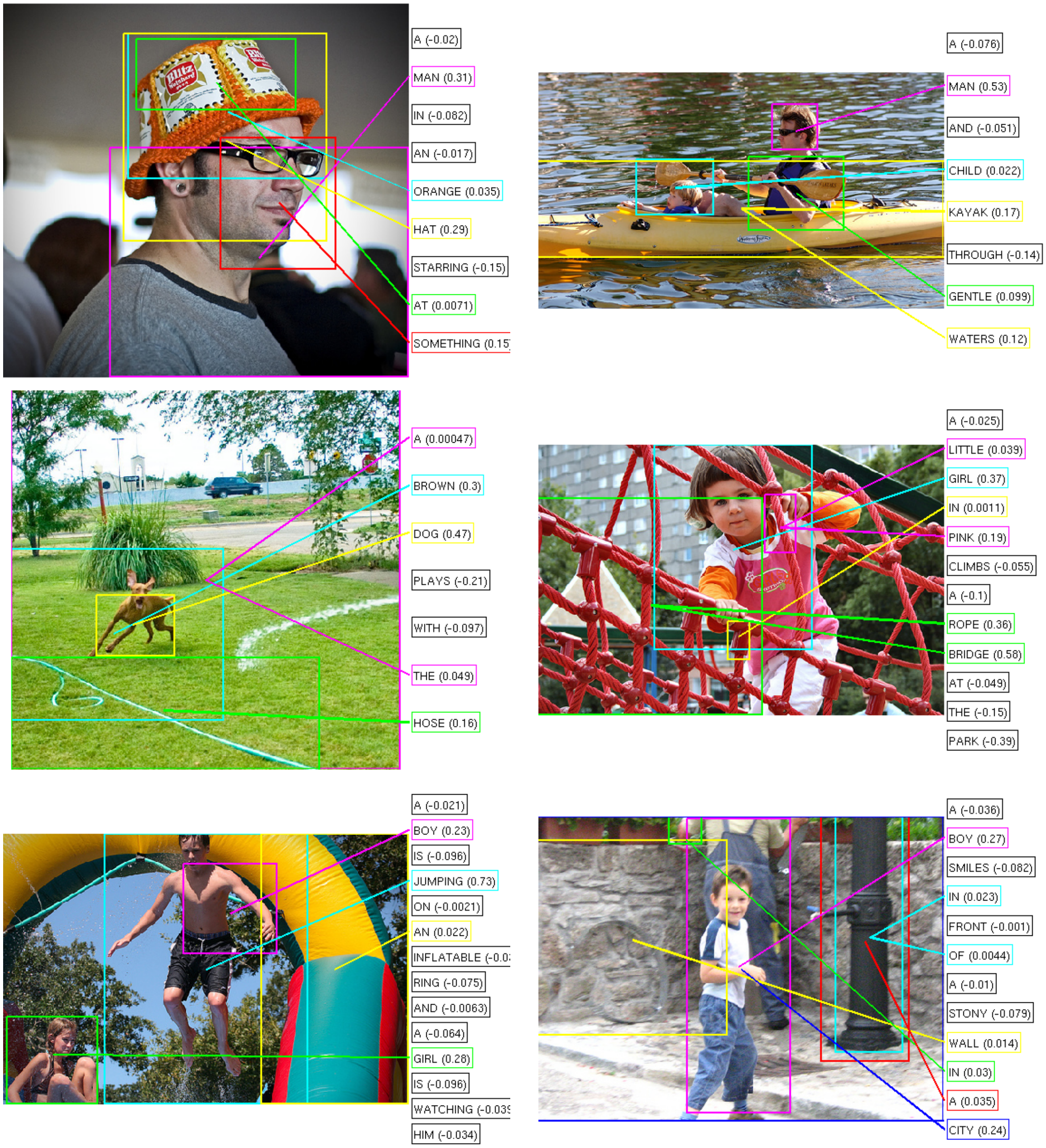


Figure 4-4: Some examples of inferred alignments on the Flickr8k data. The words for each image's caption are stacked to the right of each image, accompanied by their alignment scores. To keep the images free from too much clutter, we threshold the scores at 0, displaying a link between the word and its maximally associated object bounding box only when its score is positive. Note that the system does not actually see the text of the caption words - only a spectrogram. We replace the spectrogram in these figures with the ground truth text for the sake of clarity.

Chapter 5

Grounding Speech to Images: The Unsegmented Case

In Chapter 4, we considered learning a joint embedding and alignment model for the purpose of associating images and spoken captions describing those images. Recall that the model described in Chapter 4 relied upon the following assumptions:

1. The spectrograms of the spoken audio captions were pre-segmented at the word level,

2. A neural network pre-trained in a supervised fashion to perform isolated word recognition was used to extract initial embeddings for the spectrogram segments,

3. The images were pre-segmented into object proposal regions with a R-CNN (Girshick et al., 2013) network, which was also used to extract embeddings for said regions, that was pre-trained in a supervised fashion.

In this chapter, we do away with these assumptions and develop models capable of learning directly from unsegmented images and waveforms. Although some of the experiments to follow assume that an image network pre-trained on a separate supervised classification task is available, we also investigate what is possible to learn in a completely unsupervised fashion.

Portions of the work presented in this chapter were first published in Harwath et al. (2016); Harwath and Glass (2017).

5.1 Introduction

In this chapter, we introduce novel neural network architectures for the purpose of learning high-level semantic concepts across both the audio and visual modalities. Like the networks described in Chapter 4, the models presented here operate on contextually correlated streams of sensor data from multiple modalities, namely a visual image accompanied by a spoken audio caption describing that image. Unlike the networks from Chapter 4, these models are designed to operate on entire images and their captions. This alleviates the need for pre-segmentation of the inputs, but we show that the models are still able to learn meaning from continuous speech. We validate this experimentally by performing semantic image and caption retrieval. Finally, we conduct preliminary analysis that suggests that the networks are implicitly learning to localize important words, as well as discriminate between them.

5.2 Audio-Visual Modeling

In the simplest sense, all of our models are designed to calculate a similarity score for any given image and caption pair, where the score should be high if the caption is relevant to the image, and low otherwise. It is similar in spirit to “Siamese” models which attempt to learn a similarity measure within one modality (Chopra et al., 2005). In general, neural Siamese models consist of three components: a feature extraction network (sometimes called an embedding network), a distance function, and a loss function. At training time, two exemplars are sampled from the training set and passed through the embedding network to extract a representation for each exemplar. The scoring function computes the distance (or similarity, depending on how the loss function is formulated) between the representations, typically using standard measures such as cosine distance, Euclidean distance, or Kullback-Leibler divergence.

It is also possible to learn the distance function by using a second neural network whose inputs are a pair of embedding vectors and whose output is a scalar value. The final component of the model is the loss function, whose job is to provide a top-level gradient that can be backpropagated into the rest of the model. The loss function encodes our expectations as to whether two input exemplars should have a high distance between them (e.g. the inputs are pictures of faces from two different people) or should have a low distance between them (e.g. the inputs are two different pictures of the same person's face).

Unlike the Siamese model setting, in which a shared embedding network can be used for both inputs, here we are dealing with inputs from multiple modalities that cannot necessarily be well-modeled using a single network. Therefore, our models employ separate branches for visual inputs and for audio inputs. Broadly speaking, the branches can be trained to represent an input (an image or a caption, depending on the branch in question) two different ways:

1. **Embedding Vector Models:** The output representation takes the form of a vector. This vector encodes its entire input (image or audio caption) as a single point in a high dimensional space. Hence, the vector captures a holistic view of its input, but localized information is lost.

2. **Feature Map Models:** The output representation takes the form of an N^{th} order tensor ($N = 2$ for audio captions, $N = 3$ for images), or feature map. This feature map encodes localized representations of its input; each output unit's receptive field with respect to the input can be recovered.

We next describe our data preprocessing steps, and then discuss the specific architectures that we have developed for both of the above cases.

5.2.1 Data Preprocessing and Normalization

To preprocess our images, we follow the same scheme used by the VGG16 network Simonyan and Zisserman (2014). Because the VGG network was pre-trained on the

ImageNet ILSVRC12 (Deng et al., 2009) dataset for several of our experiments, the mean pixel value of that dataset is first subtracted from each input image. Next, the standard deviation of each of the three color channels is computed across the ILSVRC12 dataset, and the color channel of each pixel of an input image is divided by the corresponding standard deviation. The image is then resized proportionally so that its smallest dimension is equal to 256. We then we take a random 224 by 224 crop for training, or the center 224 by 224 crop for testing.

We use a log mel-filterbank spectrogram to represent the spoken audio caption associated with each image. Generating the spectrogram transforms the 1-dimensional waveform into a 2-dimensional signal with both frequency and time information. For a detailed explanation of this feature extraction procedure, we refer the reader to Section 2.1. In a manner similar to how the images are preprocessed, we compute a single scalar mean value by averaging over all frames and frequency bins across all caption spectrograms in the training set. We compute a corresponding standard deviation statistic as well. The mean is subtracted from each pixel (time-frequency bin) of an input spectrogram, and then the resulting value is divided by the standard deviation. This differs from the spectrogram normalization scheme used in Chapter 4, where we computed a mean spectrogram across the entire training set. In order to take advantage of the additional computational efficiency offered by performing gradient computation across batched input, we force every caption spectrogram to have the same size. We do this by fixing the spectrogram size at L frames (1024 to 2048 in our experiments, respectively corresponding to approximately 10 and 20 seconds of audio). We truncate any captions longer than L, and zero pad any shorter captions; approximately 66% of the captions used in the experiments in this chapter were found to be 10 seconds or shorter, while 97% were under 20 seconds.

5.2.2 Embedding Vector Models

In the case of the embedding vector model, the final layer of each branch outputs a vector of activations. This vector should capture the overall semantic content of its input image or caption, in a way that allows the semantic similarity between two

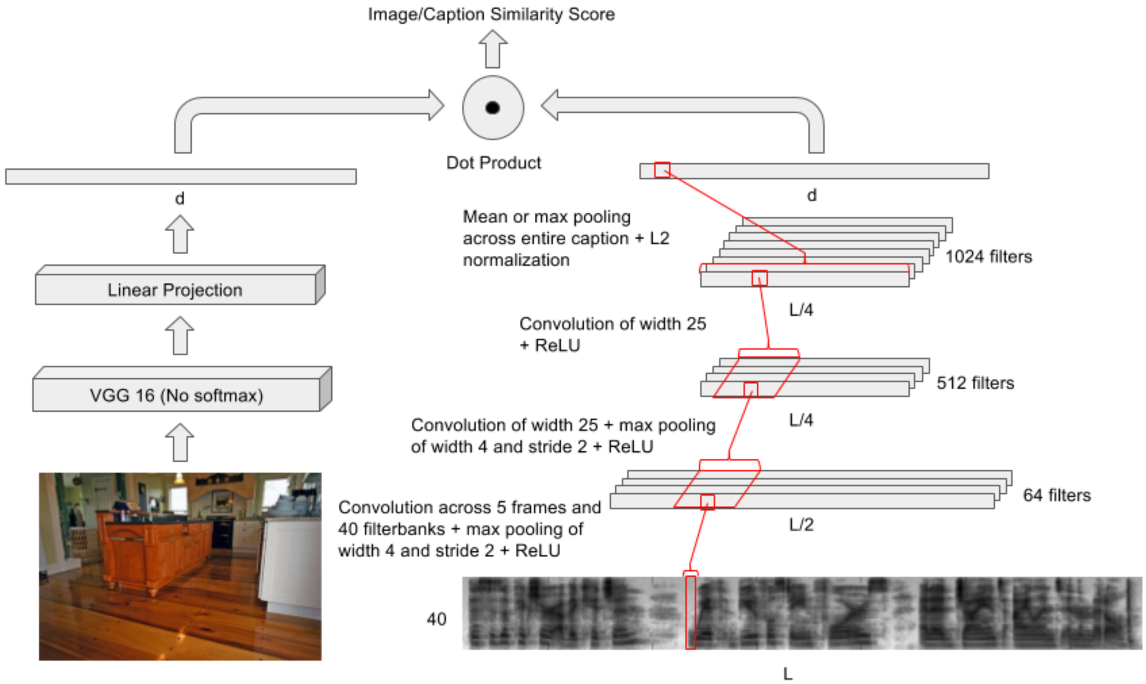

Image/Caption Similarity Score
Dot Product
d
Linear Projection
VGG 16 (No softmax)
d
Mean or max pooling
across entire caption + L2
normalization
1024 filters
L/4
Convolution of width 25
+ ReLU
512 filters
L/4
Convolution of width 25 + max pooling
of width 4 and stride 2 + ReLU
64 filters
L/2
Convolution across 5 frames and
40 filterbanks + max pooling of
width 4 and stride 2 + ReLU
40
L

Figure 5-1: The architecture of the NIPS16 audio/visual neural network with the embedding dimension denoted by d and the caption length by L. Separate branches of the network model the image and the audio spectrogram, and are subsequently tied together at the top level with a dot product node which calculates a similarity score for any given image and audio caption pair.

inputs to be computed using vector operations (such as the inner product). One variant of the general architecture we use is illustrated in Figure 5-1 (henceforth referred to as the NIPS16 architecture, as it was first published in (Harwath et al., 2016)). The VGG16 network effectively forms the bulk of the image branch; we remove the softmax classification layer, and keep the bottom portion of the network up through the `fc2` layer. At that point, we need to have a means of mapping the 4096-dimensional outputs of `fc2` into a d-dimensional vector space that the images and audio will share. For this purpose we employ a simple linear transform, allowing us to arbitrarily specify any dimension for the shared embedding space.

The audio branch of our network is also convolutional in nature and treats the spectrogram as a 1-channel (grayscale) image. However, speech spectrograms have a few interesting properties that differentiate them from images. While it is easy to imagine how visual objects in images can be translated along both the vertical and horizontal axes, the same is not quite true for words in spectrograms. A time delay manifests itself as a translation in the temporal (horizontal) direction, but a fixed pitch will always be mapped to the same frequency bin on the vertical axis. The same phone pronounced by two different speakers will not necessarily contain energy at exactly the same frequencies, but the physics is more complex than simply shifting the entire phone up and down the frequency axis. Following the technique we previously employed in Chapter 4, we size the filters of the first layer of the network to capture the entire 40-dimensional frequency axis. This means that the vertical dimension is effectively collapsed out in the first layer, and so subsequent layers are only convolutional in the temporal dimension. After the final layer, we pool across the entire caption in the temporal dimension (using either mean or max pooling). Some form of normalization is also possible to apply at this point; we use L2 normalization in our models.

In addition to the NIPS16 architecture displayed in Figure 5-1, we also experiment with the deeper audio branch presented in (Harwath and Glass, 2017), which we will refer to as the ACL17 architecture:

1. Convolution: Channels=128, Width=1, Height=40, ReLU
2. Convolution: Channels=256, Width=11, Height=1, ReLU
3. Maxpool: Width=3, Height=1, Stride=2
4. Convolution: Channels=512, Width=17, Height=1, ReLU
5. Maxpool: Width=3, Height=1, Stride=2
6. Convolution: Channels=512, Width=17, Height=1, ReLU
7. Maxpool: Width=3, Height=1, Stride=2
8. Convolution: Channels=d, Width=17, Height=1, ReLU
9. Meanpool over entire caption
10. L2 normalization

During training and testing, we compute a similarity score between an arbitrary image and caption by performing a forward pass of the inputs through their respective branches, and then computing the inner product of their output embedding vectors.

5.2.3 Feature Map (Matchmap) Models

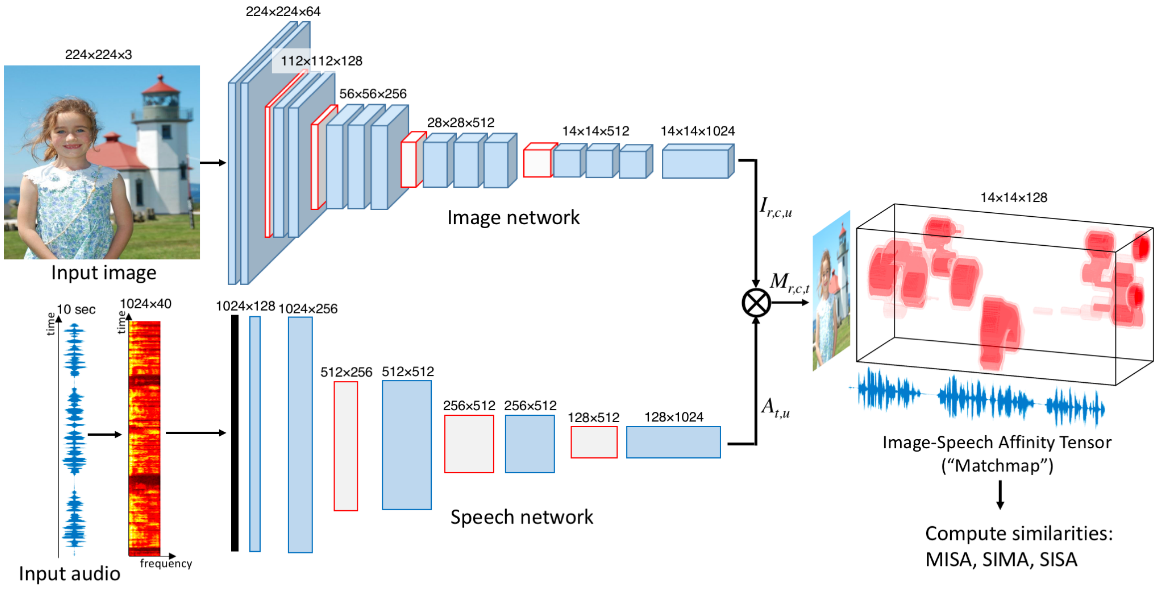


Figure 5-2: The audio-visual matchmap model architecture (left), along with an example matchmap output (right), displaying a 3-D density of spatio-temporal similarity.

In the NIPS16 and ACL17 architectures, the embedding vector representing an image is derived by flattening the convolutional feature map output by the `conv5` layer of the VGG16 network, and then sending these outputs through several fully-connected layers. One problem with this approach is that it is difficult to recover associations between any neuron in the fully connected layers and the spatially localized input stimulus which was responsible for its output. While the convolutional units in VGG16 all have spatial receptive fields which are simple to derive, the receptive field for the units in the flully connected layers is effectively the entire input image.

Rather than encapsulating the semantics of the entire image scene within a single embedding vector, an alternative approach is to use only convolutional and pooling layers in the image model. We do this by retaining only the layers up through `conv5` from the VGG16 network, discarding `pool5` and everything above it. For a 224 by 224 pixel input image, the output of this portion of the network would be a feature map across 512 channels, spanning 14 by 14 superpixels. Each superpixel location within the map possesses a receptive field that can be related directly back to the input. In order to map the image into the shared embedding space, we apply a 3 by 3,

d channel, linear convolution (no nonlinearity) to the `conv5` feature map.

The audio branch of the network can be adapted to output a feature map in a similar way. We borrow this branch from the ACL17 architecture previously described, but remove the final L2 normalization as well as the preceeding meanpooling operation over the entire caption. For an input spectrogram represented by T frames each containing 40 filterbank energies, the output of this modified audio branch will thus be a d-dimensional feature map across $\frac{T}{8}$ temporal superframes. We make one further small modification to the ACL17 model by introducing an initial batch normalization Ioffe and Szegedy (2015) layer that operates on the spectrogram input. We allow this layer to handle the spectrogram-space normalization, and thus do not manually apply mean and variance normalization to the spectrograms in this case.

We refer to the combination of the feature map-based audio and image branches as the matchmap model, visualized in Figure 5-2. To compute the overall similarity between an image and caption using the matchmap model, a simple inner product will no longer suffice; we require a function that takes as input the visual and audio feature maps and outputs a scalar score. We next describe three suitable ways of doing so.

Let I represent the output feature map output of the image network branch, A be the output feature map of the audio network branch. Our first step is to compute a 3rd order tensor M such that $M_{r,c,t} = I^T_{r,c,:}A_{t,:}$. Here we use the colon (:) to indicate selection of all elements across an indexing plane; in other words, $I_{r,c,:}$ is a d-dimensional vector representing the (r, c) superpixel coordinate of the image feature map, and $A_{t,:}$ is a d-dimensional vector representing the t^{th} superframe of the audio feature map. In other words, each element of M represents the dot product between a specific superpixel output by the final convolutional layer of the image network, and a specific superframe output by the audio network. Because M reflects the localized similarity between a small image region (possibly containing an object) and a small segment of audio (possibly containing a word), we refer to M as the matchmap tensor between and image and an audio caption. Once we have computed a matchmap tensor M, we consider three ways of deriving a similarity score from it.

The first possibility is to compute the average of all elements of M. We call this similarity scoring function SISA (sum image, sum audio):

$$\text{SISA}(M) = \frac{1}{N_r N_c N_t} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \sum_{t=1}^{N_t} M_{r,c,t} \tag{5.1}$$

As it is not completely realistic to expect all words within a caption to simultaneously match all objects within an image, we consider computing the similarity between an image and an audio caption using several alternative functions of the matchmap density. By replacing the averaging summation over image patches with a simple maximum, MISA (max image, sum audio) effectively matches each frame of the caption with the most similar image patch, and then averages over the caption frames:

$$\text{MISA}(M) = \frac{1}{N_t} \sum_{t=1}^{N_t} \max_{r,c}(M_{r,c,t}) \tag{5.2}$$

By preserving the sum over image regions but taking the maximum across the audio caption, SIMA (sum image, max audio) matches each image region with only the audio frame with the highest similarity to that region:

$$\text{SIMA}(M) = \frac{1}{N_r N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} \max_{t}(M_{r,c,t}) \tag{5.3}$$

5.2.4 Model Training

For either of the model types described above (embedding vector or matchmap), we are able to compute a score S reflecting the similarity between an arbitrary image and an arbitrary caption. In the case of the embedding vector models (NIPS16 and ACL17), we accomplish this with an inner product; in the case of the matchmap model, we employ one of SISA, MISA, or SIMA. Regardless of how we compute S, we want this score to be high for ground-truth pairs - that is, images and captions that go together - and low otherwise. We therefore specify a margin-based ranking objective function (Bromley et al., 1994) which compares the similarity scores between matched image/caption pairs and mismatched pairs. We will consider the case in which this objective is optimized with batched stochastic gradient descent. Each minibatch consists of B ground truth pairs, each of which is paired with one impostor image and one impostor caption randomly sampled from the same minibatch. Let S_j^p denote the similarity score between the jth ground truth pair, S_j^c be the score between the original image and the impostor caption, and S_j^i be the score between the original caption and the impostor image. The loss for the minibatch as a function of the network parameters θ is defined as:

$$\mathcal{L}(\theta) = \sum_{j=1}^{B} \max(0, S_j^c - S_j^p + \eta) + \max(0, S_j^i - S_j^p + \eta) \tag{5.4}$$

This loss function was encourages the model to assign a higher similarity score to a ground truth image/caption pair than a mismatched pair by a margin of η (we generally fix $\eta = 1$ in our experiments). In Karpathy et al. (2014) the authors used a similar objective function to align images with text captions, but every single mismatched pair of images and captions within a minibatch was considered. Here, we only sample two negative training examples for each positive training example, which we found led to more stable training.

5.3 Experimental Data

We perform experiments on a dataset of over 400,000 spoken image captions from the Places205 Audio Caption dataset (Zhou et al., 2014) described in Chapter 3, corresponding to over 1,000 hours of speech data. A held-out set of 1,000 image/caption pairs is used for validation.

Because we lack ground truth text transcripts for the data, we used Google's Speech Recognition public API to generate proxy transcripts which we use when analyzing our system. Note that the ASR was only used for analysis of the results, and was not involved in any of the learning. To estimate the word start and end times for our analysis figures, we used Kaldi (Povey et al., 2011) to force align the caption audio to the ASR-derived transcripts. Given the difficult nature of our data, these transcripts are by no means error free. To get an idea of the error rates offered by the Google recognizer, we manually transcribed 100 randomly selected captions and found that the Google transcriptions had an estimated word error rate of 23.17%, indicating that the transcriptions are somewhat erroneous but generally reliable.

5.4 Image Query and Annotation Experiments

To objectively evaluate our models, we adopt the same image search and annotation task used in Chapter 4, applied to our held-out validation set of 1,000 image/caption pairs. This task serves to provide a single, high-level metric which captures how well the model has learned to semantically bridge the audio and visual modalities.

All of the models used in the following experiments were trained on NVIDIA Titan X GPUs. Typical training times ranged between several days and 2 weeks depending upon the size of the dataset and number of training epochs used. In nearly all cases, we set our minibatch size to 128, used a constant momentum of 0.9, and ran SGD training until convergence (typically reached within 50 to 200 epochs). We found that an initial (batchsize-normalized) learning rate of 0.001 worked well in most cases, using a decay schedule that decreased the learning rate by a factor of 10 every 30 epochs. We found that an embedding dimension of $d = 1024$ worked well, and the retrieval performance was not overly sensitive to the exact setting of d. Good settings for d were found to be between 768 and 2048.

We experimented with many different variations on the NIPS16 model architecture, as it was the first model we developed. These variations included the number of hidden units, number of layers, filter sizes, embedding dimension, and embedding normalization schemes. When only the acoustic embedding vectors were L2 normalized, we saw a consistent increase in performance. However, when the image embeddings were also L2 normalized (equivalent to replacing the dot product similarity with a cosine similarity), the recall scores suffered. In Table 5.1, we show the impact of various truncation lengths for the audio captions, as well as using a mean or max pooling scheme across the audio caption. We found that truncating the captions to 20 seconds instead of 10 only slightly boosts the scores, and that mean and max pooling work about equally well. These experiments reflect the use of an image branch pre-trained on ImageNet, with fixed weights during our multimodal training (in other words, only the audio branch and the final projection layer of the image branch were trained). Some example search and annotation results are displayed in Figures 5-3 and 5-4.

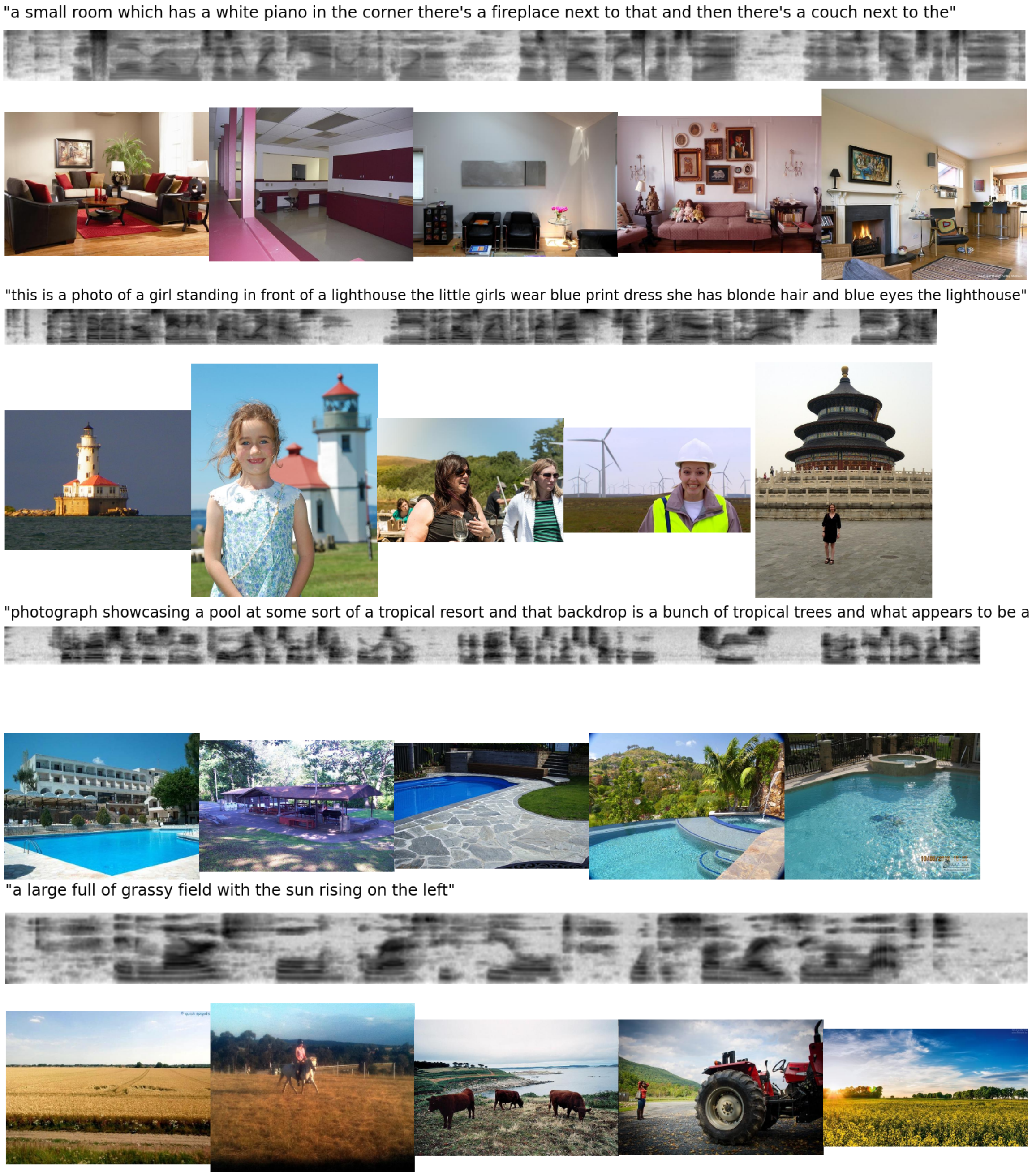
Model Variant		Search			Annotation		
Pooling type	Caption length (s)	R@1	R@5	R@10	R@1	R@5	R@10
Mean	10	.056	.192	.289	.051	.194	.283
Mean	20	.066	.215	.299	.082	.195	.295
Max	10	.069	.192	.278	.068	.190	.274
Max	20	.068	.223	.309	.061	.192	.291

Table 5.1: Experimental results for image search and annotation on a 120,000 caption subset of the Places Audio data, using variants of the NIPS16 architecture. All models shown used an embedding dimension of 1024.

5.4.1 Model Comparison on Full 400k Training Set

In Table 5.2, we display the recall scores on the full Places Audio dataset for the NIPS16, ACL17, and three variants of the matchmap model. We also compare against a text-based matchmap model that operates on the ASR transcripts of the captions. The text-based model replaces the speech audio branch with a CNN that operates on word sequences. The text branch uses a 200-dimensional word embedding layer, followed by a 512 channel, 1-dimensional convolution across windows of 3 words with a ReLU nonlinearity. A final convolution with a window size of 3 and no nonlinearity maps these activations into the 1024 multimodal embedding space. All of the models detailed in Table 5.2 utilize an image network pre-trained in a supervised fashion on ImageNet.

There are several key takeaways from the results in Table 5.2. First, a deeper audio network (ACL17) seems to offer significant improvements over the NIPS16 architecture, especially for audio caption retrieval. Similarly, the presence of the additional fully



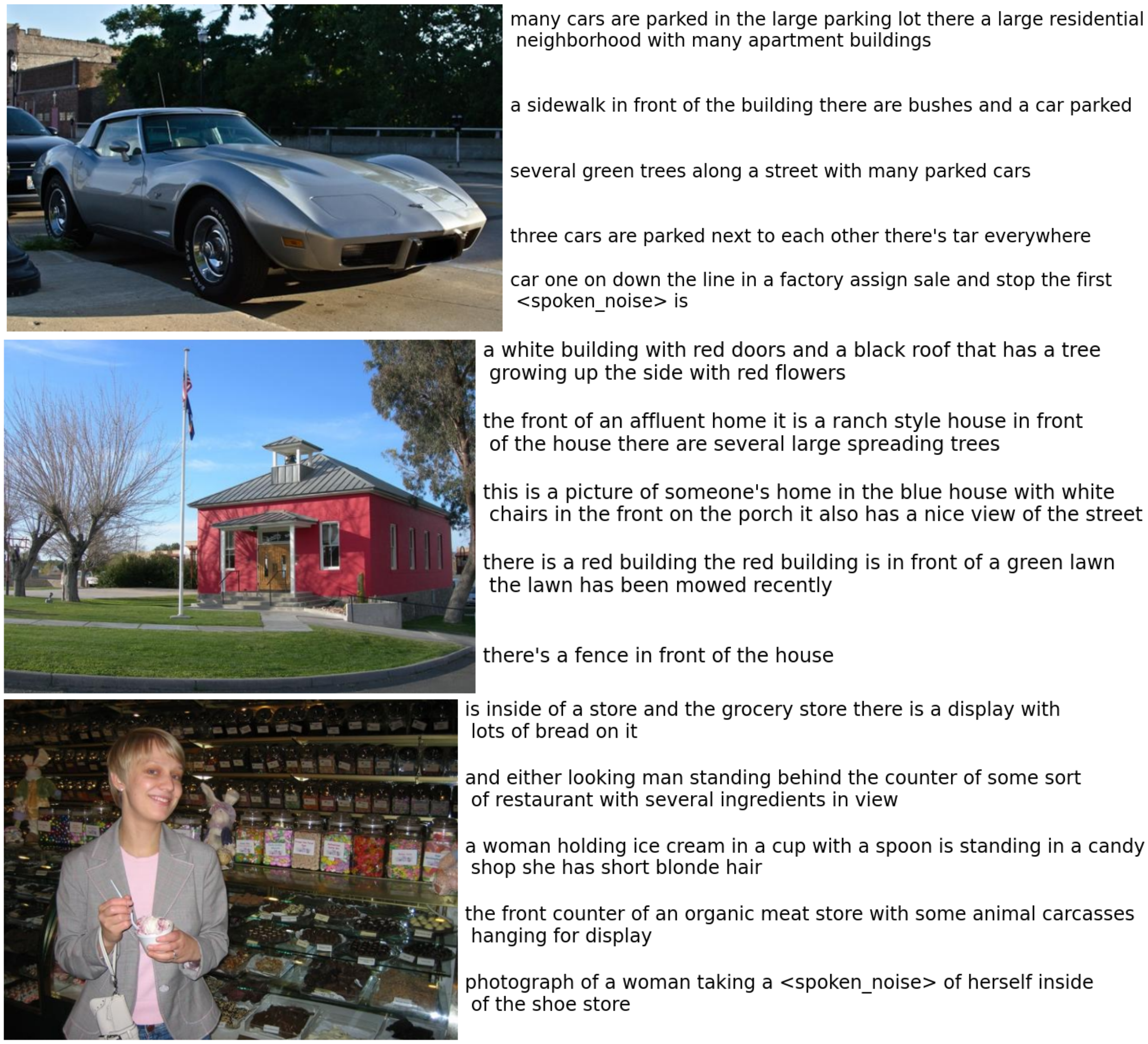
"a small room which has a white piano in the corner there's a fireplace next to that and then there's a couch next to the"

"this is a photo of a girl standing in front of a lighthouse the little girls wear blue print dress she has blonde hair and blue eyes the lighthouse"

"photograph showcasing a pool at some sort of a tropical resort and that backdrop is a bunch of tropical trees and what appears to be a

"a large full of grassy field with the sun rising on the left"

Figure 5-3: Example search results for the NIPS16 model. Shown on the top is the spectrogram of the query caption, along with its speech recognition hypothesis text. Below each caption are its five highest scoring images from the test set.



many cars are parked in the large parking lot there a large residential neighborhood with many apartment buildings

a sidewalk in front of the building there are bushes and a car parked

several green trees along a street with many parked cars

three cars are parked next to each other there's tar everywhere

car one on down the line in a factory assign sale and stop the first <spoken_noise> is

a white building with red doors and a black roof that has a tree growing up the side with red flowers

the front of an affluent home it is a ranch style house in front of the house there are several large spreading trees

this is a picture of someone's home in the blue house with white chairs in the front on the porch it also has a nice view of the street

there is a red building the red building is in front of a green lawn the lawn has been mowed recently

there's a fence in front of the house

is inside of a store and the grocery store there is a display with lots of bread on it

and either looking man standing behind the counter of some sort of restaurant with several ingredients in view

a woman holding ice cream in a cup with a spoon is standing in a candy shop she has short blonde hair

the front counter of an organic meat store with some animal carcasses hanging for display

photograph of a woman taking a <spoken_noise> of herself inside of the shoe store

Figure 5-4: Example annotation results for the NIPS16 model. Shown on the left is the query image, and on the right are the Google speech recognition hypotheses of the five highest scoring audio captions from the test set. We do not show the spectrograms here to avoid clutter.

	Caption to Image			Image to Caption		
Model	R@1	R@5	R@10	R@1	R@5	R@10
NIPS16	.148	.403	.548	.121	.335	.463
ACL17	.161	.404	.564	.130	.378	.542
Matchmap SISA	.142	.368	.510	.118	.344	.489
Matchmap MISA	.166	.413	.559	.135	.369	.499
Matchmap SIMA	.131	.360	.493	.130	.356	.467
ASR SISA (Text)	.167	.457	.603	.168	.432	.553
ASR MISA (Text)	.226	.500	.638	.177	.442	.563
ASR SIMA (Text)	.180	.462	.617	.205	.469	.598

Table 5.2: Recall scores on the heldout set of 1,000 images/captions for the four matchmap similarity functions considered. Random chance recall scores are 0.001 for R@1, 0.005 for R@5, and 0.01 for R@10.

connected layers of the VGG16 network used in the ACL17 architecture boost the caption recall scores (as compared to the matchmap model which lacks these layers). This indicates that the compact vector representation used by the ACL17 model may do a more effective job at capturing the holistic semantics of its inputs than the distributed representations learned by the matchmap models. Finally, we notice that the ASR text-based models outperform the speech-based models across the board - but not by an enormous margin. This indicates that our speech-based models are indeed learning to implicitly recognize words and their semantics - a phenomenon which we explore further in Chapter 6.

5.4.2 Image Architectures and Variable Pretraining

We also investigate the effect of using a different image architecture, Alexnet (Krizhevsky et al., 2012), as well as varying the degree of the visual pre-training. In the case of Alexnet, we discard the fully connected layers of the network, only keeping the convolutional architecture up through conv5 and before the final pooling layer. We vary the degree of visual pre-training by only using pre-trained weights up to a certain convolutional layer, randomly initializing all weights above that layer. Table 5.3 details the results of these for both the VGG16 and Alexnet. VGG16 outperforms Alexnet across the board, which is perhaps unsurprising given the fact that VGG16 is a far deeper network. In the case of both architectures, every additional pretrained layer increases the overall performance of the network by a significant amount. That said, both networks are able to achieve significantly better than chance (15 to 20 times better) recall scores even with no pretraining whatsoever. This indicates that completely unsupervised learning of both modalities is indeed possible.

	Caption to Image			Image to Caption		
Model	R@1	R@5	R@10	R@1	R@5	R@10
VGG16 P0	.030	.119	.200	.030	.110	.188
VGG16 P1	.045	.185	.289	.048	.160	.259
VGG16 P2	.074	.222	.334	.067	.203	.314
VGG16 P3	.082	.259	.384	.073	.226	.356
VGG16 P4	.118	.316	.445	.083	.276	.400
VGG16 P5	.142	.368	.510	.118	.344	.489
Alexnet P0	.019	.074	.151	.016	.088	.149
Alexnet P1	.049	.164	.256	.045	.139	.235
Alexnet P2	.066	.192	.284	.048	.170	.293
Alexnet P3	.060	.203	.297	.048	.183	.294
Alexnet P4	.069	.213	.330	.060	.204	.302
Alexnet P5	.076	.249	.356	.075	.238	.332

Table 5.3: Image and caption retrieval results for VGG16 and Alexnet with various degrees of pretraining. P0 corresponds to no pretraining, P1 corresponds to a pretrained conv1, P2 to a pretrained conv1 and conv2, and so on (P5 meaning a fully pretrained image network). In the case of the VGG16 network, each “layer” actually corresponds to each named bank of convolutions (according to the standard VGG nomenclature). All networks were trained using the SISA matchmap similarity function.

5.4.3 Padding Compensation, Activation Functions, and Random Initialization

Given that we have just shown that learning both the image network and audio network simultaneously and completely from scratch (i.e. no pre-training of the VGG16 network on ImageNet) is possible but with worse performance, we detail some preliminary strategies for bridging the gap. By placing a hyperbolic tangent nonlinearity at the overall output of the image branch, as well as replacing the final ReLU belonging to the audio branch with a hyperbolic tangent, we can restrict the possible range of values that the inner product between embedding vectors will take to the interval $[-d, d]$. We hypothesize that this constraint may help to make learning more stable when starting from a randomly initialized image network. It is also possible to counteract the effect of the zero-padding when using the Matchmap networks by only computing the matchmap up until the last output frame that does not correspond to the padded portion of an audio input. This removes the artifact of the padding during the similarity scoring; to fully take advantage of this, we extend the maximum caption length to 2048 frames, capturing the full duration of approximately 97% of our captions. The effect of these changes in the case of a pre-trained image branch as well as a randomly initialized image branch are displayed in Table 5.4. As the fully randomly initialized training case is part of our ongoing work, the models in Table 5.4 do not yet reflect an exhaustive set of experiments. We do note that significant gains can be had for the fully unsupervised training case with the addition of the tanh nonlinearities, padding compensation, and MISA scoring function, relative to the configuration detailed in Table 5.3. Table 5.4 also contains the highest scoring overall network within this thesis, which utilized an embedding dimension of 2048, ReLU nonlinearities, the MISA scoring function, and padding compensation. More experiments are necessary in order to determine the individual importance of each of these architecture modifications.

			Caption to Image			Image to Caption		
d	PT VGG16	Out Act.	R@1	R@5	R@10	R@1	R@5	R@10
1024	No	tanh	.108	.287	.404	.073	.229	.327
2048	Yes	tanh	.200	.449	.584	.115	.352	.482
2048	No	tanh	.099	.269	.370	.063	.193	.300
2048	Yes	ReLU	.193	.486	.611	.161	.412	.553
2048	No	ReLU	.092	.250	.351	.066	.200	.299

Table 5.4: Various model and training configurations when using the MISA scoring function and audio padding compensation. 'PT VGG16' indicates whether or not the image branch is pretrained with the ImageNet weights, and 'Out Act.' indicates the output activation type. For this column, 'ReLU' indicates a ReLU on the audio network output, but not on the image branch which still uses a linear output. For the 'tanh' case, hyperbolic tangents are used on the output of both network branches.

5.4.4 Preliminary Localization Analysis

While image and caption retrieval is a useful task in and of itself, our ultimate goal is to discover object-like patterns in the images, and word-like patterns in the speech. Even better, we would like to be able to semantically relate these individual patterns to one another. At the time we published the NIPS16 model architecture in Harwath et al. (2016), we had not yet developed any way of doing this. However, in that paper we performed some preliminary analysis of the NIPS16 model by computing time-dependent similarity profiles for image/caption pairs. This was done by removing the final pooling layer from the spectrogram branch of a trained NIPS16 model, leaving a temporal sequence of vectors reflecting the activations of the top-level convolutional units with respect to time. We computed the dot product of the image embedding vector with each of these vectors individually, rectified the signal to show only positive similarities, and then applied a 5th order median smoothing filter. This filter is simply to present a smoother curve for visual analysis, as it is otherwise quite jagged. We time aligned the recognition hypothesis to the spectrogram, allowing us to see exactly which words overlapped the audio regions that were highly similar to the image. Figure 5-5 displays several examples of these similarity curves along with the overlaid recognition text. In the majority of cases, the regions of the spectrogram which have the highest similarity to the accompanying image turn out to be highly informative words or phrases, often making explicit references to the salient objects in the image scenes. This suggested that our network is in fact learning to recognize audio patterns consistent with words using zero linguistic supervision whatsoever, and perhaps even more impressively is able to learn their semantics.

5.4.5 Preliminary Analysis of Word Discriminability

To further examine the high-level acoustic representations learned by the NIPS16 networks, we extracted spectrograms for 1645 instances of 14 different ground truth words from the development set by force aligning the Google recognizer hypotheses to the audio. We did a forward pass of each of these individual words through the

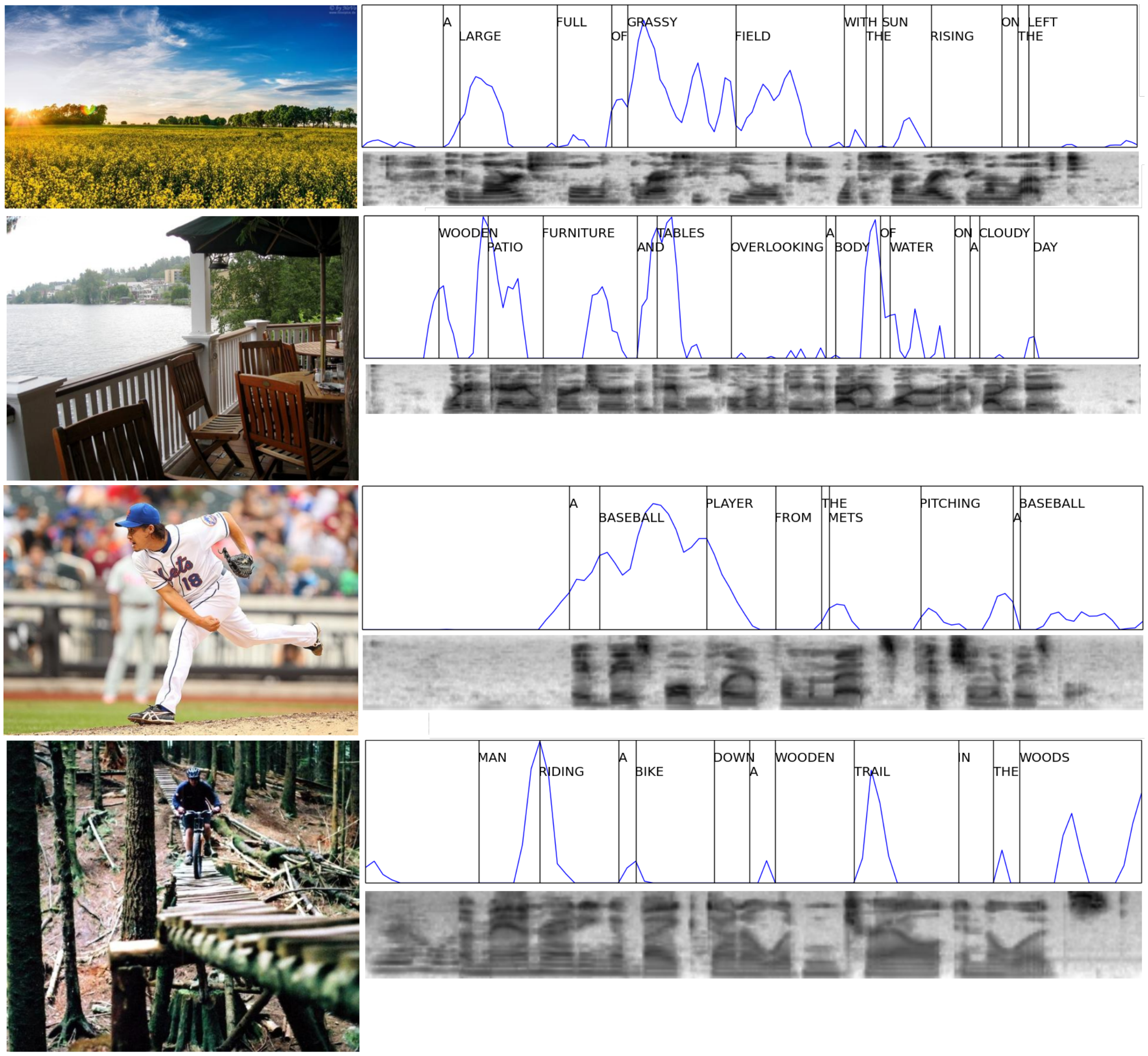


Figure 5-5: Examples of ground truth image/caption pairs along with the time-dependent similarity profile showing which regions of the spectrogram the model believes are highly relevant to the image. Overlaid on the similarity curve is the recognition text of the speech, along with vertical lines to denote word boundaries. Note that the neural network model had no access to these (or any) transcriptions during the training or testing phases.

audio branch of our network, leaving us with an embedding vector for each spoken word instance. We performed t-SNE (van der Maaten and Hinton, 2008) analysis on these points, shown in Figure 5-6. We observed that the points form pure clusters, indicating that the top-level activations of the audio network carry information which is discriminative across different words.

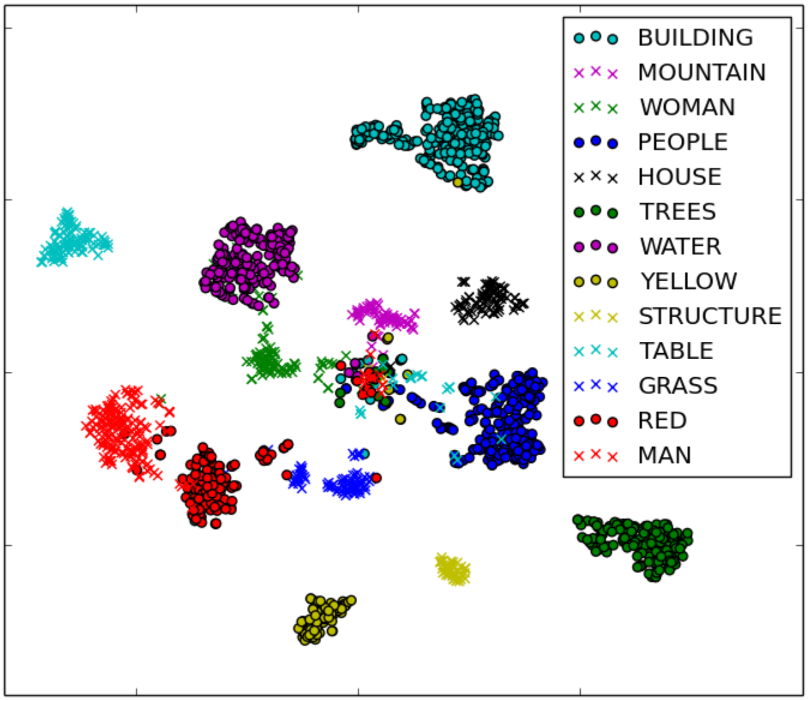


Figure 5-6: t-SNE visualization in 2 dimensions for 1645 spoken instances of 14 different word types taken from the development data.

The combination of the above two observations - namely, that the NIPS16 network was able to localize salient words within the caption spectrograms, as well as discriminate between different manually-extracted words - served as our inspiration for developing methods for automatic pattern discovery and grounding that we explore in the next chapter.

5.5 Chapter Summary

In this chapter, we presented a family of deep neural network architectures capable of learning associations between natural image scenes and accompanying free-form spoken audio captions. The networks do not rely on any form of conventional speech recognition, text transcriptions, or expert linguistic knowledge, but are able to learn to recognize semantically meaningful words and phrases at the spectral feature level. We show that this learning can take place with contextual information derived from the images as the only form of supervision, although pre-trained visual network weights from the VGG16 network improve performance. We presented experimental results in which the networks were used to perform image search and annotation tasks, as well as some preliminary analysis geared towards understanding the kinds of acoustic representations are being learned by the network. In Chapter 6, we will augment these models with the ability to perform automatic segmentation and clustering of audio-visual patterns.

Chapter 6

Jointly Discovering Words and Objects

In the previous chapter, we presented neural models capable of learning semantic associations between visual images and spoken descriptions of those images. This chapter makes use of those models for the joint discovery of word-like acoustic patterns and object-like visual patterns. We introduce two distinct methods for pattern discovery, and then chronicle a suite of experiments using those methods.

Portions of the work presented in this chapter were first published in Harwath and Glass (2017).

6.1 Problem Statement and Motivation

Recall that in Chapter 4 we introduced a highly-scaffolded neural model for learning associations between spoken words and visual objects. While the models presented were able to consistently learn these associations, this came at the cost of significant pre-training and lexical pre-segmentation of the audio signals. In Chapter 5, we removed most of this scaffolding, and presented models capable of learning audio-visual associations at the granularity of entire images and entire utterances. While these models relied on far less pre-training and no pre-segmentation, they did not possess a mechanism for producing localized alignments between an image and a caption; in other words, they were unable to identify the exact moment in time in which a speaker said the word “dog” and associate it with the minimal crop of an image scene containing a dog and little else.

In this chapter we introduce two distinct techniques for performing audio-visual pattern discovery. The first employs coupled sliding windows applied to images and spectrograms as an input pre-processing step, which are then fed into vector embedding networks of the type described in Chapter 5 to produce localized audio-visual semantic groundings. We show that this method is effective at deriving a “picture dictionary” - clusters of visual objects and the snippets of speech containing words that refer to them. However, one downside of the sliding window approach is that it requires many thousands of forward passes through the embedding network. The second technique we present utilizes the matchmap networks introduced in Chapter 5 to produce audio-visual groundings with a single forward pass. We show that connected components can be extracted from these matchmaps and clustered into word-object categories with high purity.

6.2 Pattern Grounding via Coupled Sliding Windows

Although we have trained our ACL17 multimodal network to compute embeddings at the granularity of entire images and entire caption spectrograms, we can easily apply it in a more localized fashion. In the case of images, we can simply take any arbitrary crop of an original, full-size image and resize it to 224x224 pixels. The audio network is even more trivial to apply locally, because it is entirely convolutional and the final mean pooling layer ensures that the output will be a 1024-dim vector no matter the extent of the input. The bigger question is *where* to locally apply the networks in order to discover meaningful acoustic and visual patterns.

Given an image and its corresponding spoken audio caption, we use the term grounding to refer to extracting meaningful segments from the caption and associating them with an appropriate sub-region of the image. For example, if an image depicted a person eating ice cream and its caption contained the spoken words “A person is enjoying some ice cream," an ideal set of groundings would entail the acoustic segment containing the word "person" linked to a bounding box around the person, and the segment containing the word "ice cream" linked to a box around the ice cream. We use a constrained brute force ranking scheme to evaluate all possible groundings (with a restricted granularity) between an image and its caption. Specifically, we divide the image into a grid, and extract all of the image crops whose boundaries sit on the grid lines. Because we are mainly interested in extracting regions of interest and not high precision object detection boxes, to keep the number of proposal regions under control we impose several restrictions. First, we use a 10x10 grid on each image regardless of its original size. Second, we define minimum and maximum aspect ratios as 2:3 and 3:2 so as not to introduce too much distortion and also to reduce the number of proposal boxes. Third, we define a minimum bounding width as 30% of the original image width, and similarly a minimum height as 30% of the original image height. In practice, this results in a few thousand proposal regions per image.

To extract proposal segments from the audio caption spectrogram, we similarly define a 1-dimensional grid along the time axis, and consider all possible start/end points at 10 frame (pixel) intervals. We impose minimum and maximum segment length constraints at 50 and 100 frames (pixels), implying that our discovered acoustic patterns are restricted to fall between 0.5 and 1 second in duration. The number of proposal segments will vary depending on the caption length, and typically number in the several thousands. Note that when learning groundings we consider the entire audio sequence, and do not incorporate the 10sec duration constraint imposed during training.

Once we have extracted a set of proposed visual bounding boxes and acoustic segments for a given image/caption pair, we use our multimodal network to compute a similarity score between each unique image crop/acoustic segment pair. Each triplet of an image crop, acoustic segment, and similarity score constitutes a proposed grounding. A naive approach would be to simply keep the top N groundings from this list, but in practice we ran into two problems with this strategy. First, many proposed acoustic segments capture mostly silence due to pauses present in natural speech. We solve

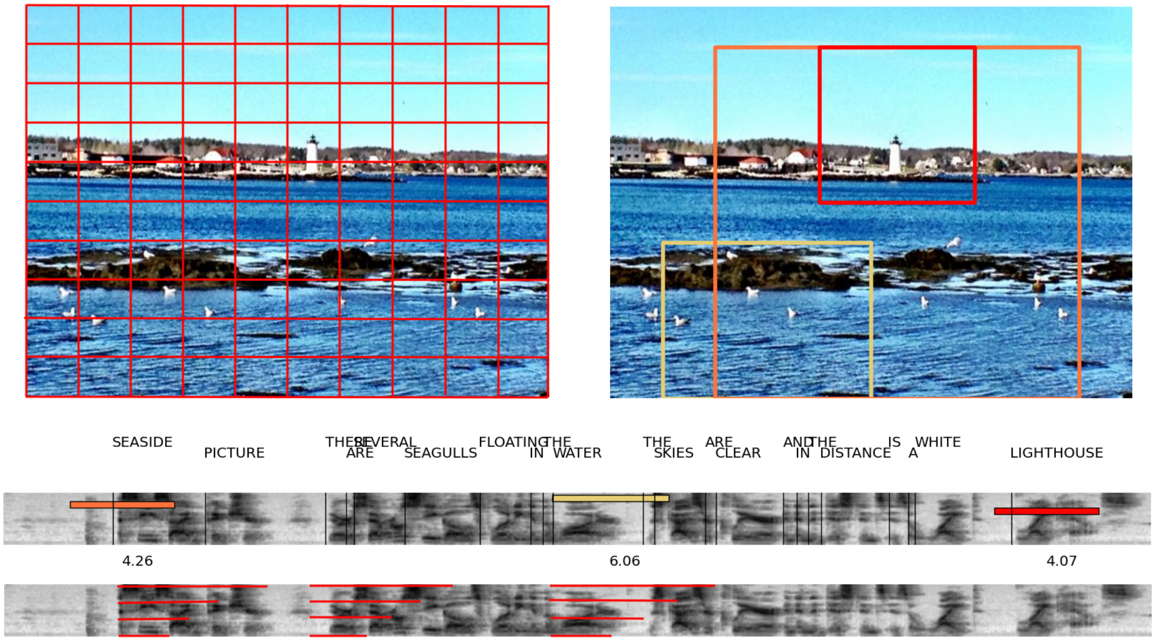


Figure 6-1: An example of our grounding method. The left image displays a grid defining the allowed start and end coordinates for the bounding box proposals. The bottom spectrogram displays several audio region proposals drawn as the families of stacked red line segments. The image on the right and spectrogram on the top display the final output of the grounding algorithm. The top spectrogram also displays the time-aligned text transcript of the caption, so as to demonstrate which words were captured by the groundings. In this example, the top 3 groundings have been kept, with the colors indicating the audio segment which is grounded to each bounding box.

this issue by using a simple voice activity detector (VAD) which was trained on the TIMIT corpus(Garofolo et al., 1993). If the VAD estimates that 40% or more of any proposed acoustic segment is silence, we discard that entire grounding. The second problem we ran into is the fact that the top of the sorted grounding list is dominated by highly overlapping acoustic segments. This makes sense, because highly informative content words will show up in many different groundings with slightly perturbed start or end times. To alleviate this issue, when evaluating a grounding from the top of the proposal list we compare the interval intersection over union (IOU) of its acoustic segment against all acoustic segments already accepted for further consideration. If the IOU exceeds a threshold of 0.1, we discard the new grounding and continue moving down the list. We stop accumulating groundings once the scores fall to below 50% of the top score in the “keep” list, or when 10 groundings have been added to the “keep” list. Figure 6-1 displays a pictorial example of our grounding procedure.

After the grounding procedure, we are left with a small set of regions of interest in each image and caption spectrogram. We use the respective branches of our multimodal network to compute embedding vectors for each grounding's image crop and acoustic segment. We then employ k-means clustering separately on the collection of image embedding vectors as well as the collection of acoustic embedding vectors. The last step is to establish an affinity score between each image cluster $\mathcal{I}$ and each acoustic cluster $\mathcal{A}$; we do so using the equation

$$\text{Affinity}(\mathcal{I}, \mathcal{A}) = \sum_{\mathbf{i} \in \mathcal{I}} \sum_{\mathbf{a} \in \mathcal{A}} \mathbf{i}^{\top} \mathbf{a} \cdot \text{Pair}(\mathbf{i}, \mathbf{a}) \tag{6.1}$$

where $\mathbf{i}$ is an image crop embedding vector, $\mathbf{a}$ is an acoustic segment embedding vector, and $\text{Pair}(\mathbf{i}, \mathbf{a})$ is equal to 1 when $\mathbf{i}$ and $\mathbf{a}$ belong to the same grounding pair, and 0 otherwise. After clustering, we are left with a set of acoustic pattern clusters, a set of visual pattern clusters, and a set of linkages describing which acoustic clusters are associated with which image clusters. In the next section, we investigate these clusters in more detail.

6.3 Sliding Window Grounding and Clustering Experiments

6.3.1 Clustering Analysis

For these experiments, we utilized a 220,000 image/caption subset of the Places Audio data. We trained the model on this data, and also performed the grounding and pattern clustering steps on the same data. This resulted in a total of 1,161,305 unique grounding pairs. For evaluation, we wish to assign a label to each cluster and cluster member, but this is not completely straightforward since each acoustic segment may capture part of a word, a whole word, multiple words, etc. Our strategy is to force-align the recognition hypothesis text to the audio, and then assign a label

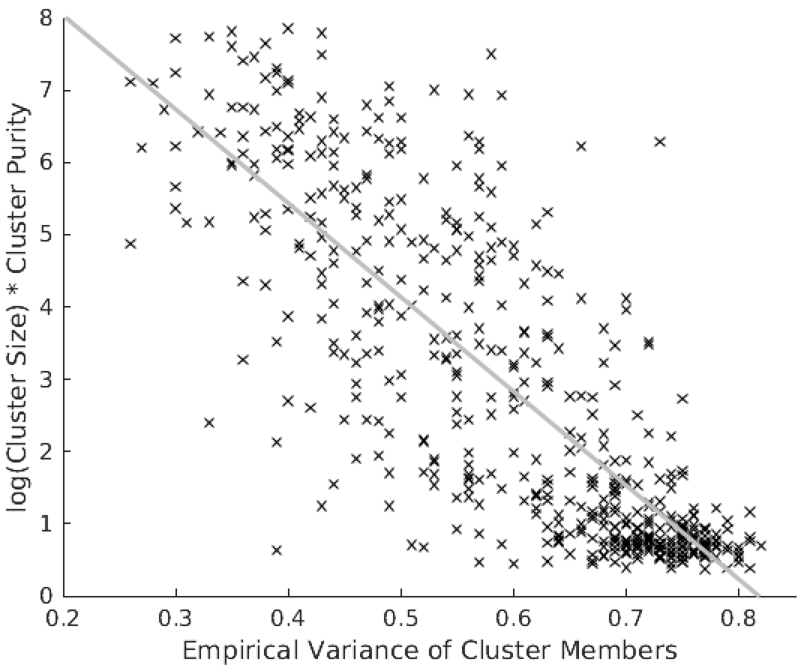


Figure 6-2: Scatter plot of audio cluster purity weighted by log cluster size vs variance for $k = 500$ (least-squares line superimposed).

string to each acoustic segment based upon which words it overlaps in time. Any word whose duration is overlapped 30% or more by the acoustic segment is included in the label string for the segment. We then employ a majority vote scheme to derive the overall cluster labels. When computing the purity of a cluster, we count a cluster member as matching the cluster label as long as the overall cluster label appears in the member's label string. In other words, an acoustic segment overlapping the words "the lighthouse" would receive credit for matching the overall cluster label "lighthouse". A breakdown of the segments captured by the "ocean" cluster and the "castle" cluster is shown in Table 6.1. We investigated some simple schemes for predicting highly pure clusters, and found that the empirical variance of the cluster members (average squared distance to the cluster centroid) was a good indicator. Figure 6-2 displays a scatter plot of cluster purity weighted by the natural log of the cluster size against the empirical variance. Large, pure clusters are easily predicted by their low empirical variance, while a high variance is indicative of a garbage cluster.

Ranking a set of $k = 500$ acoustic clusters by their variance, Table 6.2 displays some statistics for the 50 lowest-variance clusters. We see that most of the clusters

Word	Count	Word	Count
ocean	2150	castle	766
(silence)	127	(silence)	70
the ocean	72	capital	39
blue ocean	29	large castle	24
body ocean	22	castles	23
oceans	16	(noise)	21
ocean water	16	council	13
(noise)	15	stone castle	12
of ocean	14	capitol	10
oceanside	14	old castle	10

Table 6.1: Examples of the breakdown of word/phrase identities of several acoustic clusters

are very large and highly pure, and their labels reflect interesting object categories being identified by the neural network. We additionally compute the coverage of each cluster by counting the total number of instances of the cluster label anywhere in the training data, and then compute what fraction of those instances were captured by the cluster. There are many examples of high coverage clusters, e.g. the “skyscraper” cluster captures 84% of all occurrences of the word “skyscraper”, while the “baseball” cluster captures 86% of all occurrences of the word “baseball”. This is quite impressive given the fact that no conventional speech recognition was employed, and neither the multimodal neural network nor the grounding algorithm had access to the text transcripts of the captions.

To get an idea of the impact of the k parameter as well as a variance-based cluster pruning threshold based on Figure 6-2, we swept k from 250 to 2000 and computed a set of statistics shown in Table 6.3. We compute the standard overall cluster purity evaluation metric in addition to the average coverage across clusters. The table shows the natural tradeoff between cluster purity and redundancy (indicated by the average cluster coverage) as k is increased. In all cases, the variance-based cluster pruning greatly increases both the overall purity and average cluster coverage metrics. We also notice that more unique cluster labels are discovered with a larger k.

Next, we examine the image clusters. Figure 6-3 displays the 9 most central image

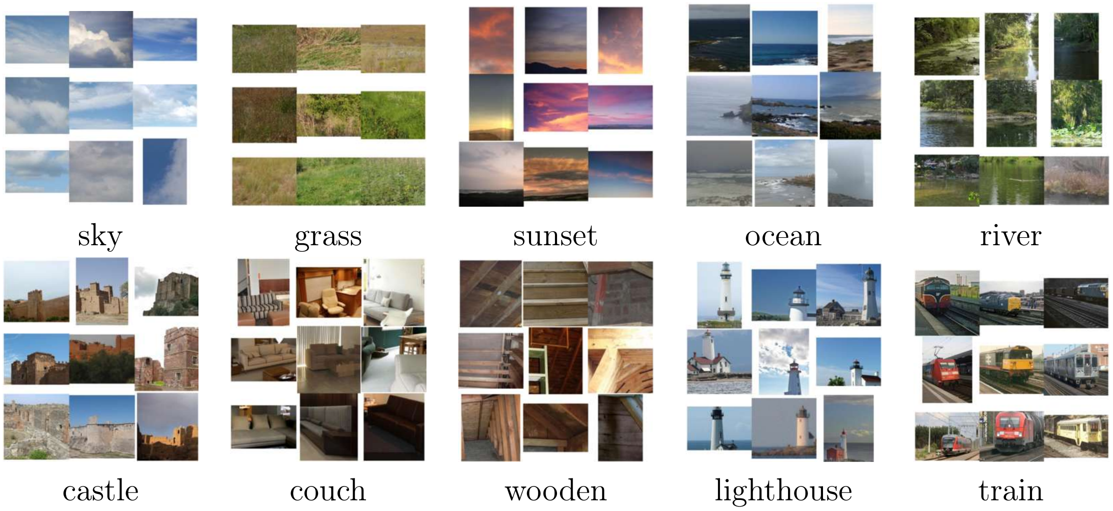


Figure 6-3: The 9 most central image crops from several image clusters, along with the majority-vote label of their most associated acoustic pattern cluster

crops for a set of 10 different image clusters, along with the majority-vote label of each image cluster's associated audio cluster. In all cases, we see that the image crops are highly relevant to their audio cluster label. We include many more example image clusters in Figures A-1 and A-2 in Appendix A.

6.3.2 Meta-Analysis of Audio Cluster Centroids

In order to examine the semantic embedding space in more depth, we took the top 150 clusters from the same $k = 500$ clustering run described in Table 6.2 and performed t-SNE (van der Maaten and Hinton, 2008) analysis on the cluster centroid vectors. We projected each centroid down to 2 dimensions and plotted their majority-vote labels in Figure 6-4. Immediately we see that different clusters which capture the same label closely neighbor one another, indicating that distances in the embedding space do indeed carry information discriminative across word types (and suggesting that a more sophisticated clustering algorithm than k-means would perform better). More interestingly, we see that semantic information is also reflected in these distances. The cluster centroids for "lake," "river," "body," "water," "waterfall," "pond," and "pool" all form a tight meta-cluster, as do "restaurant," "store," "shop," and "shelves," as well as "children," "girl," "woman," and "man." Many other semantic meta-clusters can be

Trans	$\lvert C_c \rvert$	$\lvert C_i \rvert$	Pur.	σ^2	Cov.	Trans	$\lvert C_c \rvert$	$\lvert C_i \rvert$	Pur.	σ^2	Cov.
-	1059	3480	0.70	0.26	-	snow	4331	3480	0.85	0.26	0.45
desert	1936	2896	0.82	0.27	0.67	kitchen	3200	2990	0.88	0.28	0.76
restaurant	1921	2536	0.89	0.29	0.71	mountain	4571	2768	0.86	0.30	0.38
black	4369	2387	0.64	0.30	0.17	skyscraper	843	3205	0.84	0.30	0.84
bridge	1654	2025	0.84	0.30	0.25	tree	5303	3758	0.90	0.30	0.16
castle	1298	2887	0.72	0.31	0.74	bridge	2779	2025	0.81	0.32	0.41
-	2349	2165	0.31	0.33	-	ocean	2913	3505	0.87	0.33	0.71
table	3765	2165	0.94	0.33	0.23	windmill	1458	3752	0.71	0.33	0.76
window	1890	2795	0.85	0.34	0.21	river	2643	3204	0.76	0.35	0.62
water	5868	3204	0.90	0.35	0.27	beach	1897	2964	0.79	0.35	0.64
flower	3906	2587	0.92	0.35	0.67	wall	3158	3636	0.84	0.35	0.23
sky	4306	6055	0.76	0.36	0.34	street	2602	2385	0.86	0.36	0.49
golf course	1678	3864	0.44	0.36	0.63	field	3896	3261	0.74	0.36	0.37
tree	4098	3758	0.89	0.36	0.13	lighthouse	1254	1518	0.61	0.36	0.83
forest	1752	3431	0.80	0.37	0.56	church	2503	3140	0.86	0.37	0.72
people	3624	2275	0.91	0.37	0.14	baseball	2777	1929	0.66	0.37	0.86
field	2603	3922	0.74	0.37	0.25	car	3442	2118	0.79	0.38	0.27
people	4074	2286	0.92	0.38	0.17	shower	1271	2206	0.74	0.38	0.82
people walking	918	2224	0.63	0.38	0.25	wooden	3095	2723	0.63	0.38	0.28
mountain	3464	3239	0.88	0.38	0.29	tree	3676	2393	0.89	0.39	0.11
-	1976	3158	0.28	0.39	-	snow	2521	3480	0.79	0.39	0.24
water	3102	2948	0.90	0.39	0.14	rock	2897	2967	0.76	0.39	0.26
-	2918	3459	0.08	0.39	-	night	3027	3185	0.44	0.39	0.59
station	2063	2083	0.85	0.39	0.62	chair	2589	2288	0.89	0.39	0.22
building	6791	3450	0.89	0.40	0.21	city	2951	3190	0.67	0.40	0.50

Table 6.2: Top 50 clusters with $k = 500$ sorted by increasing variance. Legend: $|C_c|$ is acoustic cluster size, $|C_i|$ is associated image cluster size, Pur. is acoustic cluster purity, σ^2 is acoustic cluster variance, and Cov. is acoustic cluster coverage. A dash (-) indicates a cluster whose majority label is silence.

seen in Figure 6-4, suggesting that the embedding space is capturing information that is highly discriminative both acoustically *and* semantically.

6.3.3 Relation of Learned Clusters to ImageNet Classes

Because our experiments revolve around the discovery of word and object categories, one question to address is the extent to which the supervision used to train the VGG network constrains or influences the kinds of objects learned. Because the 1,000 object classes from the ILSVRC2012 task (Russakovsky et al., 2015) used to train the VGG network were derived from WordNet synsets (Fellbaum, 1998), we can measure the semantic similarity between the words learned by our network and

	$\sigma^2 < 0.9$					$\sigma^2 < 0.65$				
k	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{X}\rvert$	Pur	$\lvert\mathcal{L}\rvert$	AC	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{X}\rvert$	Pur	$\lvert\mathcal{L}\rvert$	AC
250	249	1081514	.364	149	.423	128	548866	.575	108	.463
500	499	1097225	.396	242	.332	278	623159	.591	196	.375
750	749	1101151	.409	308	.406	434	668771	.585	255	.450
1000	999	1103391	.411	373	.336	622	710081	.568	318	.382
1500	1496	1104631	.429	464	.316	971	750162	.566	413	.366
2000	1992	1106418	.431	540	.237	1354	790492	.546	484	.271

Table 6.3: Clustering statistics of the acoustic clusters for various values of k and different settings of the variance-based cluster pruning threshold. Legend: $|\mathcal{C}|$ = number of clusters remaining after pruning, $|\mathcal{X}|$ = number of datapoints after pruning, Pur = purity, $|\mathcal{L}|$ = number of unique cluster labels, AC = average cluster coverage

the ILSVRC2012 class labels by using synset similarity measures within WordNet. We do this by first building a list of the 1,000 WordNet synsets associated with the ILSVRC2012 classes. We then take the set of unique majority-vote labels associated with the discovered word clusters for $k = 500$, filtered by setting a threshold on their variance ($\sigma^2 \leq 0.65$) so as to get rid of garbage clusters, leaving us with 197 unique acoustic cluster labels. We then look up each cluster label in WordNet, and compare all noun senses of the label to every ILSVRC2012 class synset according to the path similarity measure. This measure describes the distance between two synsets in a hyponym/hypernym hierarchy, where a score of 1 represents identity and lower scores indicate less similarity. We retain the highest score between any sense of the cluster label and any ILSVRC2012 synset. Of the 197 unique cluster labels, only 16 had a distance of 1 from any ILSVRC12 class, which would indicate an exact match. A path similarity of 0.5 indicates one degree of separation in the hyponym/hypernym hierarchy - for example, the similarity between “desk” and “table” is 0.5. 47 cluster labels were found to have a similarity of 0.5 to some ILSVRC12 class, leaving 134 cluster labels whose highest similarity to any ILSVRC12 class was less than 0.5. In other words, more than two thirds of the highly pure pattern clusters learned by our network were dissimilar to all of the 1,000 ILSVRC12 classes used to pretrain the VGG network, indicating that our model is able to generalize far beyond the set of classes found in the ILSVRC12 data. We display the labels of the 40 lowest variance acoustic

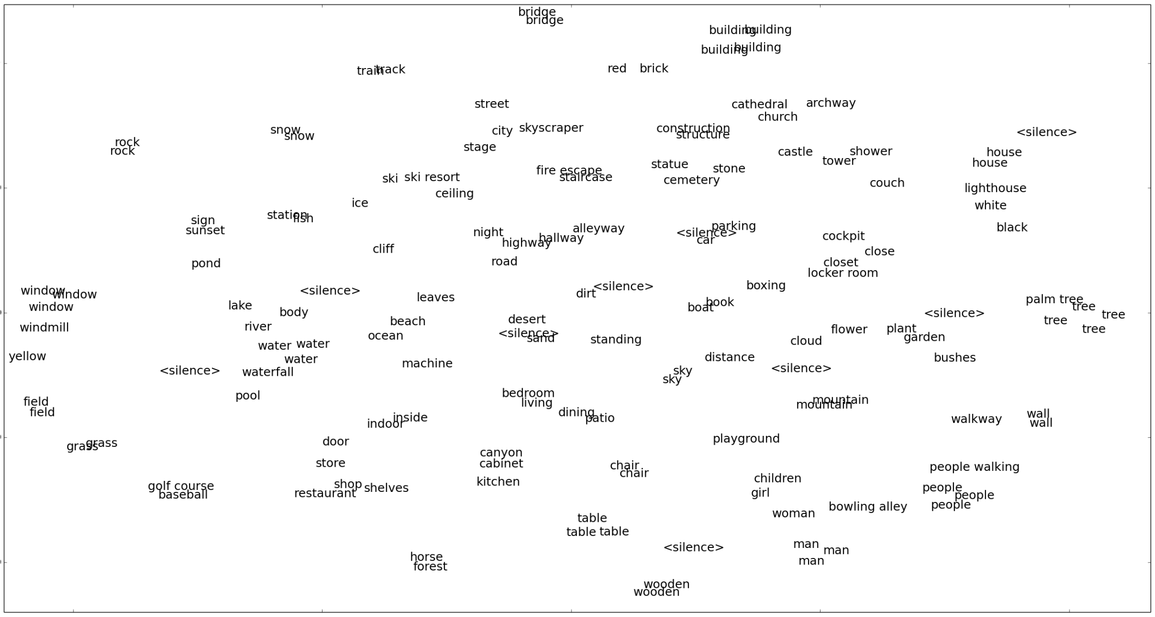


Figure 6-4: t-SNE analysis of the 150 lowest-variance audio pattern cluster centroids for $k = 500$. Displayed is the majority-vote transcription of the each audio cluster. All clusters shown contained a minimum of 583 members and an average of 2482, with an average purity of .668.

clusters labels along with the name and similarity score of their closest ILSVRC12 synset in Table A.1 in Appendix A.

6.4 Pattern Grounding with Matchmap Networks

In the previous section, we demonstrated how coupled sliding window techniques could be applied to images and spectrograms as an input pre-processing step, and then fed into our embedding vector networks (such as ACL17) to produce localized audio-visual semantic groundings. One downside of the sliding window approach is that it requires many thousands of forward passes through the embedding network; a second downside is that it is only capable of producing bounding box groundings, rather than complex shapes. In this section, we will describe an alternative approach which does not have these shortcomings. Instead, we use a matchmap network that accepts the entire image and entire caption spectrogram as inputs, and directly outputs a 3-dimensional tensor representing the semantic similarity between a (row, col) spatial

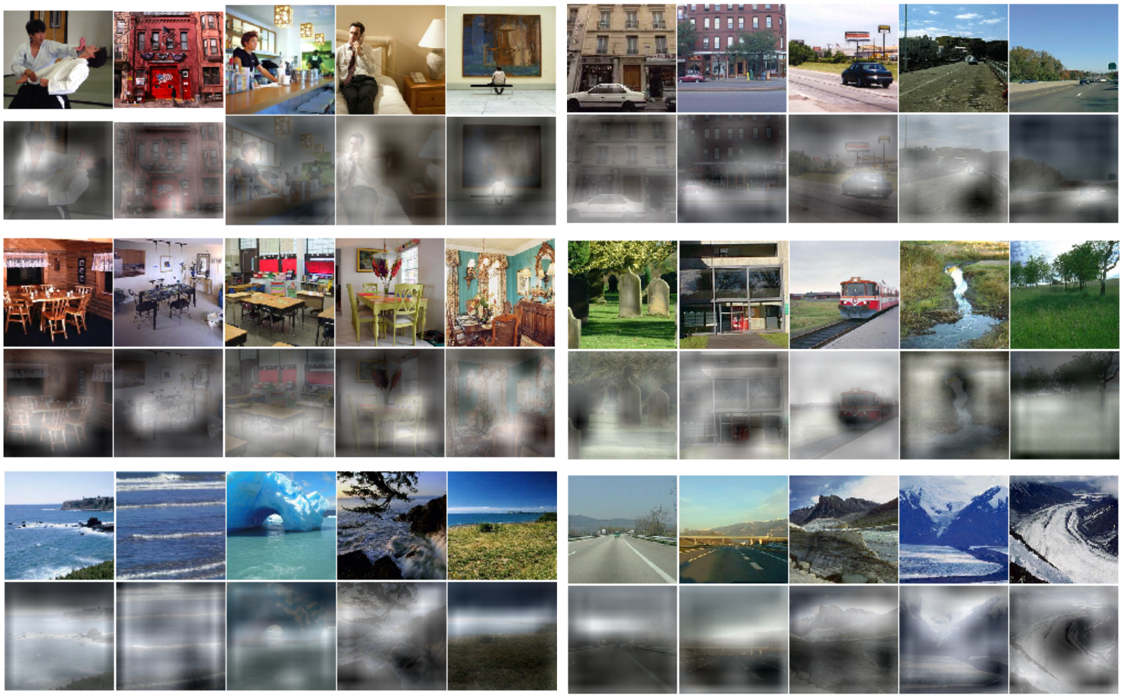


Figure 6-5: Speech-prompted localization maps for several word/object pairs. Reading across from left to right then top to bottom, the queries are instances of the spoken words “MAN,” “CAR,”, “CHAIRS”, “GRASS”, “SEA” and “MOUNTAINS” extracted from each image’s accompanying speech caption.

position within the image and an instant in time t within the caption (the M tensor previously described). Computing this matchmap is therefore is done in a single forward pass through the network. Furthermore, we will demonstrate that meaningful localizations within the matchmaps emerge naturally as a byproduct of training with the ranking-based objective function (Equation 5.4); they do not require any localized labelling or annotation of the training data. Unless otherwise noted, the variant of the Matchmap networks used for the following experiments utilized the MISA scoring function, acoustic input truncation or padding to 1024 frames with no padding compensation, a ReLU audio output and linear image output, a VGG16 image branch pre-trained on ImageNet, and $d = 1024$.

6.4.1 Speech-prompted Object Localization

To evaluate our models' ability to associate spoken words with visual objects in a more fine-grained sense, we use the spoken captions for the ADE20k dataset (Zhou et al., 2017). The ADE20k images contain pixel-level object masks and labels - in conjunction with a time-aligned transcription produced via ASR, we can associate each matchmap cell with a specific visual object label as well as a word label. These labels enable us to analyze which words are being associated with which objects. We do this by performing speech-prompted object localization. Given a word in the speech beginning at time t_1 and ending at time t_2, we derive a heatmap across the image by summing the matchmap between t_1 and t_2. We then normalize the heatmap to sit in the interval [0,1], threshold the heatmap, and evaluate the intersection over union (IoU) of the detection mask with the ADE20k label mask for whatever object was referenced by the word. Because there are a very large number of different words appearing in the speech, and no one-to-one mapping between words and ADE20k objects exists, we manually define a set of 100 word-object pairings. We choose commonly occurring (at least 9 occurrences) pairs that are unambiguous, such as the word "building" and object "building," the word "man" and the "person" object, etc. For each word-object pair, we compute an average IoU score across all instances of the word-object pair appearing together in an ADE20k image and its associated caption. We then average these scores across all 100 word-object pairs and report results for each model type in Table 6.4. We also report the IoU scores for the ASR text-based baseline models described in Section 5.4. Finally, we compare to the IoU score achieved by using the ACL17 and sliding window approach. While we have already demonstrated the effectiveness of the sliding windows for discovering meaningful patterns, the lower IoU scores on this task show the advantages of the matchmap techniques for extracting more accurate object masks. Figure 6-5 displays a sampling of localization heatmaps for several query words.

Sim. Function	Speech IoU	Text IoU
SISA	.2025	.2177
MISA	.2282	.2364
SIMA	.1831	.1975
ACL17 Sliding Window	.1548	-

Table 6.4: Speech-prompted and ASR-prompted object detection and localization IoU scores on the ADE20k data, averaged across the 100 handpicked word-object pairs.

6.4.2 Matchmap Visualizations and Videos

We can visualize the matchmaps in several ways. The 3-dimensional density shown in Figure 5-2 is perhaps the simplest, although it can be difficult to read as a still image. Instead, we can also extract volumetric connected components from the density and simultaneously project them down onto the image and spectrogram axes; visualizations of this are shown in Figures 6-6 and 6-7. For all visualizations, we found it necessary to apply a small amount of post-processing to the raw matchmaps in the form of thresholding and smoothing. The raw matchmaps can appear somewhat fragmented, so we first apply a sliding max-pooling window with a size of 8 frames across the temporal dimension of the raw matchmap. Next, we normalize the matchmap scores to fall within the interval [0, 1] and sum to 1. Finally, we keep only the cells comprising the top p percentage of the total mass within the matchmap, setting all others to zero. In practice, we found that p values between 0.15 and 0.3 produced attractive results.

It is also possible to treat the matchmap as a stack of masks overlayed on top of the image, which can then be played back as a video. By using the matchmap score to modulate the alpha channel of the image across time, it is possible to dynamically highlight the regions of the image being referred to by the audio caption at a specific point in time. To synchronize the streams, the matchmap video playback should be set at 12.5 frames per second so that it temporally aligns with the speech audio playback.

6.4.3 Clustering of Audio-Visual Patterns

The final experiment we consider is automatic discovery of audio-visual clusters from the ADE20k matchmaps. Once a matchmap has been computed for an image and

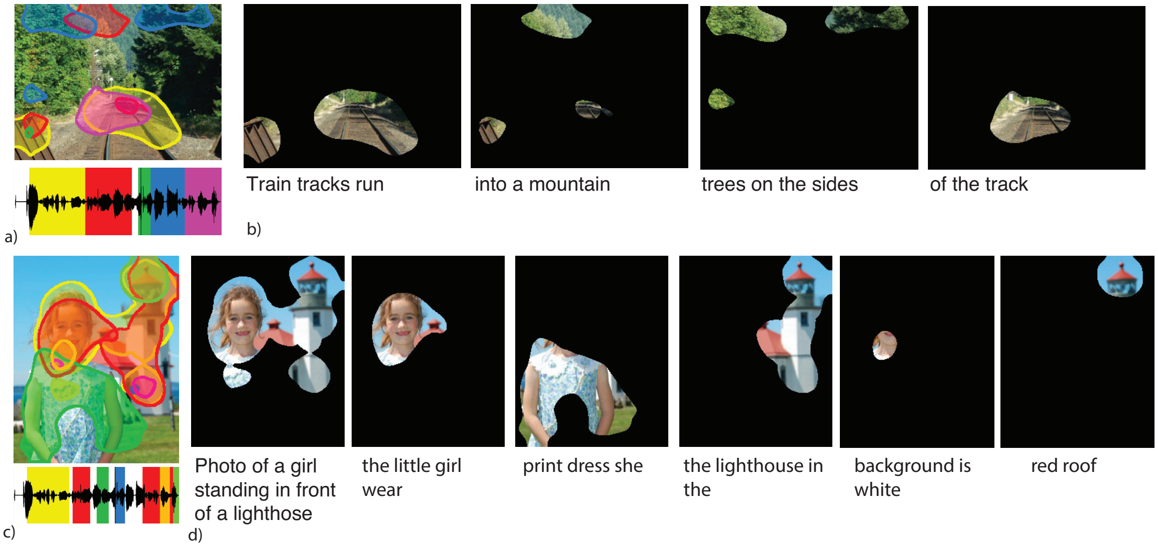


Figure 6-6: Figs (a) and (c) show two images and the speech signal. Each color corresponds to one connected component derived from two matchmaps (only large segments shown). Figs. (b) and (d) show the image segments that correspond to each piece of the speech signal. For clarity, we show at the bottom caption words obtained from transcriptions.

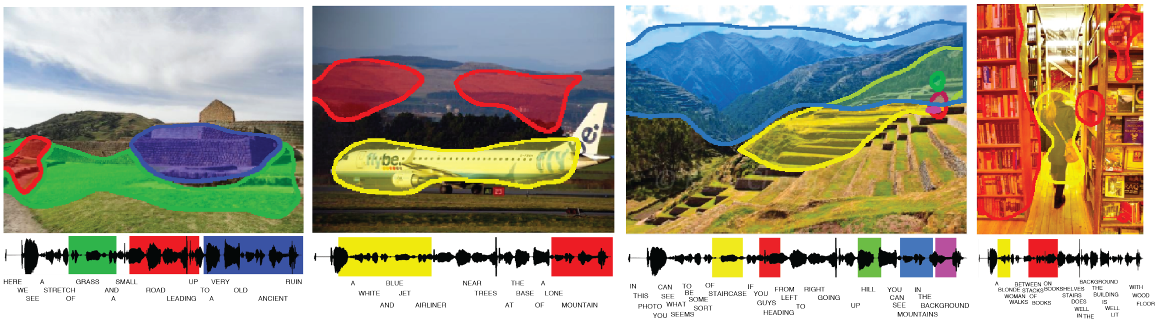

HERE WE SEE A STRETCH OF GRASS AND A SMALL ROAD LEADING UP TO A VERY OLD ANCIENT RUIN
A WHITE AND BLUE JET AIRLINER NEAR TREES AT THE BASE OF A LONE MOUNTAIN
IN THIS PHOTO YOU CAN SEE WHAT SEEMS TO BE SOME SORT OF STAIRCASE IF YOU GUYS HEADING FROM LEFT TO RIGHT GOING UP HILL YOU CAN SEE MOUNTAINS IN THE BACKGROUND
A BLONDE WOMAN WALKS BETWEEN STACKS OF BOOKS ON BOOKSHELVES STAIRS DOES WELL IN THE BACKGROUND THE BUILDING IS WELL LIT WITH WOOD FLOOR

Figure 6-7: Additional examples of discovered image segments and speech fragments.

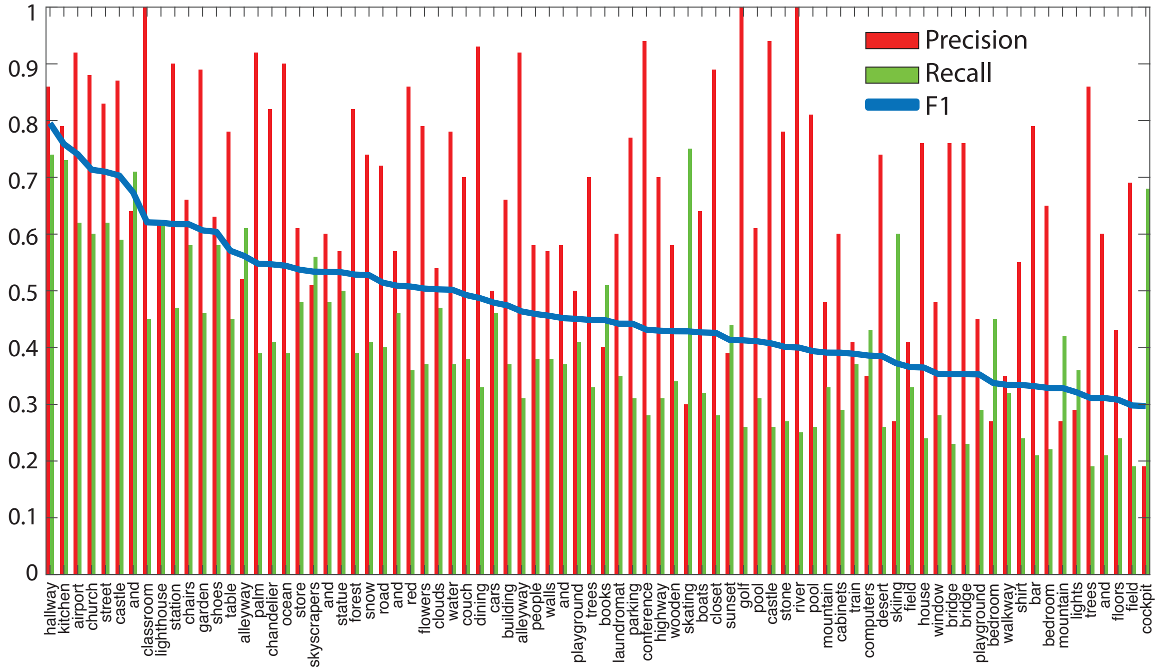


Figure 6-8: Clusters (speech and visual) found by our approach. Each cluster is labeled with the most common word. For each word we show precision (red), recall (green) and F1. For the clusters with $F1 > 0.5$ there are 28 different words discovered in ADE20k.

caption pair, we smooth it with a max pooling window of size 8 across the temporal dimension before binarizing it according to a threshold. In practice, we set this threshold on a matchmap-specific basis to be 1.5 standard deviations above the mean value of the smoothed matchmap. Next, we extract volumetric connected components and their associated masks over the image and audio. We average pool the image and audio feature maps within these masks, producing a pair of vectors for each component. Because we found the image and speech representations to exhibit different dynamic ranges, we first rescale them by the average L2 norms across all derived image vectors and speech vectors, respectively. We concatenate the image and speech vectors for each component, and finally perform Birch clustering (Zhang et al., 1996) with 1000 target clusters for the first step, and an agglomerative final step that resulted in 149 clusters.

To derive a label for each cluster, we first compute a precision for each word as the fraction of components assigned to the cluster whose audio masks overlapped (perhaps partially) with an instance of the word in question. We then compute the recall for each word by dividing this number of occurrences by the total number of times the word appeared anywhere in the ADE20k captions. Taking the harmonic mean of the precision and recall results in an F1 score, which we use to rank the words in each cluster, taking the top word label to represent the cluster.

Over the 149 clusters, we found an average F1 score of .323. Figure 6-8 displays a plot of the top clusters and their scores, which we can use to roughly gauge how many and what kinds of concepts are being learned by our models. Of course, the number of words and concepts reflected here are specific to the ADE20k dataset which is dramatically smaller than the full training set; therefore Figure 6-8 should be taken as an underestimate of what the model has learned.

6.5 Chapter Summary

In this chapter, we first described an algorithm for extraction of audio-visual groundings from the images and spoken captions using coupled sliding windows. We showed that these groundings could be clustered with high purity in terms of their underlying lexical/semantic content. We performed experiments which showed that semantic relationships between the words learned by our networks are reflected by vector distances within the multimodal embedding space. By comparing the words learned by our network to the ImageNet training labels, we found that the concepts discovered by our models generalize far beyond the label set used to pre-train the image branch of our networks.

Then, we explored the use of our matchmap neural networks for directly learning the semantic correspondences between speech frames and image pixels. We did this by evaluating their ability to perform speech-prompted object localization, audio-visual pattern discovery, and real-time, speech-driven, semantic highlighting.

In the next chapter, we generalize our networks beyond the English language, and consider the case in which we have captions from two different languages describing the same set of images. We will show that multilingual learning not only helps retrieval performance on both languages, but also enables the model to learn word-level and phrase-level translations between the languages, by using the visual domain as an interlingua.

Chapter 7

Cross-Lingual Audio-Visual Modeling

In the previous three chapters, we presented models and experiments that explored the learning of semantic correspondences between speech and visual images. We showed that learning a very rich shared embedding space is indeed possible, and can be done without any conventional supervised speech recognition models or even any text transcriptions. This suggests that our approach should be language agnostic; however, thus far our experiments have only been performed on English speech. In this chapter, we present evidence that our approach is indeed language agnostic by applying it to Hindi speech. We also demonstrate that the visual space can act as an interlingua for cross-lingual speech-to-speech retrieval, suggesting that visual grounding may be helpful for automatic speech-to-speech translation.

7.1 Speech to Speech Translation

A classic science fiction depiction of speech to speech translation is the Babel Fish from Douglas Adams' *The Hitchhiker's Guide to the Galaxy.* This small fish forms a symbiotic relationship with its host by living in their ear and telepathically translating any spoken language in the universe. While the Babel Fish is clearly outside the realm of plausibility, it is hard not to notice its parallels with modern speech-to-speech translation technologies such as Microsoft Translator or Skype Translator (Lewis, 2015). The current state-of-the-art in speech-to-speech translation is a highly engineered

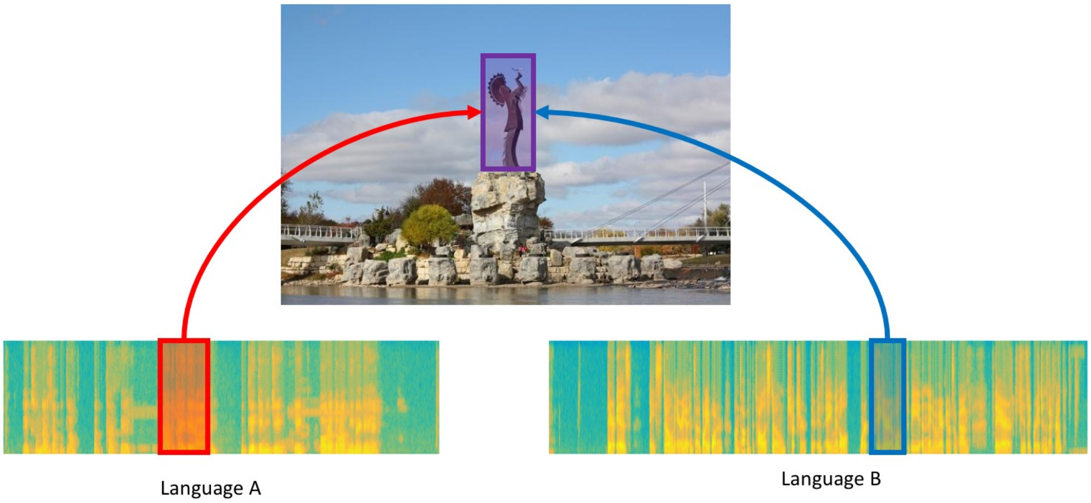


Figure 7-1: A motivating example of how the visual domain might be used as an interlingua between multiple languages.

approach that relies on first performing conventional ASR on the source speech, text-based MT to generate a translation in the target language, and Text-To-Speech (TTS) synthesis to generate the output speech in the target language. All of these constituent technologies are very resource-needy, and moreover creating an MT model between each pair in a set of N languages would require $O(N^2)$ parallel translation corpora. Here, we ask whether the visual domain can act as a kind of "Rosetta Stone" for cross-lingual learning (Figure 7-1). If cross-lingual semantics could be learned without directly parallel translation corpora, and instead be learned via description or narration of a common set of images or videos, then this would dramatically reduce the data cost of creating massively multilingual systems. We present preliminary experiments in this chapter which show that our audio-visual models may hold promise in this regard.

7.2 Experimental Data

We make use of the Places English and Hindi audio caption datasets (detailed in Chapter 3). We only use the subset of the English data for which we also have a Hindi caption, resulting in a total of 85,480 triples (image, English caption, Hindi caption). We divide this data into a training set of 84,480 image/caption triplets, and a validation set of 1,000 triplets. This set of English captions on average contain 19.3 words and have an average duration of 9.5 seconds, while the Hindi captions contain an average of 20.4 words and have an average duration of 11.4 seconds.

7.3 Models

Let our dataset be represented by N triples, (I_i, A_i^E, A_i^H), where I_i is the i^{th} image, A_i^E is the acoustic waveform of the English caption describing the image, and A_i^H is the acoustic waveform of the Hindi caption describing the same image. We consider a functional mapping $F(I_i, A_i^E, A_j^H) \mapsto (e_i^I, e_i^E, e_i^H)$ where $e_i^I, e_i^E, e_i^H \in \mathcal{R}^d$; in other words, a mapping of the image and acoustic captions to vectors in a shared, high-dimensional embedding space. Within this space, our hope is that visual-linguistic semantics are manifested as arithmetic relationships between vectors, which enables applications such as semantic retrieval. We implement this mapping using the CNN model architectures described in Chapter 5, but with three networks rather than two: one responsible for embedding the image, one for the English caption, and one for the Hindi caption.

We apply the Matchmap-VGG16 variant of the image network architecture and the Matchmap audio network architecture, with a shared embedding dimension of $d = 2048$ (which we found was more amenable to cross-lingual learning). We use the same data pre-processing steps outlined in Chapter 5, with the spectrogram size fixed at 1024 frames. However, for the experiments in this chapter we do not use the Matchmap-based similarity scoring functions. Instead, we apply global mean pooling to the outputs of the image and audio networks to derive a trio of 2048 dimensional embedding vectors.

7.4 Experiments

7.4.1 Model Training Procedure

The objective functions we use to train our models are all based upon the same margin ranking criterion used throughout this thesis (Bromley et al., 1994). However, in this chapter we generalize beyond audio-to-visual matching and consider audio-to-audio matching as well. We define a more general form of the margin ranking objective function:

$$\text{rank}(a, p, i) = \max(0, \eta - s(a, p) + s(a, i)) \tag{7.1}$$

where a is the anchor vector, p is a vector "paired" with the anchor vector, i is an "imposter" vector, $s()$ denotes a similarity function, and η is the margin hyperparameter. For a (a, p, i) triplet, the loss is zero when the similarity between a and p is at least η greater than the similarity between a and i; otherwise, a loss proportional to $s(a, i)$ is incurred. This objective function therefore encourages the anchor and its paired vector to be "close together," and the the anchor to be "far away" from the imposter. In all of our experiments, we fix $\eta = 1$ and let $s(x, y) = x^T y$

Given that we have images, English captions, and Hindi captions, we can apply the margin ranking criterion to their neural embedding vectors 6 different ways: each input type can serve as either the anchor point, or as the paired and imposter points. For example, an image embedding may serve as the anchor point, its associated English caption would be the paired point, and an unrelated English caption for some other image would be the imposter point. We can even form composite objective functions by performing multiple kinds of ranking simultaneously. We consider several different training scenarios:

1. English ↔ Image
2. Hindi ↔ Image
3. English ↔ Hindi

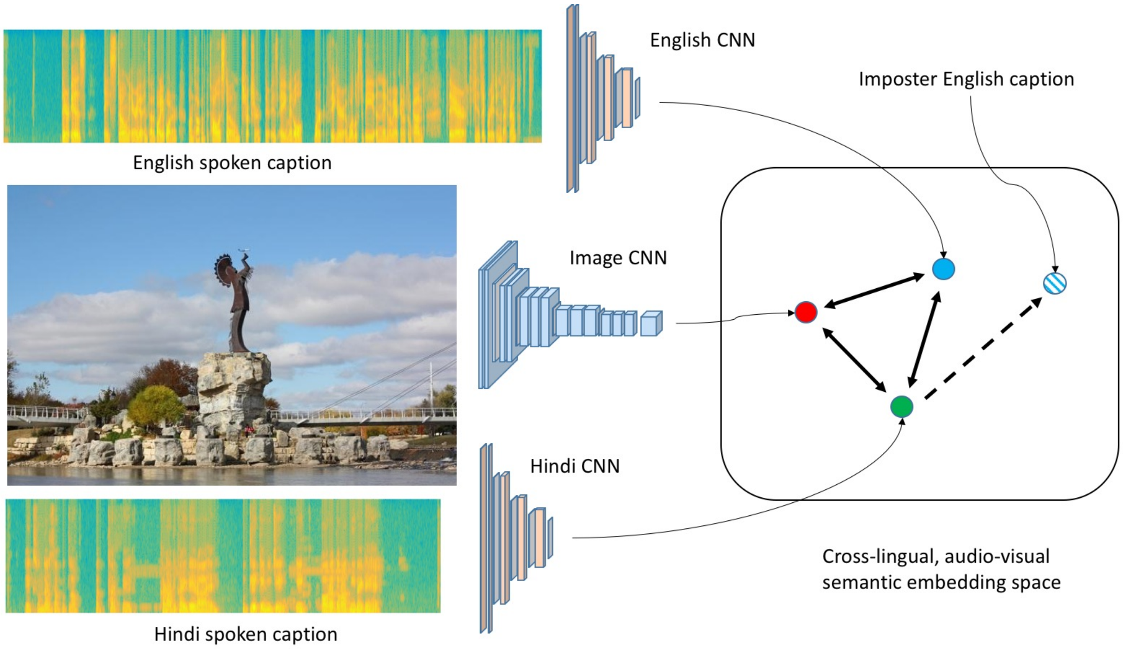


Figure 7-2: Illustration of how images, English captions, and Hindi captions are embedded into a shared space by our models. The triangle of solid black double arrows represent the 6 possible directions of retrieval. An example of the margin ranking loss is shown with the embedded Hindi caption as the anchor point, its paired English caption as the pair point (solid blue circle) and a randomly selected English caption as the imposter point (dashed blue circle pointed to by dashed arrow). The objective function attempts to force the imposter caption to be less similar to the anchor caption than the paired caption. This can also be viewed in the context of the retrieval task, in which the solid blue English caption competes against the dashed blue caption when the solid green Hindi caption is submitted as a query.

4. English ↔ Image ↔ Hindi
5. Hindi ↔ English ↔ Image ↔ Hindi

In each scenario, ↔ denotes a bidirectional application of the ranking loss function to every tuple within a minibatch of size B, e.g. "English ↔ Image" indicates that the terms $\sum_{j=1}^{B} \text{rank}(e_j^I, e_j^E, e_k^E)$ and $\sum_{j=1}^{B} \text{rank}(e_j^E, e_j^I, e_l^I)$ are added to the overall loss, where $k \neq j$ and $l \neq j$ are randomly sampled indices within a minibatch. This is similar to the criteria used in Gella et al. (2017) for multilingual image/text retrieval, except we randomly sample only a single imposter per (a, p) pair. An illustrative example of the embedding and retrieval framework is displayed in Figure 7-2.

We trained all models with stochastic gradient descent using a batch size of 128 images with their corresponding captions. All models except the audio-to-audio (no image) were trained with the same learning rate of 0.001, decreased by a factor of 10 every 30 epochs. The audio-to-audio network used an initial learning rate of 0.01, which resulted in instability for the other scenarios. We divided training into two "rounds" of 90 epochs (for a total of 180 epochs), where the learning rate is reset back to its initial value starting at epoch 91, and then allowed to decay again. We found this schedule achieved better performance than a single round of 90 epochs, especially for the training scenarios involving simultaneous audio/image and audio/audio retrieval.

7.4.2 Evaluation: Audio-Visual and Audio-Audio Retrieval

To evaluate our models numerically, we turn again to the audio-visual retrieval task, but with a new twist. We also evaluate direct audio-to-audio retrieval between English and Hindi, which we view as a weak form of speech-to-speech translation. We therefore explain the retrieval task in more general terms here. Imagine that we have a library L of M target vectors, $L = t_1, t_2, \ldots, t_M$. Now assume that we are given a query vector q which is known to be associated with one of the target vectors, but we do not know exactly which one; our goal is to retrieve this target from the library. Given a similarity function $s(q, t)$ (defined as $s(q, t) = q^T t$ in our experiments), we rank all of the target vectors in descending order of their similarity to q, and retrieve the top scoring 1, 5, and 10 target vectors. If the correct target vector is in the retrieved set, a hit is counted; otherwise, we count the result as a miss. With a set of query vectors covering the entire library (that is, a set of M vectors containing every target vector's associated query vector), we can compute recall scores over the entire query set for each retrieval set size. Recall that the five training scenarios detailed above consider 6 distinct pairwise directions of ranking during training; for example, we can consider the case in which an English caption is the query and its associated image is the target and vice-versa, or the case in which a Hindi caption is the query and the English caption associated with the same underlying image is the target. We apply the retrieval evaluation task to those same directions, and for each model report the relevant recall at the top 1, 5, and 10 returned results.

7.4.3 Experimental Results and Discussion

Audio-visual retrieval recall scores on the 1,000 exemplar validation set are displayed in Table 7.1, while Table 7.2 displays the audio-audio results. In the tables, "English caption" is abbreviated as E, "Hindi caption" as H, and "Image" as I. All models were trained with two consecutive rounds of 90 epochs, though in all cases they converged before epoch 180. Random chance recall scores for all cases are $R@1 = .001$, $R@5 = .005$, $R@10 = .01$. We found that a small amount of relative weighting was necessary for the H↔E↔I↔H loss function in order to prevent the training from completely favoring audio/image or audio/audio ranking over the other; weighting the E↔H ranking loss 5 times higher than that of the E↔I and H↔I losses produced good results. In all cases, the model trained with the H↔E↔I↔H loss function is the top performer by a significant margin. This suggests that the additional constraint offered by having two separate linguistic accounts of an image's visual semantics can improve the learned representations, even across languages. However, the fact that the E↔I↔H model offered only marginal improvements over the E↔I and H↔I models suggests that to take advantage of this additional constraint, it is necessary to enforce semantic similarity between the captions associated with a given image.

	E → I			I → E			H → I			I → H		
Model	1	5	10	1	5	10	1	5	10	1	R5	10
E↔I	.065	.236	.367	.086	.222	.343	-	-	-	-	-	-
H↔I	-	-	-	-	-	-	.061	.185	.303	.064	.186	.277
E↔I↔H	.062	.248	.360	.077	.247	.350	.066	.205	.307	.078	.208	.306
H↔E↔I↔H	.083	.282	.424	.080	.252	.365	.080	.25	.356	.074	.235	.354

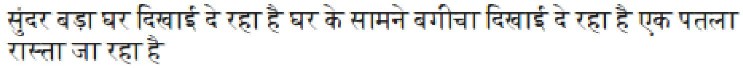
Table 7.1: Summary of audio-visual retrieval recall scores for English and Hindi monolingual and multilingual models.

Perhaps most interesting are our results on cross-lingual speech-to-speech retrieval. We were surprised to find that the E↔H model was able to work at all, given that the retrieval was performed directly on the acoustic level without any linguistic supervision. Even more surprising was the finding that the addition of visual context by the H↔E↔I↔H model approximately doubled the audio-to-audio recall scores

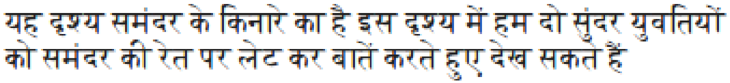
	E → H			H → E		
Model	1	5	10	1	5	10
E↔H	.011	.042	.075	.013	.059	.104
E↔I↔H	.005	.012	.018	.004	.016	.027
H↔E↔I↔H	.034	.114	.182	.033	.121	.203

Table 7.2: Summary of audio-audio retrieval recall scores for English and Hindi. Even though the E↔I↔H configuration is not specifically trained for the English/Hindi audio-to-audio retrieval tasks, we perform the evaluation anyway for the sake of comparison.

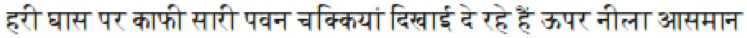
across the board, as compared to the E↔H model. This suggests that the information contained within the visual modality provides a strong semantic grounding signal that can act as an "interlingua" for cross-lingual learning. Three randomly selected examples of audio-to-audio retrieval are shown below: first the text transcriptions of three Hindi captions, followed by the transcriptions of their top-1 retrieved English captions using the H↔E↔I↔H model. The English result is denoted by "E:", and the approximate Hindi translation of the query is denoted by "HT:". Note that the model has no knowledge of any of the ASR text shown below; the text is strictly for analysis purposes.


सुंदर बड़ा घर दिखाई दे रहा है घर के सामने बगीचा दिखाई दे रहा है एक पतला
रास्ता जा रहा है

HT: "There is big beautiful house. There is a garden in front of the house. There is a slender road"
E:"A small house with a stone chimney and a porch"


यह दृश्य समंदर के किनारे का है इस दृश्य में हम दो सुंदर युवतियों
को समंदर की रेत पर लेट कर बातें करते हुए देख सकते हैं

HT: "This is a picture next to the seashore. Two beautiful girls are laying on the sand, talking to each other"
E:"A sandy beach and the entrance to the ocean the detail in the sky is very vivid"


हरी घास पर काफी सारी पवन चक्कियां दिखाई दे रहे हैं ऊपर नीला आसमान

HT: "There are many windmills on the green grass"
E:"There is a large windmill in a field"

7.4.4 Analysis of Audio-to-Audio Matchmaps

Our audio-to-audio caption retrieval results indicate that our models are learning some form of cross-lingual semantics between English and Hindi. A natural question to consider is whether *localized* regions of cross-lingual semantic similarity can be recovered, and whether the underlying spoken words contained within these intervals constitute reasonable word-level (or phrase-level) translations. Removing the final mean pooling layers from the audio networks enables us to compute a 2-dimensional audio Matchmap (or more simply, a similarity matrix) between the English and Hindi captions (belonging to the same underlying image), in a manner similar to how we computed audio-visual Matchmaps in Chapter 5. Figure 7-6 depicts an example of this; assuming that the output of the English audio network is a matrix of size (N_E, d) and the output of the Hindi audio network is a matrix of size (N_H, d), we take the dot product of each English frame with each Hindi frame to produce a matrix of size (N_E, N_H).

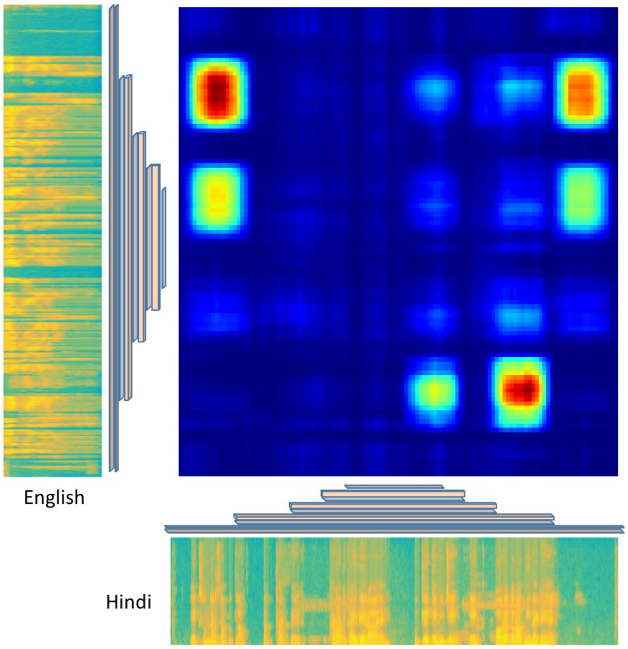
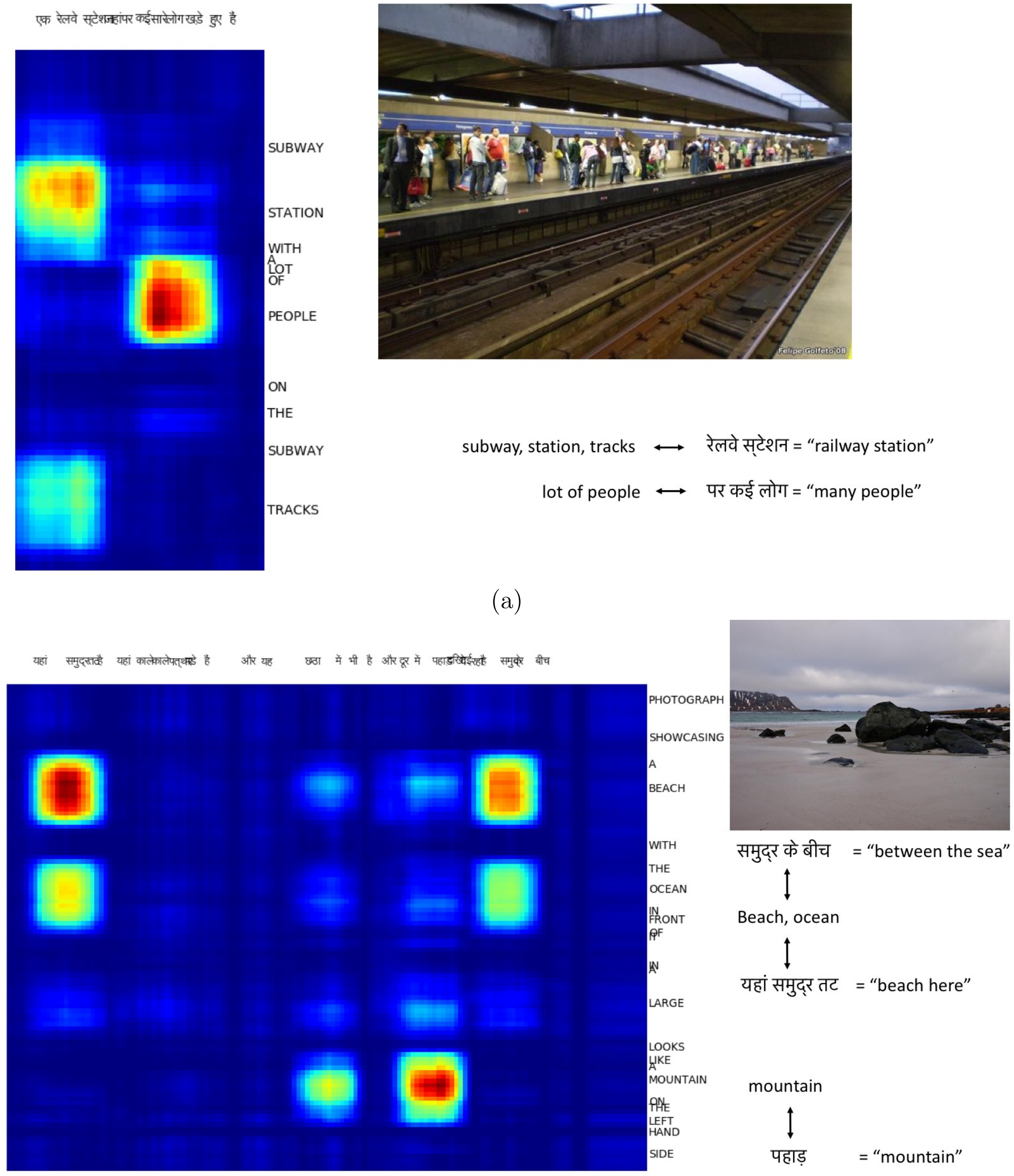


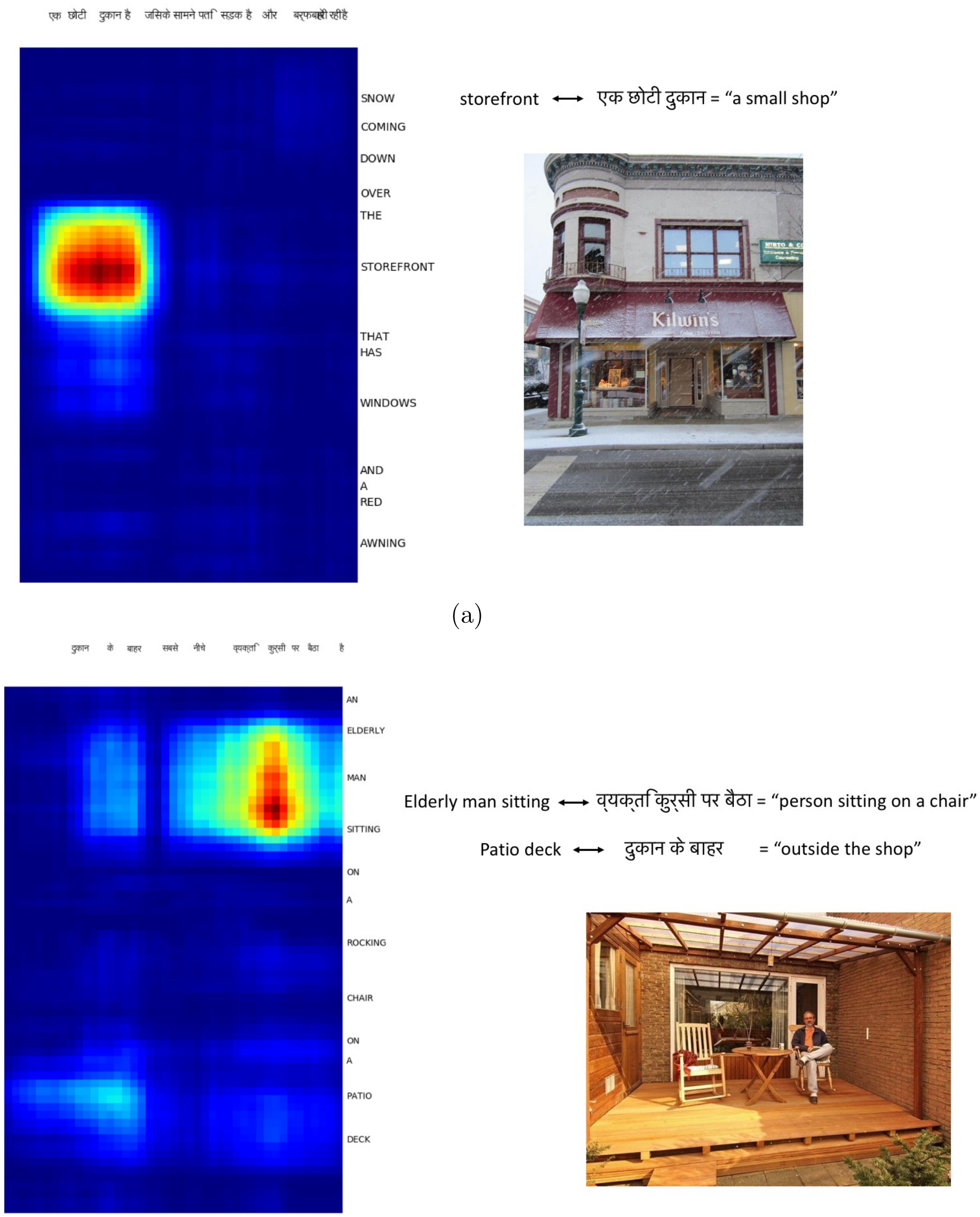
Figure 7-6: Example of how speech-to-speech matchmaps are derived from our models. The element at location (i, j) in the matchmap matrix reflects the similarity score (e.g. dot product) between the i^{th} frame of the English network's output and the j^{th} frame from the Hindi network's output. Segments with high similarity can be spotted visually in the resulting matchmap.



(a)

(b)

Figure 7-7: Two examples of audio-to-audio matchmaps for English and Hindi captions describing the same underlying image.



(a)

(b)

Figure 7-8: Two examples of audio-to-audio matchmaps for English and Hindi captions describing the same underlying image.

Visual inspection of these matrices reveals regions of low similarity (blue) and regions of high similarity (red). Figures 7-7 and 7-8 display a series of these Matchmaps with the force-aligned text of the captions displayed along the edges of each Matchmap.

Alongside each example, we manually extract the words underlying each high similarity region, along with an approximate English translation of the Hindi text. In nearly all cases, the high similarity regions do in fact reflect semantic translations between individual words (as well as short phrases). We have not yet performed a large-scale analysis of the quality of these translation alignments. However, we believe that it should be possible to automatically extract them for that purpose, which we plan to pursue in the near future.

7.5 Chapter Summary

In this chapter, we applied our audio-visual association models to both English and Hindi captions, along with their paired visual images. The successful application to Hindi provides evidence of the language-agnosticism of these models. We also showed that multilingual variants of our models can outperform their monolingual counterparts for speech/image association. Finally, we performed experiments on direct audio-to-audio retrieval between Hindi and English, suggesting that a shared visual context can contribute dramatically to the learning of cross-lingual semantics. Future experiments should analyze whether any sort of alignment can be inferred between the English and Hindi speech, and if these alignments correspond to word or phrase-level translations. We believe that the approaches presented in this work are a promising early step towards speech-to-speech translation models that would not require any form of annotation beyond asking speakers to provide narrations of images, videos, etc. in their native language.

The next and final chapter summarizes the contributions of this thesis, enumerates topics for future work, and offers a closing statement.

Chapter 8

Conclusions

8.1 Thesis Summary

This thesis has chronicled the story of our investigations into joint modeling of speech audio and visual images. Chapter 4 documented our exploratory efforts and presented a proof-of-concept model capable of aligning pre-segmented audio and images. Encouraged by those results, we embarked on the collection of a far larger dataset and developed more refined models that did not require pre-segmentation of either input modality. This work was detailed in Chapters 5 and 6, and represents the first published successful efforts for unsupervised speech-to-image learning directly at the waveform level. Finally, in Chapter 7 we demonstrated the language agnosticism of our modeling approach. We collect a second set of spoken captions in Hindi and successfully applied our audio-visual models to them. We then went beyond monolingual modeling and showed that multilingual models could achieve superior performance for image/caption retrieval tasks. Finally, we presented experimental evidence that cross-modal learning could dramatically improve cross-lingual learning.

8.2 Thesis Contributions

We reiterate a summary of the contributions made by this thesis here:

1. **Introduction of models capable of mapping complex visual images and**

unsegmented, continuous speech into a shared, semantic vector space. We introduce a more advanced modeling framework based on deep convolutional neural networks that is capable of learning the semantic association between unsegmented images and their spoken captions. We show that these models can embed entire image frames and entire spoken captions as fixed points in a high dimensional, multimodal vector space. In this space, semantic relationships are preserved via vector operations such as the inner product. This enables high-level semantic similarity between image scenes and their captions to be computed via vector operations in the embedding space, which we utilize to perform semantic image search from spoken queries.

2. **Demonstration that the internal representations learned by the models recognize and associate individual words and objects.** We explore two distinct ways of extracting localized segments containing word-like units and object-like image regions: 1) using coupled sliding windows imposed upon the input, and 2) extracting connected components from 3-dimensional spatial-temporal association maps derived from the neural model's internal feature maps. We demonstrate that in both cases, the extracted patterns can be grouped into very pure clusters using simple algorithms, suggesting that the representations learned by the networks capture a significant amount of high-level linguistic abstraction.

3. **Demonstration of the language-agnostic nature of the models.** Using an additional spoken caption dataset collected in Hindi, we train a set of audio-visual association networks. We show that the caption-to-image (and vice versa) retrieval scores achieved by the Hindi model are close to those achieved with a similarly sized English dataset, suggesting that our approach is indeed language agnostic.

4. **Demonstration of the models' ability to learn cross-lingual semantics.** In addition to training Hindi-language variants of the audio-visual association models originally trained on English, we train a *triplet* model that utilizes a

shared image model in conjunction with an English speech model and a Hindi speech model. We demonstrate that such a network can not only perform image/caption retrieval in either language alone, but also can retrieve the Hindi caption associated with the image associated with an English query caption (and vice versa). While the cross-lingual speech-to-speech retrieval scores we achieve are lower than the speech-to-image and image-to-speech scores, they are many times better than chance and suggest a promising new direction for speech-to-speech translation research.

5. **Collection of a very large, multilingual spoken caption dataset.** We collected 40,000 English captions for the Flickr 8k dataset (Rashtchian et al., 2010), over 400,000 English captions for the Places 205 dataset (Zhou et al., 2014), nearly 10,000English captions for the ADE20k dataset (Zhou et al., 2017), and nearly 100,000 Hindi captions for the Places 205 dataset.

8.3 Future Directions

The work presented in this thesis is highly exploratory, and thus the possibilities for future work are incredibly fertile. We enumerate several promising directions here.

8.3.1 Image and speech synthesis

Given that our models are able to associate natural image scenes with spoken audio captions describing those scenes, a natural follow-up question to ask is whether there might be a way to actually *generate* an image given a spoken description, or vice-versa. It has already been shown to be possible to generate images from a text caption (Reed et al., 2016), so generating images from spoken audio captions may not be out of the question. Going in the other direction, it may even be possible to directly generate spoken audio captions given a visual input.

8.3.2 Robotics

Robots represent a computational engine coupled to physical hardware enabling them to interact with the real world in a meaningful way. They are therefore by nature multimodal devices, and a natural application of the work presented in this thesis. A robot with microphones and cameras would be sufficiently equipped to apply our models and algorithms as-is; however, many robots also utilize sensory inputs beyond sound and vision, such as touch sensors. These additional inputs could conceivably add even more contextual richness to the modeling framework. Perhaps the most compelling aspect of incorporating our technology into robotic systems is the robot's potential for movement and environmental interaction. Visually grounded navigation ("Follow the red brick road until you reach a grove of four trees and then turn left.") and referential commands ("Pick up the brown box next to the door and put it down on top of the green shelf on the other side of the room.") are perfect examples of this.

8.3.3 Fusion with other forms of language

An interesting application to consider is how our models might be applied for a conventional ASR task. Given that the models learn quite rich linguistic representations from the audio-visual training scheme, it is worth considering whether a large portion of that information could be transferred to a supervised task such as ASR. Were that the case, it is not unreasonable to assume that such a model might require less transcribed training data than a conventional model, since it has already learned relatively robust and invariant acoustic representations of speech.

While making the link with digitized text would enable ASR, learning from images of printed text may enable our models to perform optical character recognition (OCR), or even to learn human handwriting. In this same spirit, videos depicting sign language accompanied by spoken translations could be used to learn correspondences between spoken and signed words.

8.3.4 Speech-to-speech translation

While only dealing with speech retrieval and not true translation, Chapter 7 provided evidence that the common contextual grounding offered by the visual domain can act as an effective interlingua between two languages. If these ideas could scale, it is conceivable that a true speech-to-speech translation system could be developed without the need for any text transcriptions or manually aligned corpora whatsoever. The implications here are enormous; current translation datasets rely upon bilingual humans to provide gold-standard direct translations from text in the source language to text in the target language. There are many languages worldwide with less than a million speakers (Lewis et al., 2016), and many which do not even have a stable orthography. Expert bilingual translators for every possible language pair are in short supply, and thus their services are expensive. If it were possible to create a machine translation engine that only relied on narrations of images, videos, etc. in each language it was designed to handle, it would completely alleviate the need for manual translations and present an enormous technological breakthrough. While a speech-based system would also do away with the need for text transcriptions all-together, the idea of visually grounding translations could conceivably be applied to text-based MT systems as well.

8.3.5 Bilingual to many-lingual

In Chapter 7, we demonstrated that the addition of a second language to our audio-visual association models offered performance improvements over a monolingual model; it is conceivable that an N-lingual model might offer additional gains. We also believe that our methods have the potential to enrich "hub and spoke" models of machine translation that rely upon an interlingual representation to translate between language pairs for which no training data is available. When a resource-rich language, such as English, is used as a translation interlingua, there is an opportunity for linguistic subtleties to be lost. It is possible that linguistic representations enriched with multimodal semantics may be able to better preserve these subtleties.

8.3.6 Images to videos

All of the work we have done thus far has utilized still frame images as the visual context. However, still frames do not capture the full scope of the world, as they cannot show dynamics or movement. The pattern clusters detailed in Chapter 5 contain mostly nouns, a few adjectives, and few verbs or adverbs. If we desire a complete characterization of language, then one logical route to learning verbs and adverbs is to model videos depicting actions.

Videos also represent a vast ocean of pre-existing data that could be exploited for learning. Movies, television broadcasts, YouTube videos, and so on generally contain soundtracks that are in some way related to their visual content. The nature of this relation is varied: during a cooking show, often times the host describes exactly what they are doing as they are doing it. In an action movie, the environmental sounds of car engines and explosions may dominate. In both cases, however, meaningful correspondences exist between the modalities.

8.3.7 New tasks to learn different aspects of language

In the same way that videos may enable our models to learn different parts of speech, such as verbs and adverbs, it is also worth considering whether other forms of spoken narration would lead to richer representations of language in other ways. For example, spatial relations do not appear to be well-modeled by our current scheme. An appropriate dataset that places a focus on object relations (along with an appropriate model and objective function) may be one way to tackle this. Datasets such as CLEVR (Johnson et al., 2017) may be able to facilitate this kind of learning.

8.3.8 Generalization to modalities beyond vision and speech

There is no reason why our models need be limited only to speech and still-frame images. The audio modality carries far more information than just language, including environmental sounds, which could likely be modeled via the techniques presented in this thesis. Beyond that, there are many more sensory modalities which might be merged together to learn even more holistic representations of the real world. For example, in a robotics application, input from touch sensors could be correlated with audio and visual information.

8.3.9 Fully-segmented models for end-to-end pattern discovery

The first methodology for audio-visual pattern discovery detailed in Chapter 5 took the form of a post-processing algorithm to extract clusterable localizations from an already-trained non-segmental model. We then attempted to push these models further towards an end-to-end style architecture, an approach which is gaining popularity in the deep learning community. By end-to-end, we mean that the model would completely encapsulate all of the computation necessary to produce a desired output, without any model factorization via system blocks or any post-processing steps. In our case, one possible desired output would be a full segmentation and labelling of all speech audio and all visual images within our dataset. Even the matchmap models from Chapter 5 do not achieve this, as they rely on a connected component analysis of their thresholded outputs followed up by a conventional clustering step. What makes segmentation and clustering difficult in an end-to-end neural architecture is the fact that boundary variables and clustering assignments are by nature discrete, and therefore non-differentiable. Because all of the widely used training algorithms for deep neural networks are variants of gradient-based hill climbing algorithms, non-differentiability presents a serious hurdle for training. One possibile route around this problem is to allow the network to make discrete decisions during the forward pass, but to use a differentiable surrogate function during the backward pass. A good example of this idea applied to word segmentation in character strings was presented in (Chung et al., 2017); similar ideas could potentially be applied to spectrograms for our purposes. What makes this approach appealing is the fact that the model would output a complete characterization of its inputs. Currently, our models "pick out" salient patterns within their inputs, while ignoring the rest. Humans are able to characterize everything they see and hear, and so our models should be able to as well. These approaches may also enable the modeling of long-term linguistic dependencies, another currently impoverished aspect of our models.

8.3.10 Incorporating an interactive feedback loop

This work was inspired in part by unsupervised speech processing algorithms such as S-DTW (Park and Glass, 2008) and Lee and Glass' generative model of speech (Lee and Glass, 2012). What sparked the ideas that lead to this thesis was the observation that while algorithms like S-DTW have only the speech audio signal to work with, humans are able to take advantage of many rich sources of information at once. The advances made by thesis represent the addition of only one of these additional sources of information - vision, and in a still-frame context at that. There are many aspects of human learning that go beyond the paradigm we investigate here. One aspect that we believe is significant is the feedback loop of interactivity. Humans are not only able to observe their environment, but are also able to take action to change it. This feedback is also manifested in dialog between multiple agents; the primary purpose of language is to enable us to communicate with other humans, and so it is worth asking to what degree a computer can model human language without being capable of dialog.

8.3.11 Deeper analysis of learned representations

The speech representations learned by our models were analyzed by (Drexler and Glass, 2017), who provided evidence that different levels of linguistic abstraction were modeled by different layers in our audio networks. For example, the representations at the lowest layers of the network were more speaker-dependent than the upper layers, and also captured phonetic information as opposed to semantic information, which was more concentrated at the upper layers. However, many simple questions, such as "how many words does the network know?" do not have straightforward answers. A deeper analysis of the representations learned by our models - both in the acoustic and visual space - may lead to new insights and improved models.

8.4 Closing Statement

As a technology, automatic speech recognition has gone from science fiction to tangible reality in under 50 years. It has become a ubiquitous and essential part of many people's lives, and a lucrative product in the portfolio of many companies. Similar advancements has been achieved by natural language processing, computer vision, and many other machine learning subfields. In nearly all cases, however, the learning algorithms (as well as the data) that power these applications are unimodal. They are completely isolated from one another. They do not take advantage of the incredibly rich web of reciprocal context that exists in the world that humans are immersed in. Our computing systems continue to grow in computational power and storage capacity. The sensors and devices that facilitate data collection continue to grow smaller, more ubiquitous, and more integrated into our daily lives.

We thus have an incredible opportunity before us to create machine learning systems that treat *all* modalities as first-class citizens. Humans learn to communicate across many modalities with enormous flexibility, robustness, and ease. The fact that these skills are learned organically via immersion and interaction is one of most astounding and unique aspects of our nature. Endowing computers with this ability to *learn to communicate* would enable them to realize their full potential as intellectual partners to humans, bringing us tremendous benefit as we march forward into the future.

Appendix A

Detailed Experimental Results

A.1 Relation Between Word Clusters and Imagenet Synsets

Table A.1 displays the 40 lowest variance acoustic clusters (detailed in Section 6.3.3) paired with their closest ILSVRC12 synset label.

A.2 Additional Sliding Window-Based Pattern Clusters

Figures A-1 and A-2 display additional "picture dictionary" cluster visualizations from the procedure detailed in Section 6.3.1.

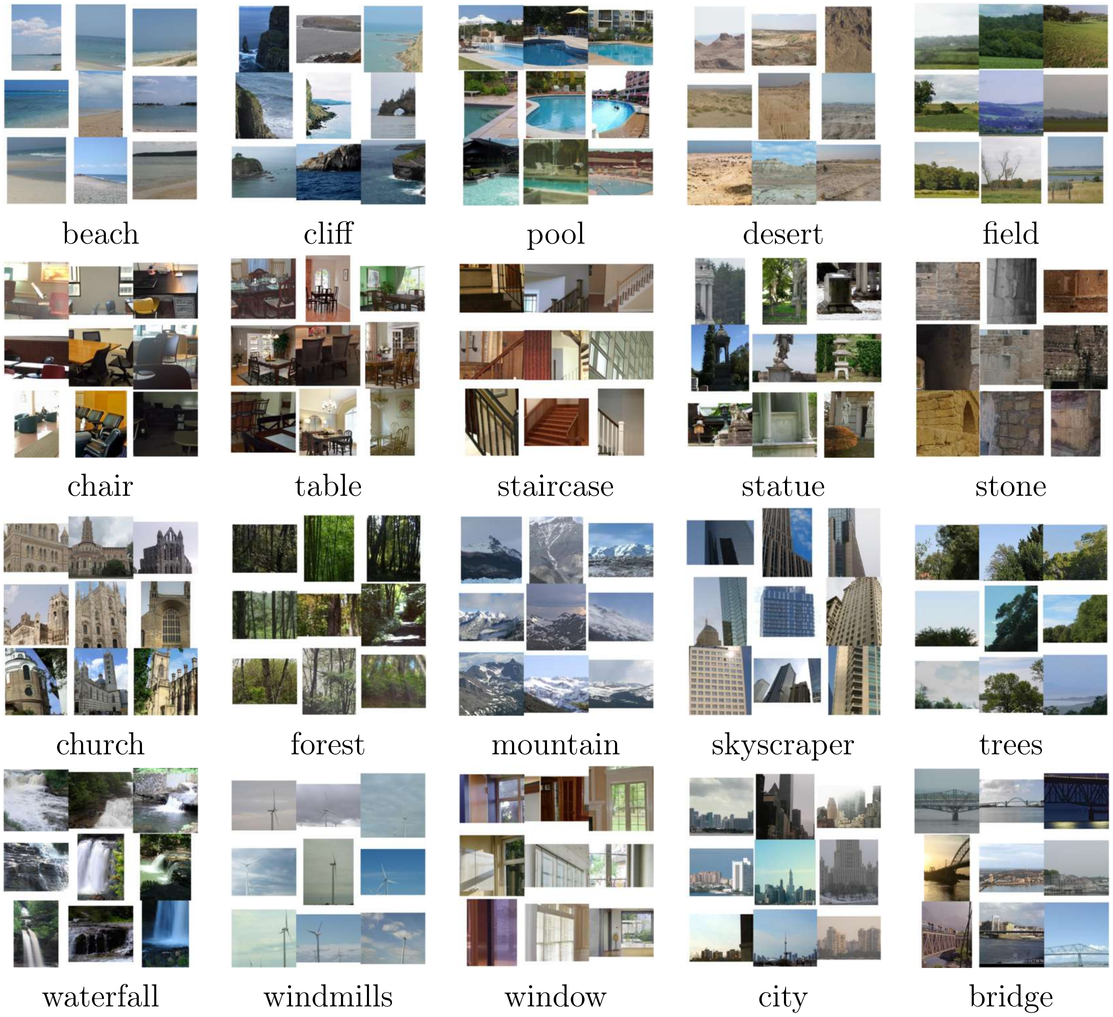


Figure A-1: Additional audio-visual cluster visualizations (1 of 2)

Cluster	ILSVRC synset	Similarity
snow	cliff.n.01	0.14
desert	cliff.n.01	0.12
kitchen	patio.n.01	0.25
restaurant	restaurant.n.01	1.00
mountain	alp.n.01	0.50
black	pool_table.n.01	0.25
skyscraper	greenhouse.n.01	0.33
bridge	steel_arch_bridge.n.01	0.50
tree	daisy.n.01	0.14
castle	castle.n.02	1.00
ocean	cliff.n.01	0.14
table	desk.n.01	0.50
windmill	cash_machine.n.01	0.20
window	screen.n.03	0.33
river	cliff.n.01	0.12
water	menu.n.02	0.25
beach	cliff.n.01	0.33
flower	daisy.n.01	0.50
wall	cliff.n.01	0.33
sky	cliff.n.01	0.11
street	swing.n.02	0.14
golf course	swing.n.02	0.17
field	cliff.n.01	0.20
lighthouse	beacon.n.03	1.00
forest	cliff.n.01	0.20
church	church.n.02	1.00
people	street_sign.n.01	0.17
baseball	baseball.n.02	1.00
car	freight_car.n.01	0.50
shower	swing.n.02	0.17
people walking	(none)	0.00
wooden	(none)	0.00
rock	toilet_tissue.n.01	0.20
night	street_sign.n.01	0.14
station	swing.n.02	0.20
chair	barber_chair.n.01	0.50
building	greenhouse.n.01	0.50
city	cliff.n.01	0.12
white	jean.n.01	0.33
sunset	street_sign.n.01	0.11

Table A.1: The 40 lowest variance, uniquely-labeled acoustic clusters paired with their most similar ILSVRC2012 synset.

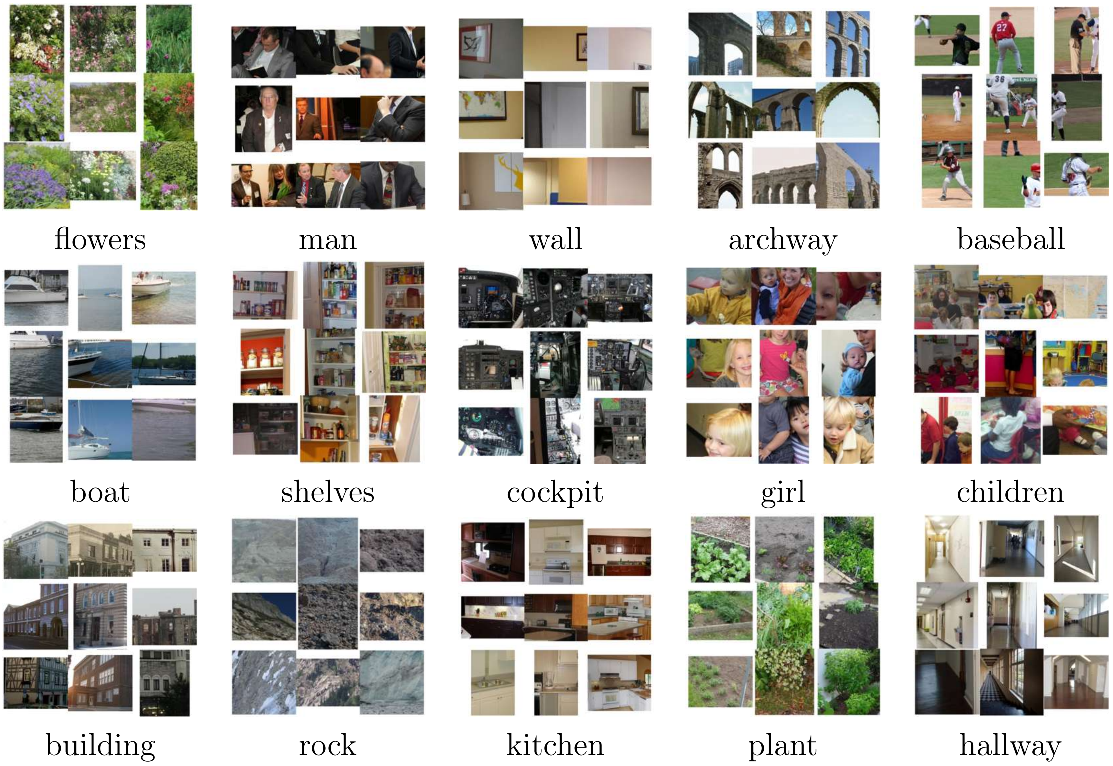


Figure A-2: Additional audio-visual cluster visualizations (2 of 2)

Bibliography

Gilles Adda and Joseph Mariani. 2010. Language resources and Amazon Mechanical Turk: Legal, ethical and other issues. In *Proc. International Conference on Language Resources and Evaluation (LREC)*.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, Z. Lawrence, and D. Parikh. 2015. VQA: Visual question answering. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.

K. Appel and W. Haken. 1977. Every planar map is four colorable. part i: Discharging. *Illinois J. Math.* 21(3):429–490. https://projecteuclid.org:443/euclid.ijm/1256049011.

Ibrahim Badr, Ian McGraw, and James Glass. 2011. Pronunciation learning from continuous speech. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

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

James Baker. 1975. The DRAGON system - an overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing (TASSP)* 23(1):24–29.

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando DeFreitas, David M. Blei, and Michael I. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research (JMLR)* 3:1107–1135.

Samy Bengio and Georg Heigold. 2014. Word embeddings for speech recognition. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

Alessandro Bergamo, Loris Bazzani, Dragomir Anguelov, and Lorenzo Torresani. 2014. Self-taught object localization with deep networks. *CoRR* abs/1409.3964. http://arxiv.org/abs/1409.3964.

Jeff Bilmes. 1998. A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. In *Technical Report ICSI-TR-97-021*.

Christopher M. Bishop. 2011. *Pattern Recognition and Machine Learning*. Springer.

Leonard Bloomfield. 1933. *Language*. Holt, New York.

Leon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of Computational Statistics*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, Morgan-Kaufmann, pages 737–744.

Kevin Carbotte. 2017. *Nvidia's New Titan V Pushes 110 Teraflops From A Single Chip*. http://www.tomshardware.com/news/nvidia-titan-v-110-teraflops,36085.html.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend, and spell. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. 2015. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger, New York.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *Proc. International Conference on Learning Representations (ICLR)*.

Ramazan Cinbis, Jakob Verbeek, and Cordelia Schmid. 2016. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 39(1):189–203.

Corinna Cortes and Vladamir Vapnik. 1995. Support vector networks. *Machine Learning* 20(3):273–297.

S.B. Davis and P. Mermelstein. 1980. *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. Morgan Kaufmann Publishers Inc., San Francisco.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. Unsupervised visual representation learning by context prediction. *CoRR* abs/1505.05192. http://arxiv.org/abs/1505.05192.

Mark Dredze, Aren Jansen, Glen Coppersmith, and Kenneth Church. 2010. NLP on spoken documents without ASR. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*.

Jennifer Drexler and James Glass. 2017. Analysis of audio-visual features for unsupervised speech recognition. In *Grounded Language Understanding Workshop*.

Emmanuel Dupoux. 2016. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *CoRR* abs/1607.08723. http://arxiv.org/abs/1607.08723.

Hao Fang, Saurabh Gupta, Forrest Iandola, Srivastava Rupesh, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, Platt John C., C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Karen Fort, Adda Gilles, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics* 37:413–240.

Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Proc. Neural Information Processing Systems (NIPS)*.

John Garofolo, Lori Lamel, William Fisher, Jonathan Fiscus, David Pallet, Nancy Dahlgren, and Victor Zue. 1993. The TIMIT acoustic-phonetic continuous speech corpus.

Spandana Gella, Rico Sennrich, Frank Keller, and Mirella Lapata. 2017. Image pivoting for learning multilingual multimodal representations. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

James Glass. 2012. Towards unsupervised speech processing. In *Information Science, Signal Processing and their Applications (ISSPA)*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.

Sharon Goldwater, Thomas Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: exploring the effects of context. *Cognition* 112:21–54.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge.

Inc. Google. 2018. *Google Cloud Speech API Language Support*. https://cloud.google.com/speech/docs/languages.

Joris Guérin, Olivier Gibaru, Stéphane Thiery, and Eric Nyiri. 2017. CNN features are also great at unsupervised classification. *CoRR* abs/1707.01700. http://arxiv.org/abs/1707.01700.

David Harwath and James Glass. 2015. Deep multimodal semantic embeddings for speech and images. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

David Harwath and James Glass. 2017. Learning word-like units from joint audio-visual analysis. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

David Harwath, Timothy J. Hazen, and James Glass. 2012. Zero resource spoken audio corpus analysis. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

David Harwath, Antonio Torralba, and James R. Glass. 2016. Unsupervised learning of spoken language with visual context. In *Proc. Neural Information Processing Systems (NIPS)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR* abs/1512.03385. http://arxiv.org/abs/1512.03385.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18:1527–1554.

Peter Howell and Karima Kadi-Hanifi. 1991. Comparison of prosodic properties between read spontaneous speech material. *Speech Communication* 10:163–160.

Eric Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. 2001. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Journal of Machine Learning Research (JMLR)*.

Aren Jansen, Kenneth Church, and Hynek Hermansky. 2010. Toward spoken term discovery at scale with zero resources. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

Aren Jansen and Benjamin Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

Fred Jelinek. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE* 64(4):532–556.

Fred Jelinek. 2004. Some of my best friends are linguists. In *Proc. International Conference on Language Resources and Evaluation (LREC)*.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *arXiv preprint arXiv:1408.5093*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Fei fei Li, C. Lawrence Zitnick, and Ross B. Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pages 1988–1997.

Mark Johnson. 2008. Unsupervised word segmentation for Sesotho using adaptor grammars. In *ACL SIG on Computational Morphology and Phonology*.

Peter Jusczyk. 1997. *The discovery of spoken language*. MIT Press, Cambridge.

Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

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

Andrej Karpathy, Armand Joulin, and Fei-Fei Li. 2014. Deep fragment embeddings for bidirectional image sentence mapping. In *Proc. Neural Information Processing Systems (NIPS)*.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mirjam Killer, Sebastian Stuker, and Tanja Schultz. 2003. A tree-trellis based fast search for finding the n-best sentence hypothesis in continuous speech recognition. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*.

Andy Klein. 2017. *Hard Drive Cost per Gigabyte*. https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

P. Koehn, F.J. Och, and D. Marcu. 2013. Statistical phrase-based translation. In *NAACL*.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. 2014. What are you talking about? text-to-image coreference. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pages 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Chia-Ying Lee and James Glass. 2012. A nonparametric Bayesian approach to acoustic model discovery. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chia-Ying Lee, Timothy J. O'Donnell, and James Glass. 2015. Unsupervised lexicon discovery from acoustic input. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

M. Paul Lewis, Gary F. Simon, and Charles D. Fennig. 2016. *Ethnologue: Languages of the World, Nineteenth edition*. SIL International. Online version: http://www.ethnologue.com.

William Lewis. 2015. Skype translator: Breaking down language and hearing barriers. In *Translating and the Computer*.

Bo Li, Tara Sainath, Arun Narayanan, Joe Caroselli, Michiel Bacchiani, Ananya Misra, Izhak Shafran, Hasim Sak, Golan Pundak, Kean Chin, Khe Chai Sim, Ron J. Weiss, Kevin Wilson, Ehsan Variani, Chanwoo Kim, Olivier Siohan, Mitchel Weintraub, Erik McDermott, Rick Rose, and Matt Shannon. 2017. Acoustic modeling for Google Home. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

Seppo Liannainmaa. 1970. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. In *Master's Thesis, Univeristy of Helsinki*.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsung-Yi Lin, Michael Marie, Serge Belongie, Lubomir Bourdev, Ross Girshick, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. 2015. Microsoft COCO: Common objects in context. In *arXiv:1405.0312*.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *Proc. International Conference on Machine Learning (ICML)*.

Warren McCulloch and Walter Pitts. 1943. A logical calculus of ideas immanent in nervous activity. In *Bulletin of Mathematical Biophysics*.

Ian McGraw, Ibrahim Badr, and James Glass. 2013. Learning lexicons from speech using a pronunciation mixture model. *IEEE Transactions on Audio, Speech and Language Processing* 21(2):357–366.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. 2015. Eesen: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

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

Abdelrahman Mohamed, George Dahl, and Geoff Hinton. 2012. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech and Language Processing* 20(1):14–22.

B.C.J. Moore. 1997. *An Introduction to the Psychology of Hearing*. Academic Press.

Lucas Ondel, Lukas Burget, and Jan Cernocky. 2016. Variational inference for acoustic unit discovery. In *5th Workshop on Spoken Language Technology for Under-resourced Language*.

Alan Oppenheim and Ronald Schafer. 2009. *Discrete Time Signal Processing*. Prentice-Hall.

Alex Park and James Glass. 2005. Towards unsupervised pattern discovery in speech. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

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

Douglas B. Paul and Janet M. Baker. 1992. The design for the Wall Street Journal-based CSR corpus. In *ACL Workshop of Speech and Natural Language*.

J. Pennington, R. Socher, and C.D. Manning. 2014. Glove - global vectors for word representation.

Fernando Pereira, Michael Riley, and Richard Sproat. 1994. Weighted rational transductions and their application to human language processing. In *Proc. NAACL Conference on Human Language Technologies (NAACL-HLT)*.

Sarah Perez. 2017. Siri usage and engagement dropped since last year, as Alexa and Cortana grew. *TechCrunch* https://techcrunch.com/2017/07/11/siri-usage-and-engagement-dropped-since-last-year-as-alexa-and-cortana-grew.

Steven Pinker. 1994. *The Language Instinct*. William Morrow and Company, New York.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The Kaldi speech recognition toolkit. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

Lawrence Rabiner, Chin-Hui Lee, B.H. Juang, and Jay Wilpon. 1989. Hmm clustering for connected word recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using Amazon's Mechanical Turk. In *Proc. NAACL Conference on Human Language Technologies (NAACL-HLT)*.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. *CoRR* abs/1605.05396. http://arxiv.org/abs/1605.05396.

Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater. 2015. A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. 1986. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, volume 1, chapter 8, pages 318–362.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y.

B.C. Russell, A.A. Efros, J. Sivic, W.T. Freeman, and A. Zisserman. 2006. Using multiple segmentations to discover objects and their extent in image collections. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Patricia Saylor. 2015, Available at https://github.com/psaylor/spoke. *Spoke: A Framework for Building Speech-Enabled Websites*. Master's thesis, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139.

Holger Schwenk and Jean-Luc Gauvain. 2005. Training neural network language models on very large corpora. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*.

Claude Shannon and Warren Weaver. 1949. *The Mathematical Theory of Communication*. The University of Illinois Press.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

Richard Socher and Fei-Fei Li. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Frank Soong and Eng-Fong Huang. 1991. A tree-trellis based fast search for finding the n-best sentence hypothesis in continuous speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

L. Specia, S. Frank, K. SimaâĂŹan, and D. Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the first ACL Conference on Machine Translation*.

Statista. 2018. *Number of Smartphone Users Worldwide*. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

Kenneth N. Stevens. 2000. *Acoustic Phonetics*. MIT Press.

R. Thiolliere, E. Dunbar, G. Synnaeve, M. Versteegh, and E. Dupoux. 2015. A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

Jasper Uijlings, Koen van de Sande, Theo Gevers, and Arnold Smeulders. 2013. Selective search for object recognition. *International Journal of Computer Vision* 104(2):154–171.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research (JMLR)* 9:2579–2605.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dimitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

M. Weber, M. Welling, and P. Perona. 2010. Towards automatic discovery of object categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ron Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proc. Annual Conference of International Speech Communication Association (INTERSPEECH)*.

Paul Werbos. 1982. Applications of advances in nonlinear sensitivity analysis. In *System modeling and optimization*.

Peter Young, Alica Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dong Yu and Li Deng. 2014. *Automatic Speech Recognition: A Deep Learning Approach*. Springer.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD international conference on Management of data*. pages 103–114.

Yaodong Zhang and James Glass. 2009. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *Proc. IEEE Workshop on Automfatic Speech Recognition and Understanding (ASRU)*.

Yaodong Zhang, Ruslan Salakhutdinov, Hung-An Chang, and James Glass. 2012. Resource configurable spoken query detection using deep boltzmann machines. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene CNNs. In *Proc. International Conference on Learning Representations (ICLR)*.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. In *Proc. Neural Information Processing Systems (NIPS)*.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.