University of Zurich[UZH]

Voice isolation, speech transcription and speaker re-identification in video

Thesis January 4, 2021

Patrick Düggelin
of Wangen SZ, Switzerland

Student-ID: 14-704-738
patrick.dueggelin@uzh.ch

Advisor: **Dr. Luca Rossetto**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

Acknowledgements

I want to thank Prof. Abraham Bernstein, Ph.D., and my supervisor Dr. Luca Rossetto for the opportunity to write a master thesis on the topic of multi-modal speech recognition in the context of video retrieval. I am very thankful for Dr. Luca Rossetto's continued support and valuable insights while writing this thesis. Additionally, I would like to thank the open-source community, without which this work would not be possible, especially the contributors to the projects used in this work: DeepSpeech[1], Kaldi[2], Pyannote-Audio[3], Deep-Learning-Based Audio-Visual Speech Enhancement and Separation[4], Audio-Visual Scene Analysis with Self-Supervised Multisensory Features[5], Denoiser[6], Whoosh[7]. Lastly, I want to thank my girlfriend Stefanie for her patience, moral support, and proofreading.

[1] https://github.com/mozilla/DeepSpeech/graphs/contributors
[2] https://github.com/kaldi-asr/kaldi/graphs/contributors
[3] https://github.com/pyannote/pyannote-audio/graphs/contributors
[4] https://github.com/danmic/av-se/graphs/contributors
[5] https://github.com/andrewowens/multisensory/graphs/contributors
[6] https://github.com/facebookresearch/denoiser/graphs/contributors
[7] https://github.com/mchaput/whoosh/graphs/contributors

Abstract

Speech is a salient information channel in recorded media, usually containing relevant semantic information complementing the visual signal. In a video retrieval setting, the speech signal can be transcribed automatically to enable spoken document retrieval by text query. Even though not the only factor, automatic transcription performance is the most important for the quality of such a retrieval system. In this work, we first assess the transcription quality of current state-of-the-art ASR systems and quantify the errors such systems make on a realistic dataset. We then examine if audio-visual speech enhancement methods can be used to improve the transcription quality. Based on these two preliminary studies' findings, we build three spoken document retrieval pipelines to index videos by what was said. We evaluate these systems on a set of manually captioned YouTube videos and find that speech enhancement slightly increases retrieval performance.

Zusammenfassung

Videos enthalten oftmals gesprochene Sprache mit zusätzlichen semantischen Informationen die das visuelle Signal ergänzen. Im Kontext von Video-Retrievalsystemen kann das Sprachsignal automatisch transkribiert werden, um die Suche nach gesprochenen Inhalten per Textabfrage zu ermöglichen. Die Qualität der automatischen Transkription ist nicht der einizge, aber einer der wichtigsten Faktoren für die Qualität eines solchen Retrievalsystems. In dieser Arbeit bewerten wir zunächst die Transkriptionsqualität aktueller State-of-the-Art automatischer Spracherkennungssysteme und quantifizieren die Fehler, die solche Systeme auf einem realistischen Datensatz machen. Anschließend untersuchen wir, ob audio-visuelle Ansätze zur Unterdrückung von Hintergrundgeräuschen eingesetzt werden können, um die Transkriptionsqualität zu verbessern. Basierend auf den Ergebnissen dieser beiden Vorstudien erstellen wir drei Pipelines für die Suche nach gesprochenen Inhalten in Videos. Wir evaluieren diese Systeme auf von Hand transkribierten YouTube-Videos und stellen fest, dass die Unterdrückung von Hintergrundgeräuschen die Retrievalleistung leicht erhöht.

Contents

1

Introduction

Motivation Videos often contain speech with additional semantic information complementing the visual signal. In a media retrieval setting, the speech signal can be transcribed automatically to enable speech-based retrieval by text query.

However, the automatic transcription of speech - commonly known as *Automatic Speech Recognition (ASR)* - is a non-trivial task in an unconstrained setting. While *ASR* has been an active research area for several decades, many of the existing approaches work well only in a constrained setting like dictation or speech-based user interfaces [Huang and Deng, 2010]. The main causes for the limited performance of *ASR* systems in an unconstrained setting are threefold: ASR systems and their applications expect a clean voice signal and are consequently ineffective in noisy acoustic environments. While their performance is reasonable for tasks with a limited vocabulary, they often struggle with unknown words, particularly proper names are a problem. Additionally, most *ASR* systems only work well for a specific set of speakers with a limited number of speech characteristics and accents [Preethi, 2017]. Third, in an unconstrained setting like the transcription of general video with multiple (previously unknown) speakers, various background noises, and occasional music, the accuracy of current *ASR* approaches degrades rapidly.

While most state-of-the-art *ASR* systems [Hannun et al., 2014, Amodei et al., 2015, Collobert et al., 2016] are solely based on the audio signal, human speech perception is naturally multi-modal. Studies as early as 1954 have shown that, particularly in noisy environments, speech intelligibility is improved by visual cues [Sumby and Pollack, 1954]. The visual modality can be used to improve state-of-the-art *ASR* systems' performance in the unconstrained setting of video transcription because video transcription is inherently multi-modal. More specifically, we can use recently proposed methods that can perform audio source separation on video by jointly considering both the visual and the aural modality [Gabbay et al., 2018, Owens and Efros, 2018, Afouras et al., 2018a, Ephrat et al., 2018, Zhao et al., 2018]. Using such a source separation approach as a preparation step for a state of the art *ASR* system has the potential to improve the transcription quality greatly. Improved transcription quality automatically leads to improved video retrieval by text queries.

Scope This thesis aims to identify and combine several state-of-the-art approaches for (multi-modal) source separation, automatic speech recognition, and speaker diarization

into a pipeline that is reliably able to determine what was said when and by whom, given any video as input. The goal of the pipeline is to be easily integrated into the open-source content-based multimedia retrieval system *vitriver* [Rossetto et al., 2016] to support both verbatim dialog search and speech-based topic extraction.

Structure This work is structured as follows: in Chapter 2, we discuss audio-visual speech recognition and examine the individual components of such systems with a special focus on the systems used throughout this thesis. We then conduct two preliminary studies on the *ASR* and the speech enhancement components of a speech-based video retrieval system: In Chapter 3, we analyze the typical errors of current state-of-the-art *ASR* systems, focusing on common characteristics of speech in videos: noisy environments, wildly different speaker characteristics, and an open set of speakers.

Chapter 4 discusses whether audio-visual speech enhancement approaches can be used to improve the *ASR* output in the context of video retrieval. Given the results of the two preliminary studies, in Chapter 5, we then build a prototype for speech-based video retrieval, and evaluate it on a realistic dataset. Chapter 6 identifies the limitations of this thesis and points out future work to be conducted to potentially improve our approach. Finally, we draw a conclusion in Chapter 7.

2

Background & Related Work

In this chapter, we discuss the background and related work. We start by giving an overview of classical uni-modal Automatic Speech Recognition in Section 2.1. We briefly look at the history and challenges of such systems and discuss the general architecture. We then examine the individual components of *Deep Speech* in detail and end this section with an analysis of *ASR* in the context of video retrieval. Next, in Section 2.2, we discuss multi-modal approaches, focusing on *audio-visual speech enhancement (ASVE)*. Finally, in Section 2.3, we discuss speaker re-identification. Overall, we put a special focus on the systems used in this thesis, but we also take a brief look at the history, datasets, and evaluation metrics commonly used.

2.1 Automatic Speech Recognition

2.1.1 Overview

History *Automatic Speech Recognition* has been an active area of research for at least five decades. According to [Yu and Deng, 2015], it became important as one of the last decade's emerging human-machine interaction methods. *ASR* use outside of research increased due to performance improvements enabled by two key factors: First, the computational power available today is orders of magnitude above the level just a decade ago, and second, the amount of transcribed real-life speech data increased. A metric that shows the progress of *ASR* systems over time particularly well is the number of different words a system can recognize: While IBM Shoebox[1], one of the first experimental *ASR* systems in the 60s, worked on just ten words, the systems of the 70s already considered thousands of words [Lowerre, 1976]. The statistical HMM systems developed since the 80s expanded the vocabulary to ten thousand words, and current deep learning-based systems work with millions of words.

Challenges While *ASR* improved drastically, the main challenges, especially in the context of open-domain video transcription, remain the same: Environmental noise, such as ambient music or a car passing in the background, can drastically affect *ASR*

[1] https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html

performance. Another challenge is rare or unknown words to the system, which might be misrecognized. Lastly, most *ASR* systems do not reach the same level of transcription quality for varying speaker characteristics. While they might produce nearly perfect transcriptions for male US speakers, a Swiss speaker, for example, might lead to imperfect transcriptions.

Architecture Figure 2.1 illustrates the four components of typical ASR systems: Signal processing and feature extraction, acoustic model, language model, and hypothesis search [Yu and Deng, 2015]. With the rise of ever-larger deep-learning models, the formerly distinct components begin to merge towards single end-to-end models [Collobert et al., 2016]. However, to think in distinct components still helps to grasp the fundamental concepts.

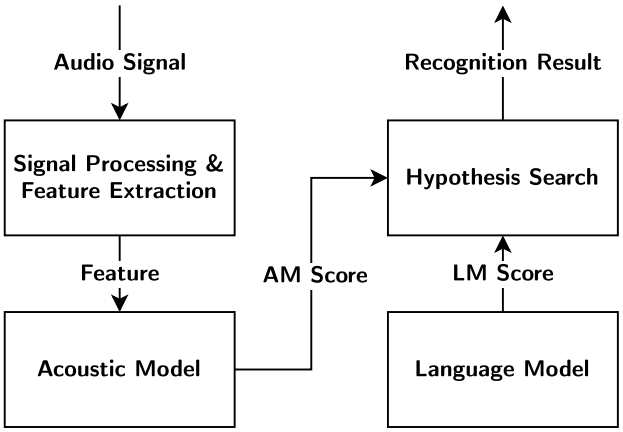

Figure 2.1: Architecture of ASR Systems [Yu and Deng, 2015].

The signal processing and feature extraction component takes a raw audio signal to produce salient feature vectors. This first step aims to get accurate speech representations that capture only the audio signal components representing the actual spoken words. These representations aim to discard information about irrelevant parts of the audio signal, such as background noise or individual speaker characteristics. Also, the resulting feature vectors should be relatively low-dimensional and uncorrelated. To accomplish this, signal processing is often used to enhance the speech by removing noise and channel distortions before computing the actual feature vectors from the processed audio. Since the 80s, most *ASR* systems' feature extraction components relied on hand-crafted feature extraction methods, the most common at the time of writing being *Mel-frequency cepstral coefficients (MFCCs)* [Chen et al., 1976]. Recently, however, [Schneider et al., 2019] proposed *wav2vec*, a new semi-supervised approach for learning robust speech representations.

The acoustic model takes in the variable-length sequence of feature vectors and produces sequences of text along with their respective probabilities. Most traditional acoustic models rely on heavily engineered stages, which need to be tuned by experts in the

domain. In particular, *Hidden Markov Model (HMM)* approaches are based on the assumption that the sequence of text consists of words, which, in turn, are a sequence of phonetic units (usually triphones), and each phonetic unit is represented as a sequence of HMM states [Rabiner and Juang, 1986]. Linguistic experts then create a look-up table of words with their corresponding phonetic units. A graph is constructed, whereby the nodes are states of phonetic units connected to form words, in a manner that the best path in the graph corresponds to the most likely word sequence for a given feature sequence [Bengio and Heigold, 2014]. While in the past acoustic models were predominantly based on *HMMs*, recently, deep-learning-based approaches rapidly took the lead. Deep neural networks (DNNs) first replaced Gaussian Mixture Models (GMMs) to model the probability of a feature vector being a certain phone, while HMMs were still used for sequence modeling. Later, end-to-end approaches [Bengio and Heigold, 2014, Hannun et al., 2014, Amodei et al., 2015, Collobert et al., 2016] were proposed, which make fewer assumptions about the underlying phonemic structure and instead directly generate text sequences from audio feature sequences. These end-to-end models need no more engineered domain knowledge but are purely data-driven instead. As such, they require much more data to train. In Section 2.1.2 we look at an implementation of such an end-to-end approach: Mozilla's *Project Deep Speech*[2], and how it solves these challenges.

The language model (LM) estimates the probability of hypothetical word sequences, or *LM Score*, by learning the correlations of words on large text corpora. Assuming the acoustic model generates from an audio sample two similar-sounding word sequences: "recognize speech," and "wreck a nice beach," the language model then assigns a probability to each sequence based on its context trained on. Assume the LM was trained on this thesis's text, "recognize speech" would most likely be assigned a higher probability. ASR systems typically used statistical n-gram language model implementations such as *KenLM* [Heafield, 2011], but recently deep-learning-based models such as *BERT* [Devlin et al., 2018] emerged.

Because computing the probabilities of all possible text sequences is infeasible, especially with the large vocabularies and unconstrained sequence length in video transcription, *beam search* is normally used for hypothesis search. The beam search strategy generates the transcripts from left-to-right one word at a time while keeping a fixed number (beam) of the most likely candidates at each time step.

Data Training statistical automatic speech recognition systems requires hundreds of hours of transcribed speech. Newer neural end-to-end approaches need even more data. Many data sets are openly available in various domains such as telephone conversations, audiobooks, speech commands, broadcast news, etc. The most notable being Switchboard [Godfrey and Holliman, 1997], which consists of approximately 260 hours of telephone conversations of more than 500 speakers, and LibriSpeech [Panayotov et al., 2015], which consists of one thousand hours of transcribed audiobooks. While research ASR systems commonly use these classical data sets to compare their performance, com-

[2] https://github.com/mozilla/DeepSpeech

mercial ASR systems are trained on much more diverse data, often proprietary to the company. One initiative to provide a large, diverse data set to the public is Mozilla's Common Voice [Ardila et al., 2020]. It uses crowd-sourcing to collect and validate speech samples from a wide variety of speakers. For systems to be more robust to environmental noise, they can be trained on data containing realistic natural background noise, such as CHiME [Barker et al., 2018], or clean speech augmented with environmental noise, such as MS-SNSD [Reddy et al., 2019].

Evaluation To measure the performance of ASR systems, they are evaluated on a test set of transcribed speech samples, which were not seen in training. The system's output - called hypothesis - is compared to what was actually said - called reference - using a given metric. The most common metric to measure ASR performance is the word error rate (WER), which is based on the *Levenshtein Distance* of words between the reference and hypothesis. It is computed as:

$$WER = \frac{S + D + I}{N} \tag{2.1}$$

Where S is the number of substituted words, D is the number of deleted words and I is the number of inserted words in the hypothesis, and N is the total number of words of the reference. While WER captures transcription quality on word-level quite well, it doesn't capture transcriptions' semantic meaning. A word spelled differently in the hypothesis than in the reference counts as an error, while semantically, the transcription is correct. Phoneme error rate (PER) or character error rate (CER) can, in some cases, mitigate these effects.

2.1.2 Deep Speech

This section describes Mozilla's *Project Deep Speech* architecture in more detail, as it is the main *ASR* system used in this work. The project aims to create an open-source speech recognition engine that runs inference on consumer-grade hardware. As such, it is perfectly suited for the transcription of videos for retrieval. The architecture of Mozilla's Project Deep Speech differs in several aspects from the original implementation, which used to be based on *Deep Speech* [Hannun et al., 2014].

The system uses a recurrent neural network (RNN) to generate English text directly from speech spectrograms without the need for engineered phonemic knowledge. Figure 2.2 gives an overview of the network architecture and its components, each of which we will explain in detail.

Let x be a single utterance (Represented by the audio waveform on the very bottom in Figure 2.2) and y its transcription (Represented by the letters on the very top in Figure 2.2). Each utterance x is a time-series of length T, where every time-slice is a vector of audio features, x_t, where $t = 1, ..., T$. While the original implementation of *Deep Speech* by [Hannun et al., 2014] directly uses speech spectrograms as features,

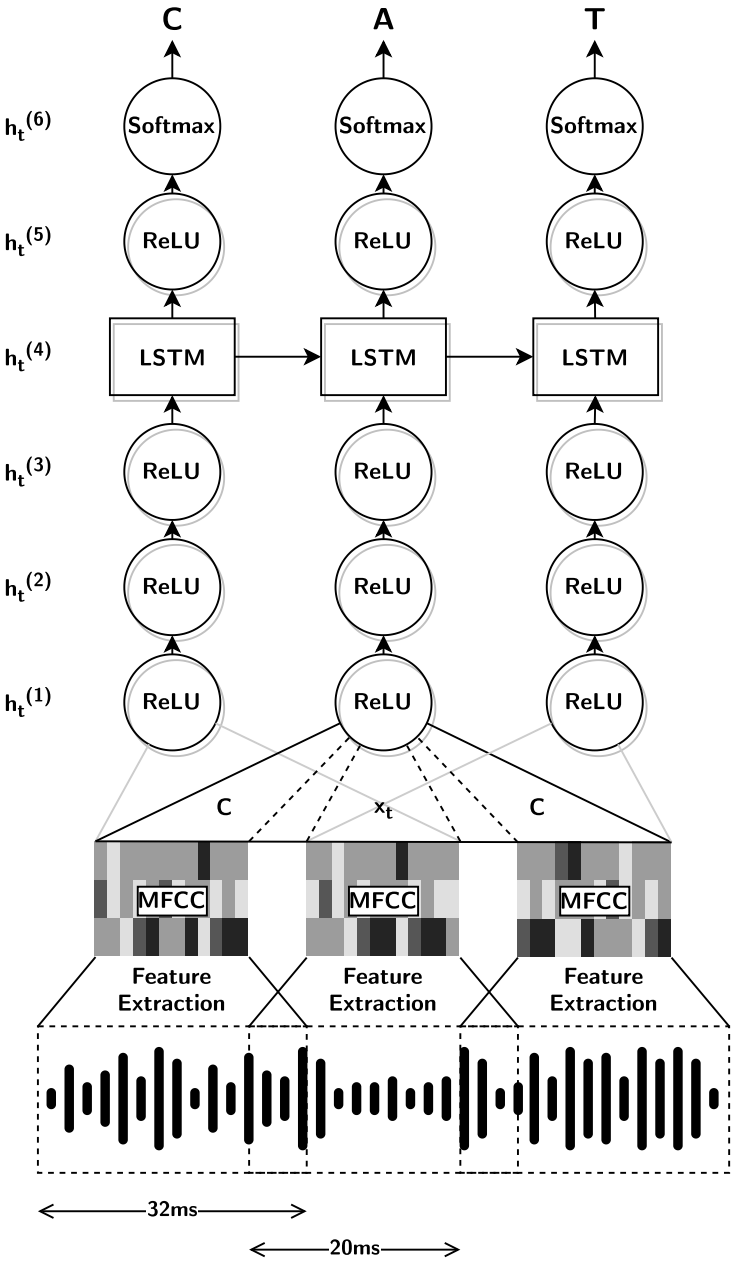

Figure 2.2: Architecture of Mozilla Voice STT (Adapted from [Hannun et al., 2014]).

Mozilla's implementation uses *MFCCs*[3]. So $x_{t,p}$ denotes the p-th *MFCC* feature in the audio frame at time t. In contrast to the original implementation, in this work, we

[3]For an overview of why *MFCCs* make good features for ASR and a detailed tutorial how they are computed from an audio-signal, we refer the reader to http://practicalcryptography.com/miscellaneous/machine-learning

use Mozilla Voice STT's default frame length of 32ms with a step size of 20ms and a *Hamming Window* [Smith, 2011][4] to smooth the individual frames. The bottom part of Figure 2.2 illustrates this feature extraction step.

The goal of the RNN is to generate from an input sequence x, a sequence of character probabilities $\hat{y}_t = P(c_t \mid x)$, where for English $c_t \in \{a, b, c, ..., z, space, apostrophe, blank\}$. The model comprises 6 layers, the first 3 being non-recurrent, followed by a recurrent LSTM layer, another non-recurrent layer, and the output layer. While the original implementation uses a bi-directional RNN [Schuster and Paliwal, 1997], Mozilla's implementation uses an unidirectional LSTM [Hochreiter and Schmidhuber, 1997].

More formally, for an input x, the hidden units at layer l are denoted $h^{(l)}$, with the convention that $h^{(0)}$ is the input. For the first layer at each time step t, the output depends on the *MFCC* frame x_t along with a context of C frames of each side. (While in practice, we use $C = 9$, in Figure 2.2, we show the context to be a single frame $C = 1$ for a better overview.) The following non-recurrent layers use the output of the previous layer independently for each time step. So for each time step t, the first 3 layers are computed by:

$$h_t^{(l)} = g(W^{(l)} h_t^{(l-1)} + b^{(l)}) \tag{2.2}$$

where the activation function $g(z) = min\{max\{0, z\}, 20\}$ is a clipped rectified-linear unit (ReLu) and $W^{(l)}, b^{(l)}$ are the weight matrix and bias parameters for layer l.

The LSTM layer $h^{(4)}$ uses the outputs of the previous layers $h_t^{(3)}$ and the output of the previous time step $h_{t-1}^{(4)}$:

$$h_t^{(4)} = g(W^{(4)} h_t^{(3)} + W^{(r)} h_{t-1}^{(4)} + b^{(4)}) \tag{2.3}$$

Layer 5 is the same as the first three layers, with the input being the output of the recurrent layer:

$$h_t^{(5)} = g(W^{(5)} h_t^{(4)} + b^{(5)}) \tag{2.4}$$

The outputs of layer 5 are fed into an output layer, which approximates the character probabilities for each time step t and character k in the vocabulary:

$$h_{t,k}^{(6)} = \hat{y}_{t,k} = \sigma(W^{(6)} h_t^{(5)} + b^{(6)})_k \tag{2.5}$$

where σ is the softmax defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.6}$$

Given the predicted character probabilities $\hat{y}_t$, *Connectionist Temporal Classification (CTC)* loss [Graves et al., 2006] is used to train the network using the Adam optimizer. *CTC* is a loss function for sequence modeling commonly used to train end-to-end *ASR* models. In particular, *CTC* solves the problem of missing alignments between the audio of the input and the corresponding characters in the output [Hannun, 2017][5].

[4] https://ccrma.stanford.edu/~jos/sasp/Hamming_Window.html
[5] https://distill.pub/2017/ctc/

2.1.3 Automatic Speech Recognition in the Context of Video Retrieval

Overview Content-based video retrieval deals with indexing and searching videos in large databases. Content-based means the search analyzes not only the metadata but also the actual content of the video, such as shapes, colors, or objects. [Patel and Meshram, 2012]. While classical video retrieval is mostly concerned with visual content, videos are multi-modal. They often contain speech, so we naturally want to be able to search them by what was said. This task of indexing and searching videos (or any other documents) by what was said is commonly known as spoken document retrieval (SDR). The aim of SDR is to find documents or excerpts that directly contain the words in a textual query or are semantically similar. To this end, the documents are transcribed using an *ASR* system and indexed by the resulting transcriptions. The indexing process often involves transforming words to their canonical root stems and removing frequently occurring words without semantic meaning - so-called stopwords. The remaining words are compiled into an inverted index, which maps the documents to the words they contain. When the user issues a query, each document in the database is compared to the query using a matching function $R(Q, D)$, which computes the relevance R of the query Q to a document D [Hauptmann, 2006]. While each of the SDR steps poses its own challenges and can be optimized individually, we focus mostly on improving ASR in this work.

Evaluation Spoken document retrieval can be evaluated on different levels. The performance of the system as a whole is measured in terms of how well the spoken documents are retrieved, but measuring the performance of sub-components - in particular speech-recognition, segmentation, and retrieval - can be useful as well.

Even though there is a great number of retrieval quality metrics, in this section we focus on the performance measures relevant for this work, which are *Mean Reciprocal Rank*, *Mean average precision*, and *Discounted Cumulative Gain*. First, the *Mean Reciprocal Rank (MRR)* measure is a simple indicator of retrieval performance. Given a single query, the reciprocal rank is defined as the reciprocal of the rank of the first relevant result to the query. For a set of queries, *MMR* is then the mean of the reciprocal ranks of all queries [Craswell, 2009]. More formally, for a set of queries Q, *MRR* is defined as:

$$MRR = \left\{ \sum_{i=1}^{|Q|} 1/r_i \right\} / \mid Q \mid \tag{2.7}$$

Where r_i is the rank of the first relevant document to the query.

Mean average precision (MAP) is one of the most common metrics to measure open-ended retrieval performance. First, *Precision* is the fraction of retrieved documents relevant to the query in relation to all retrieved documents. Given a single query, if the retrieved documents are ranked, *average precision* is the average of the *precision* scores after each rank. Documents that are not retrieved are counted towards the average with a *precision* of 0. [Hauptmann, 2006]. *Average precision*, therefore, rewards systems that

retrieve relevant documents at high ranks. More formally, given a total of N_r items in the database relevant to a query, assume that the system retrieves k relevant items and they are ranked as r_1, r_2, ..., r_k. Then, the *average precision AP* is computed as [Zhang and Zhang, 2009]:

$$AP = \left\{ \sum_{i=1}^{k} i/r_i \right\} /N_r \tag{2.8}$$

Mean average precision (MAP) is the mean of *average precision* values over a set of queries. *MAP* can only be used if the number of relevant documents (N_r) to a certain query is known in advance, which is not realistic for large SDR systems. As an alternative, *(average) precision at k* can be used, which limits the number of considered documents to k. *Precision at k* has two drawbacks: First, it doesn't consider the actual rank of the documents among the top k, and second, if there are fewer than k relevant documents, even perfect systems get a score lower than 1. Additionally all *precision*-based metrics are based on binary relevance judgments, while some documents might be more relevant than others. To compensate for this weakness *Discounted Cumulative Gain* [Järvelin and Kekäläinen, 2002] can be used, which allows for graded relevance judgments. The assumption of *DCG* is that particularly relevant documents that appear further down in a list of search results should be penalized, as the graded relevance value is reduced logarithmically proportional to the rank of the result. More formally, for a given rank position p, the *Discounted Cumulative Gain* is computed as [Järvelin and Kekäläinen, 2002]:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.9}$$

Where rel_i is the relevance grade of the document at rank i.

Challenges The challenges for *ASR* systems mentioned in 2.1.1, namely environmental noise, vocabulary size, and speaker characteristics, are especially present in spoken document retrieval because documents contain real-world data not recorded specifically for *ASR*. Vocabulary size is a problem in particular: because most *ASR* systems have a fixed vocabulary of words, out-of-vocabulary (OOV) words will never be recognized and can therefore not be used in the retrieval process. The problem is exacerbated by the fact that words not included in the vocabulary are often misrecognized instead of being marked as OOV, which leads to false positives on retrieval [Hauptmann, 2006].

Naturally, the retrieval performance increases with better *ASR* performance. [Hauptmann and Wactlar, 1997] conducted a study on the correlation between speech recognition performance and spoken document retrieval performance. The study compares WER to MAP for different levels of transcription quality. The baseline ASR system used in the study has a WER of 50%. To generate better transcripts, the output of the system is combined with the reference, and to generate worse transcripts, they randomly introduced errors. The generated transcripts are then used in a spoken

document retrieval system. They find that for low WER, retrieval is robust enough to compensate. More specifically, MAP degrades very little for transcripts with increasing WER below 20%. At WER higher than 35%, the information retrieval effectiveness starts to decline noticeably [Hauptmann and Wactlar, 1997].

Another challenge is the identification and removal of non-speech sections. An ASR system might for example misrecognize an introductory sound-track, which is usual in web-videos for speech. Voice activity detection systems can be used to filter such non-speech sections. As a plus, audio streams can be automatically segmented into coherent passages, which again increases SDR performance.

2.2 Audio-Visual Approaches

The general idea of audio-visual speech recognition is to complement the audio signal with information present only in the video signal. It is motivated by the fact that human speech perception is multi-modal by nature, which is, for example, demonstrated by the McGurk effect. In their paper, [Mcgurk and Macdonald, 1976] describe an illusion that occurs when the audio signal of a particular sound is paired with the video signal of another sound, which results in the perception of a third sound. The visual information a person receives when they see a person speaking changes how they perceive the sound.

2.2.1 Audio-Visual Speech Recognition

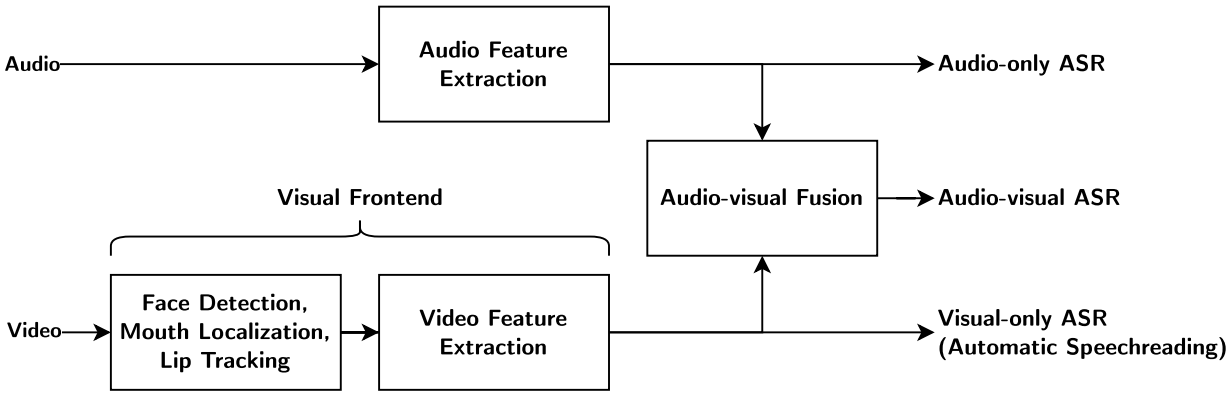

Figure 2.3: The main building blocks of an audiovisual automatic speech recognizer as described in [Potamianos et al., 2012].

Research in audio-visual speech recognition has been conducted since the mid-eighties [Petajan et al., 1988, Benoit, 1996, Heckmann et al., 2001]. Most of these earlier systems follow a similar structure of building blocks outlined in Figure 2.3: A visual front-end consisting of face detection, mouth localization and lip-reading components, a speech recognition component, and an audio-visual fusion strategy. While most of the approaches promise an improvement of audio-only *ASR* performance, there are two main

drawbacks. First, they are rarely tested on a common audio-visual data set and, as a consequence, are hard to compare. Second, most audio-visual speech recognition studies are conducted on data sets with a minimal scope, consisting of short small vocabulary utterances voiced by a small number of speakers [Cooke et al., 2006, Harte and Gillen, 2015]. For *ASR* systems in the context of speech-based video retrieval, data sets of a much broader scope are needed.

More recent systems use deep learning for the individual components, such as face detection or *ASR*, or even drop the visual front-end and use an end-to-end approach [Palaskar et al., 2018]. However, most audio-visual *ASR* approaches are still based on the architecture shown in Figure 2.3, which focuses as much on the visual front-end as it focuses on ASR. On the upside, systems using this approach can often perform automatic lipreading "for free," but on the downside, they make assumptions about the domain of the videos used. In particular, if face detection is used in the visual frontend, videos strictly need to show a speaker's face. While such approaches work well, for example, to transcribe presentations, they are not suitable for general video retrieval because the videos can be from wildly different domains. Consequently, we focus on uni-modal *ASR* systems as described in Section 2.1 as our starting point. The video signal can then potentially be used to improve the individual *ASR* components: Feature extraction, acoustic model, language model, or hypothesis search. [Gupta et al., 2017], for example, show that visual context - object and scene features in particular - can be used to adapt the acoustic model of a speech-to-text system and improve word error rate on general video transcription. In this work, we focus on improving the feature extraction component using audio-visual speech enhancement.

2.2.2 Audio-Visual Speech Enhancement

Overview Speech enhancement aims to extract a target speech signal from a mixture of sounds produced by multiple sources (for example, environmental noise, background music, and other speakers). As such, it is a sub-problem of the more general task of audio source separation, which is concerned with the separation of any specific sounds and not just speech. Traditionally, these tasks are approached using signal processing and/or machine learning applied to the audio signal [Michelsanti et al., 2020]. Recently however, multiple approaches exploit the multi-modality of speech to improve speech enhancement using deep learning techniques. Similar to the visual front end of end-to-end audio-visual speech recognition systems, most of the approaches rely on visual features based on the speakers' mouth region [Michelsanti et al., 2020]. [Owens and Efros, 2018], on the other hand, use self-supervised learning to pre-train audio-visual embeddings, which can not only be used for speech enhancement but also other audio-visual downstream tasks. Similarly, [Zhao et al., 2018] use large amounts of unlabeled video data to learn to locate image regions producing sounds and separating them.

Evaluation The performance of speech enhancement methods can be evaluated in terms of speech quality and intelligibility. While perceived speech quality is largely subjective and varies between listeners, intelligibility can be measured more objectively

[Michelsanti et al., 2020]. There is a wide range of commonly used listening tests involving human end-users to evaluate speech quality and intelligibility[6]. Still, because such tests are very time consuming, we resort to automatic metrics. The automatic evaluation metrics used for *audio-visual speech enhancement (ASVE)* are largely the same as for audio-only speech enhancement. In general, the automatic evaluation of speech enhancement is fairly complex compared to ASR evaluation because the metrics need to consider audio signals instead of text. The most widely used evaluation metric for speech enhancement is *perceptual evaluation of speech quality (PESQ)* [Rix et al., 2001], originally designed for telephone networks and codecs. The algorithm consists of several components such as level equalization, time alignment, filtering, disturbance processing, perceptual filtering, and time averaging. Using these steps, the algorithm tries to imitate the speech perception of humans [Rix et al., 2001]. There are many other metrics similar to *PESQ*, but most of them have the same limitation: They require a clean reference to compare the degraded signal. In the training phase, clean speech is mixed with environmental noise to produce a sample. In this case, the clean speech sample can be used as a reference. To measure speech enhancement performance "in the wild", these metrics are not useful because there is no reference to the enhanced speech. As such, these metrics are not useful to evaluate the speech enhancement performance for web-video. In this case, the enhanced speech can be fed into an *ASR* system and evaluated using WER.

Data As for the other tasks, an important success factor for *audio-visual source enhancement* using deep learning is the amount of data available. [Michelsanti et al., 2020] find that *GRID* [Cooke et al., 2006] and *TCD-TIMIT* [Harte and Gillen, 2015] are the most commonly used data sets for deep learning-based AVSE. *GRID* consists of 34 speakers who were videotaped, pronouncing 1000 sentences each in a controlled environment. The speakers are recorded in an acoustically isolated booth with a uniform blue background, their face is uniformly illuminated and all sentences have the same structure: <command(4)><color(4)><preposition(4)><letter(25)><digit(10)><adverb(4)>, where the number in brackets indicates the number of possible words at this position. Therefore, the vocabulary is limited to just 51 words, which possibly limits the generalization performance of a model trained on this data set. *TCD-TIMIT* is built in the same manner: 62 speakers pronouncing phonetically balanced sentences of the *TIMIT* corpus in front of a green screen. Recently, [Ephrat et al., 2018] introduced *AVSpeech*, a large-scale audio-visual dataset of speakers "in the wild". In total, the dataset contains about 4700 hours of short segments of 290 000 YouTube videos, spanning a wide variety of people, languages, and face poses. Other efforts were made to collect datasets of TED talks, British television programs, and movies [Chung and Zisserman, 2016, Afouras et al., 2018b, Roth et al., 2020]. While these datasets show speakers in a more natural setting, they still all focus to show the speakers' faces. This is probably because most ASVE approaches use face crops as visual features, and in turn leads to more such approaches.

[6]For an extensive list of commonly used listening tests and automatic metrics, we refer the reader to https://github.com/danmic/av-se

2.3 Speaker Re-Identification

Overview The term speaker identification is used ambiguously: On one hand, it is used to identify a previously known speaker. On the other hand, it is also used to determine whether multiple speech samples are from the same, unknown speaker. The former is also commonly known as *speaker recognition*, while the latter is known as *speaker verification* [Hansen and Hasan, 2015]. For video retrieval, we are interested in a type of *speaker verification*, more precisely whether a speaker in a certain video segment also speaks in another segment.

Recent speaker identification systems use deep learning to build feature representations of audio segments that are able to discriminate speakers: so-called *speaker embeddings*. Clustering or similarity measures can then be used to group or identify similar speakers. In a simple setting, where each audio segment contains a single speaker, *speaker embeddings* can be directly computed from the features extracted from raw audio. However, in a more realistic setting where multiple speakers may have a conversation with silence between turns or overlapping speech, a more robust pipeline is needed. This task of partitioning an audio stream into segments according to the speaker's identity is called *speaker diarization*. Figure 2.4 shows a *speaker diarization* pipeline implemented in *pyannote.audio* [Bredin et al., 2019]. While some recent approaches use supervised learning instead of clustering for speaker diarization [Zhang et al., 2019], many systems are still based on similar pipelines to *pyannote.audio*. As this pipeline based approach is fundamental to this work, we will have a closer look at it.

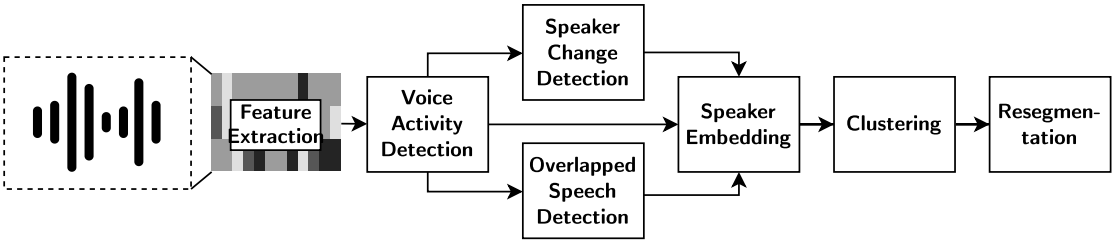

Figure 2.4: Speaker diarization pipeline as described in [Bredin et al., 2019].

Speaker Diarization Pipeline The first step in a *speaker diarization* pipeline is to extract salient features for the downstream tasks, mostly *MFCCs* are used. These features are then used in the following sequence labeling tasks: *voice activity detection (VAD)*, *overlapping speech detection*, and *speaker change detection*. *Pyannote.audio* provides generic building blocks to train such neural sequence labeling models [Bredin et al., 2019]. Long audio files of variable length are not practical or efficient, therefore *pyannote.audio* uses shorter fixed-length sequences. At training time, fixed-length sub-sequences are drawn randomly from the training set in order to form mini-batches. At test time, audio files are processed using overlapping sliding windows of the same length as used in training, and the resulting predictions are averaged for the

overlap. The step following feature extraction is voice activity detection, which detects speech regions in a given audio stream or recording. To this end, a binary sequence labeling model, as described above, can be used to classify audio segments as speech and non-speech. Given the speech regions of the audio stream, *speaker change detection*, and *overlapped speech detection* – which are again sequence labeling problems – are used to segment the audio into segments with distinct speakers.

These audio segments, more precisely the sequence of features extracted from the audio segments, are used to compute the actual speaker embeddings. While many speaker diarization systems use so-called *x-vectors* [Snyder et al., 2018] as embeddings, there are different approaches to directly compute them using metric learning, such as triplet loss.

The pipeline's individual building blocks can be tuned individually or as a pipeline whose hyper-parameters are jointly tuned. Joint optimization mostly yields better results than the combination of multiple building blocks that were trained independently [Yin et al., 2018].

3

Speech-to-text Study

Speech in web-video varies wildly between individual videos: A recorded lecture about biochemistry, for example, features a single lecturer speaking clearly, with little environmental noise. In contrast, a video about people reacting to a videogame includes multiple speakers and background music. At first glance, it might seem easier to automatically transcribe the first video than the second because it contains fewer speakers and background noise. But there is a third challenge for automatic transcription: Vocabulary. While the second video speakers use a rather simple and general vocabulary, the vocabulary used in the video about biochemistry is much more specific and harder to recognize. The above example shows the main challenges for speech recognition in web-video, namely different levels of environmental noise, a large range of speaker characteristics (accent, speech rate, pitch, etc.), and an open vocabulary. To be useful for video retrieval, automatic transcripts need to be of a certain quality, and therefore *ASR* systems need to be able to handle the above challenges.

This chapter focuses on the above challenges for *ASR*. We quantify the performance of three *ASR* systems under different conditions. In Section 3.1, we start by giving an overview of the experimental setup. In particular, we introduce the research questions to be answered and give an overview of the data and the systems used in the experiments. The following Section 3.2 goes into more detail for each research question and shows the respective results.

3.1 Experimental Setup

Goals This preliminary study aims at assessing *ASR* performance and quantifying shortcomings. More precisely, we are interested in the transcription performance of large-vocabulary continuous speech recognition systems in terms of word error rate under the conditions prevalent in web-video. The *ASR* performance is of interest to video retrieval because there is clearly a correlation between *ASR* performance and retrieval performance, as described in Section 2.1.3.

As a first step to identify potential shortcomings of *ASR* systems, we need to know how well they perform out of the box. While most systems report their performance in terms of WER, for example, on the *Librispeech* [Panayotov et al., 2015] test set, we're

interested in the performance in a more realistic setting. We therefore ask the following research question:

> **RQ_1: Baselines** How well do current state-of-the-art *ASR* systems perform in a realistic setting?

We use the results of this first research question as a baseline to investigate the effect of environmental noise and speaker characteristics on *ASR* performance. To quantify these effects, we ask the second and third research question:

> **RQ_2: Environmental Noise** How much does **environmental noise** influence *ASR* performance?

> **RQ_3: Speaker Characteristics** How much do **speaker characteristics** influence *ASR* performance?

Systems We focus on using open-source *ASR* systems providing pre-trained models. The most popular open-source *ASR* projects, measured in github stars are *Kaldi*[1][Povey et al., 2011] and *Deep Speech*[2]. While *Kaldi* is a toolkit for speech recognition and implements various models and ASR related tools, *Deep Speech* implements a single deep learning-based model, which we discussed in Section 2.1.2. Because state-of-the-art *ASR* is largely dominated by commercial systems using proprietary algorithms and training data, we also include *Google Cloud Speech-to-Text*[3] as a reference. Building our own *ASR* model is out of scope for this thesis, and while we could have trained or fine-tuned existing models on our own data, this would have required a lot of computational resources and time. Additionally, the licenses for some of the commonly used training corpora are quite expensive.

Table 3.1 shows the three systems used in this study in more detail and compares the relevant features. The *Deep Speech* model uses a newer architecture than the *Kaldi* model and is trained on notably more data.

The direct comparison of the system's performance might seem unfair because of the wildly different techniques and amounts of data used in training. While this certainly influences overall performance, we are more interested in the differences in performance concerning background noise and speaker characteristics. In particular, we are interested in whether one of the systems is systematically more robust regarding these challenges than others.

[1] http://kaldi-asr.org/
[2] https://github.com/mozilla/DeepSpeech
[3] https://cloud.google.com/speech-to-text
[4] https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.2
[5] https://kaldi-asr.org/models/m1
[6] Accessed in November 2020

	Deep Speech	**Kaldi**	**Google**
Architecture	DNN	DNN-HMM	-
Model Version	0.9.2[4]	M1[5]	Default[6]
Training Data	Fisher, LibriSpeech, Switchboard, Common Voice, WAMU (NPR)	Fisher	-
Data Augmentation	Background noise and speech, pitch, tempo, volume etc.	Impulse responses and noises	-

Table 3.1: ASR systems used for the speech-to-text study.

Data For this study, we use a subset of the English *Common Voice*[7] dataset, which is part of an initiative by Mozilla to build a publicly available multilingual speech dataset. To this end, they use crowdsourcing to collect and validate the utterances of voluntary contributors. The resulting dataset features a wide range of speakers with different accents, of different ages and gender. In addition to the ground-truth label, utterances are also annotated with these speaker characteristics if the speaker agrees.

The subset we use for the study comprises 10 000 randomly sampled utterances, which have an average duration of 4,1 seconds and 8,4 words. As shown in Figure A.1, the distribution of speaker characteristics is roughly the same as reported for the whole dataset. *CommonVoice* does not represent the audio in web-video optimally, but to conduct a study on the influence of background noise and speaker characteristics, we need to be able to control these variables individually. There is no database containing speech in web-video with the needed annotations, so we chose to use an audio-only dataset for this preliminary study.

Evaluation We evaluate the transcription quality using *word error rate*, as computed by Equation 2.1. Before computing WER, we normalize both the hypothesis and the reference text using the following steps: removing punctuation, lowercase, stripping consecutive spaces, and splitting the text into words at spaces. The resulting word error rate is usually between 0 and 1, but it can be higher if the hypothesis is longer than the reference and contains many errors.

3.2 Results

RQ_1: Baselines To answer the first research question, we use the three systems listed in Table 3.1 to automatically transcribe *the Common Voice* subset described above. For

[7] https://commonvoice.mozilla.org/en (We use version 6.1)

Deep Speech, we use the default scorer (a KenLM language model) to re-score beams during hypothesis search. We evaluate the results in terms of word error rate.

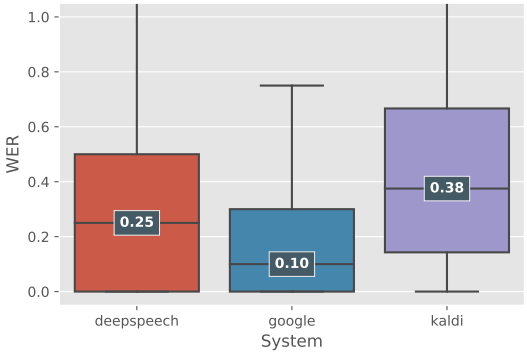

Figure 3.1: ASR performance of the systems under test on the *Common Voice* subset in terms of word error rate.

Figure 3.1 shows the results for each of the systems: Clearly, Google performs best with a WER median of 10%, followed by Deep Speech with a median of 25%, and *Kaldi* with a median of 37.5%. Looking at the quantity and diversity of training data available to the systems, the performance difference between them does not surprise.

The relationship between word error rate and usability of transcriptions for certain tasks has been studied previously. [Munteanu et al., 2006], for example, find that transcripts with a word error rate equal or less than 25% are acceptable for the task of question answering in webcast archives. More importantly, as mentioned in Section 2.1.3, [Hauptmann and Wactlar, 1997] find that for word error rates higher than 35%, spoken document retrieval performance starts to decline rapidly. Even without added background noise, with a median WER of 37.5%, the *Kaldi* model does therefore not perform well enough out of the box for spoken document retrieval. *Deep Speech*, on the other hand, could potentially be used for our task.

RQ_2: Environmental Noise To answer research question two, we evaluate the influence of environmental noise on *ASR* performance by mixing the samples of the *Common Voice* subset with noise from *MS-SNSD*[8] [Reddy et al., 2019] at different signal-to-noise ratios (SNR). SNR is defined in our case as the ratio of the power of a speech signal to the power of background noise. Because the mixed audio signals have a wide dynamic range, SNR is measured using the logarithmic decibel scale. We use mixtures with the following signal-to-noise-ratios: 0dB, 5dB, 10dB, 20dB and 40dB, where 0dB is a one-to-one mixture of speech and noise and the 40dB-mixture contains very little noise. We mix each sample in our subset of *Common Voice* with background noise, where the noises are randomly sampled from the *MS-SNSD* testset, which consists of realistic environmental noises such as air conditioners, background speech, or airport

[8] https://github.com/microsoft/MS-SNSD

announcements. We omit *Kaldi* in this experiment because of the bad baseline results. Because of limited resources, we apply the Google system to every other SNR mixture.

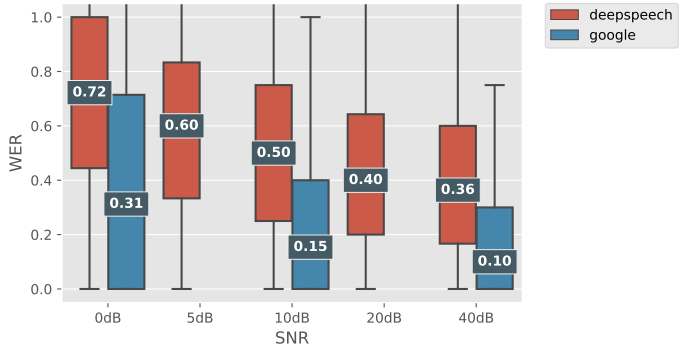

Figure 3.2: Influence of environmental noise on *ASR* performance.

Figure 3.2 shows the results in terms of WER in relation to the signal-to-noise ratio. We find that WER is approximately linearly related to the SNR of the mixture for both systems, while on a much lower level for the Google system.

RQ_3: Speaker Characteristics We cannot manually control the speaker characteristics independently of other factors like we control environmental noise in the previous experiment. Instead, we use the information provided by the speakers, which accepted the collection of meta-data. Therefore, we can only make statements about the correlation of speaker characteristics on transcription quality and not on their influence. We use the transcriptions of the baseline experiment but analyze WER grouped by speaker characteristics. We omit utterances without an annotation for the analyzed characteristic. Additionally, for the results to be somewhat meaningful, we only analyze utterances annotated with a characteristic that occurs more than 100 times in our testset, leaving us with six accents and six age groups.

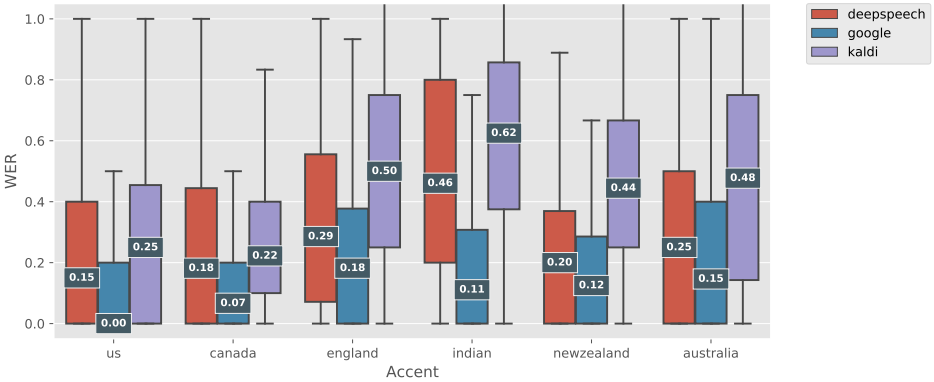

Figure 3.3: Correlation of accent and *ASR* performance.

Figure 3.3 shows the correlation of the accent of speakers to the word error rate. We find that WER differs wildly between accents for all systems, with lower values for US English and higher values for other accents. Some of the variability can be explained by the amount of training data available for the given accent. In particular, *Deep Speech*, which was trained partly on *Common Voice*, shows a WER distribution similar to the distribution of accents in the training data. In contrast to the other systems, the Google system performs rather well for Indian accents.

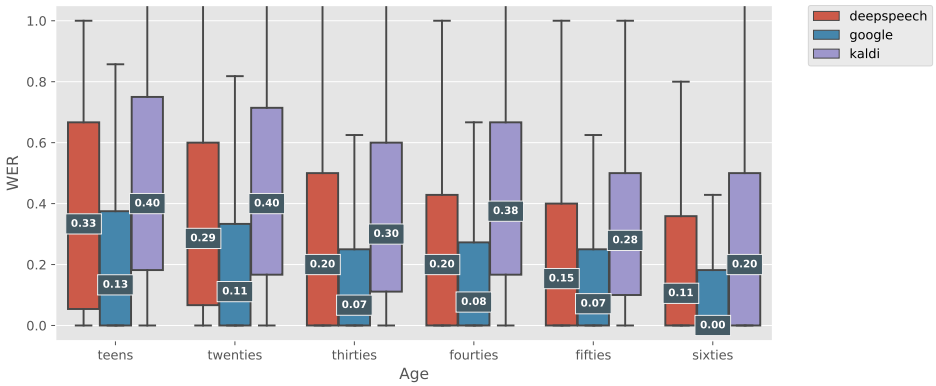

Figure 3.4: Correlation of age and *ASR* performance.

Figure 3.4 shows the correlation of the age of speakers to the word error rate. Again, word error rates differ between age groups for all systems, with older speakers generally resulting in better transcriptions. Given the gender of a speaker, the difference in terms of WER is small for all systems. Overall we find that word error rate differs wildly given different speaker characteristics for all current *ASR* systems.

4

Multi-modality Study

The speech-to-text study results show that even little environmental noise leads to poorer automatic transcription performance, which might influence retrieval performance. Speech enhancement has been successfully used to improve *ASR* in noisy conditions. In our case, traditional audio-only speech enhancement approaches can potentially be improved by using the video signal as a second input. Recently many deep learning-based audio-visual speech enhancement approaches have been proposed [Afouras et al., 2018a, Gabbay et al., 2018], but few evaluate if the enhanced speech improves *ASR* performance.

This section focuses on the available open-source audio-visual speech enhancement approaches and analyzes if they can be used to improve *ASR* performance in noisy conditions. In Section 4.1, we discuss the goals in more detail and show the data and systems used, while Section 4.2 presents the results. We also discuss the limitations of this particular study in Section 4.3, because they are relevant for Chapter 5. However, the overall limitations of this work are not discussed before Chapter 6.

4.1 Experimental Setup

Goals This second preliminary study's main goal is to evaluate to which extent current open-source audio-visual source enhancement approaches can be used to improve *ASR* performance in the context of general video retrieval. We therefore ask the following research question:

> **RQ$_4$** Can audio-visual speech enhancement help to improve automatic speech recognition performance?

We are interested in the quantitative improvements in terms of WER, but also in a qualitative assessment, in which cases *AVSE* helps, and in which it doesn't.

System In their overview, [Michelsanti et al., 2020] provide an extensive list of recent deep-learning-based audio-visual speech enhancement methods, which is also published on github[1]. While there are many promising approaches, few publish their code and

[1] https://github.com/danmic/av-se#audio-visual-speech-enhancement-and-separation

even fewer publish a pre-trained model. Table B.1 shows an overview of the approaches and whether they are open-source and publish a pre-trained model or not. Because implementing our own audio-visual speech enhancement model is out of scope, we are limited to the few available open-source approaches.

In this study, we use the approach presented in [Gabbay et al., 2018][2] which provide their implementation and a recipe to train a speech enhancement model. We train the model on 80% of the *GRID* corpus [Cooke et al., 2006], holding back 10% for validation and 10% for testing because there is no official train/test split. We use background noise from the *MS-SNSD* [Reddy et al., 2019] training set and mix it at an SNR of 5dB. [Gabbay et al., 2018] found that additionally using speech as background noise improves the models' performance, so we also mix clean speech of the *MS-SNSD* training set with the original speech from *GRID*.

Data & Evaluation We use the 10% of *GRID* held back for testing, mixed with noise and speech from the MS-SNSD testset for evaluation. We first apply the system described above to these mixtures to remove background noise and get an enhanced version as an output. Then, to evaluate the potential improvement of transcription quality provided by speech enhancement, we automatically transcribe the original speech sample, the mixture, and the enhanced version. For transcription, we use the same *Deep Speech* model as for the speech-to-text study, including the scorer. We then compare the relative improvement in terms of WER between the mixture and the enhanced version. We note that for a perfect speech enhancement system, WER would reach the level of the original speech sample.

4.2 Results

The GRID corpus seems very easy to automatically transcribe because of its limited vocabulary and little environmental noiseeems . With an ASR model trained on the same data, this is certainly the case, but unfortunately, the *Deep Speech* model performs very badly even on the clean samples without background noise. We assume this is due to a particularly specific vocabulary and a very high rate of speech. As we are not interested in the absolute value but in the relative improvement of WER between the noise mixture and the enhanced version, the bad transcription quality itself is not a big issue.

Figure 4.1 shows the distribution of word error rates for the three settings: The transcriptions of the unaltered video files of our *GRID* testset reach a median WER of 67%. The noisy mixtures are unusable for automatic transcription and yield a median WER of 100%. The transcriptions of the enhanced mixtures result in a WER of 87%, or in other words, speech-enhancement restored around 40% of the ASR performance. We therefore conclude, that audio-visual speech enhancement can potentially be used to improve automatic transcription quality to a certain extent.

[2]The original repository is no longer available, our fork lives at https://github.com/pypae/audio-visual-speech-enhancement

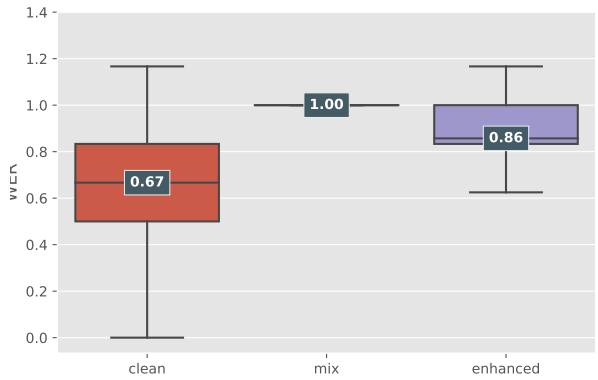

Figure 4.1: Transcription quality in terms of WER for the three settings: clean, mixed and speech enhanced.

4.3 Limitations

In Section 2.2.2, we talked about the narrow scope of *GRID*, consisting of only 30 speakers, a small vocabulary, and a static background. This is a major drawback because for the model to generalize well for any web-video, a dataset with a wider variety is needed. With *AVSpeech* [Ephrat et al., 2018] and *VoxCeleb2* [Chung et al., 2018], two datasets fulfill these criteria. To our knowledge, the semi-supervised approach of [Owens and Efros, 2018] is the only one providing a model pre-trained on one of these datasets. Their model, however, is trained for on/off-screen separation, which is closely related to speech enhancement. We experimented with their model and found it not to fit our use case because we want to include off-screen speech but reduce any on-screen background-noise. Because of the sheer size of the data - around 10TB for *AVSpeech* - it is unrealistic to train our own model reaching good performance for this work.

While we find that current audio-visual speech enhancement approaches can improve automatic transcription quality, certain conditions have to be met: As described in Section 2.2.2, most of the approaches track the faces of the speakers and only extract visual information from these regions. Therefore, for them to work, a speaker's face needs to be frontally visible and fully in frame, which is not the case for a large part of general web-videos. There are some approaches that use semi-supervised learning techniques to extract visual features, which could potentially be used more generally. But unfortunately, most audio-visual speech datasets are specifically made for the face tracking use-case and as a result, the unsupervised approaches might learn to focus on the same facial features.

To evaluate their approach, [Ephrat et al., 2018] compare their audio-visual speech enhancement approach to an audio-only approach. While they find the audio-visual model to perform better in separating multiple speakers, the audio-only approach yields the same enhancement quality for a single speaker with background noise.

5

Speech-based Video Retrieval

Given the results of the two preliminary studies, in this chapter, we investigate the feasibility of a speech-based video retrieval system based on currently available open-source components. Most of the existing speech-based video retrieval systems focus on a specific domain of videos: [Yang and Meinel, 2014], and later [Radha, 2016], for example, focus on the domain of lectures. The goal of our system, on the other hand, is to be as general as possible and work with web-video "in the wild". Therefore, we first collect a dataset of manually captioned web-videos. We then build a prototype of a speech-based video retrieval pipeline, experimenting with different settings, trying to improve retrieval performance.

This chapter is structured as follows: In Section 5.1 we first introduce our goals. Next, we focus on the data used, then we talk about the individual components of our retrieval prototype, and lastly, we discuss the evaluation techniques used. In the following Section 5.2, we present the results.

5.1 Experimental Setup

5.1.1 Goals

This study's main goal is to investigate the feasibility of a speech-based video retrieval system based on currently available open-source components. We build a baseline system and evaluate its performance both on known items and realistic user queries. We try to improve upon the baseline results, using speech enhancement and speaker diarization.

5.1.2 Data

Requirements The requirements a dataset for the evaluation of a speech-based video retrieval system must meet are twofold: First, the videos should represent general web-video "in the wild" as well as possible. As such, they should be of various formats, quality, and content. While one video might contain multiple speakers in frame, another one might be a voice-over without a visible speaker. To cover a wide range of video characteristics, the collection should be reasonably large. Second, the videos need to be reliably captioned, such that closed captions can be used as the ground-truth for what

was said when. Using the captions as text queries to the system allows the system to be evaluated using known-item search. There are several datasets designed for video-retrieval that satisfy the first requirement. [Rossetto et al., 2018] for example present *V3C*, a video collection of creative common attributed videos published on Vimeo[1]. While they collect multiple meta-data attributes, such as category, title, and description language, they do not provide closed captions. On the other hand, some large video datasets for audio-visual speech recognition, such as *LRS3-TED* [Afouras et al., 2018b] meet the second requirement. While these datasets often provide transcriptions and even word-alignments, they are mostly focused on single, visible speakers, and therefore do not represent web-video "in the wild".

Construction Due to the lack of a dataset meeting all our requirements, we construct our own. We base it on a study conducted by [Rossetto and Schuldt, 2017], which analyzes the metadata associated with more than 120 million *Vimeo* and *YouTube*[2] videos. In particular, we only consider *YouTube* videos with a creative commons attribution and use the number of closed captions associated with the video as an indicator if the video was captioned manually. More specifically, if a video has two or more associated closed captions, we assume that at least one was created manually. In contrast, others may either be manually created or automatically generated. For the videos with at least two captions, we then use the *YouTube Data API*[3] to determine if the audio language is English. If so, we download the video and the associated primary closed caption track using *youtube-dl*[4]. As the last step, because for some videos, the main captions are in another language than the video itself, we remove the videos where the main captions are not in English. The transcriptions should be treated with caution because there might still be some captions that are not actual transcriptions of the video's speech but rather descriptive captions.

Statistics Due to resource limitations, we conduct our study on a subset of only 1 743 videos, even though the whole dataset comprises more than 5 000 videos. The 1 743 videos are on average around 11 minutes long, with 75% being shorter than 12 minutes. We observe a long tail distribution, with some videos being longer than an hour. They contain a total of over 290 000 closed caption segments with an average duration of 3,5 seconds and an average word count of 9,4 words.

5.1.3 System

Overview Figure 5.1 shows an overview of the speech-based video retrieval prototype we built for this study. For the prototype, we manually pieced the individual components together, with the goal of them being potentially integrated into *vitrivr*. It is not a

[1] https://vimeo.com/
[2] https://www.youtube.com/
[3] More specifically, we use the `snippet.defaultAudioLanguage` attribute of the videos.
[4] https://github.com/ytdl-org/youtube-dl

complete retrieval system in the sense that new videos can automatically be processed and added to the index, but rather serves to evaluate the feasibility of a speech-based approach.

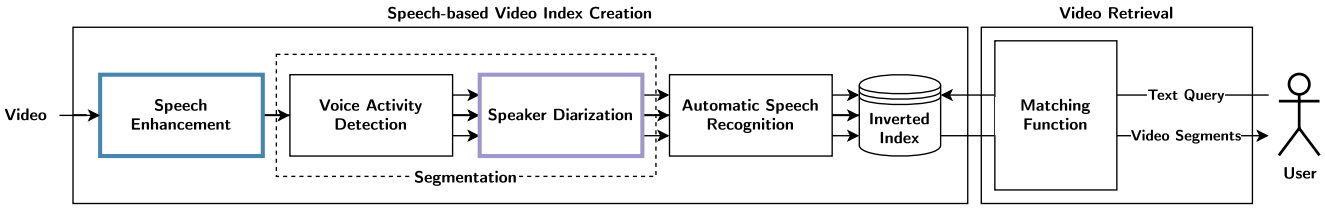

Figure 5.1: Overview of our speech based video retrieval prototype.

In this section, we discuss the individual components of this prototype and how they work together. We compare three systems and evaluate their retrieval performance individually: First, the "baseline" system does not include speech enhancement or speaker diarization. The second system, called "denoiser", includes speech enhancement to potentially improve ASR performance in noisy conditions, while the third system uses speaker diarization, trying to improve segmentation and enabling speaker re-identification.

Speech Enhancement In Chapter 3, we found that ASR performance is affected by environmental noise, and in Chapter 4, we found that audio-visual speech enhancement can potentially be used to improve ASR performance. We experimented with the audio-visual speech enhancement model of [Gabbay et al., 2018] described in Section 4.1, but due to the limitations of this approach described in Section 4.3, it does not generalize well enough to provide useful output on the web-video dataset. As a consequence, we use the audio-only speech enhancement approach of [Defossez et al., 2020][5], which subjectively performs much better. We first use *ffmpeg*[6] to extract the audio from the videos and then apply the *master64* pre-trained model.

Segmentation Most ASR systems cannot handle arbitrarily long audio streams but rather work with a fixed number of audio frames. We therefore need to segment the audio before transcription. A simple approach would be to split the audio into segments of a given duration, say ten seconds, but an obvious downside of this approach is that a segment boundary can fall in the middle of a word. Splits in sub-optimal locations most likely lead to recognition errors, so it's essential that the segments contain coherent speech, with distinct boundaries between segments. For retrieval, segmentation is also relevant for another reason: The segments should be semantically self-contained, and in the best case, correspond with the respective sentences.

We therefore use voice activity detection as segmentation for the "baseline" and "denoiser" systems. More specifically, we use *webrtcvad*[7], which is integrated into *Deep*

[5] https://github.com/facebookresearch/denoiser
[6] https://ffmpeg.org/
[7] https://github.com/wiseman/py-webrtcvad

Speech. For the "diarization" system, we use *pyannote.audio*, the speaker diarization approach explained in Section 2.3 as a segmentation step. As the 'pyannote.audio' model includes voice activity detection, we replace *webrtcvad* with this implementation. On the one hand, using speaker diarization allows for speaker re-identification within the same video. On the other hand, it has the potential to improve segmentation in multi-speaker scenarios.

Automatic Speech Recognition Arguably the most important part of the pipeline is the speech recognition system. Given the results of Chapter 3, we again use the same pre-trained *Deep Speech* model explained in Section 2.1.2, including the language model for re-scoring.

Index Creation To be able to query the speech segments more efficiently, we use an inverted index. We first aggregate segments into chunks of at least ten seconds to reduce possible errors from the previous segmentation step. We then use *whoosh*[8] to build the inverted index, storing the video id, the segment boundaries, the hypothesized transcription and, for the "diarization" system, the speaker. In more detail, per default, *whoosh* lowercases, tokenizes, and removes stopwords from the hypothesized transcription. To allow for fast retrieval, each segment is then indexed by the remaining tokens.

Retrieval To search the indexed documents, we query the index using *whoosh*. To this end, the text query is first processed using the same processing steps as used when indexing segments. The remaining terms are then used by the *BM25*[9] matching function, which estimates the relevance of video segments to the given search query using a bag-of-word approach. Meaning it ranks video segments based on the query terms appearing in each transcription, regardless of their proximity within the segment.

We additionally experiment with fuzzy search based on the *Levenshtein Distance* between the query and the hypothesized transcriptions to re-score the ranking result.

5.1.4 Evaluation

To evaluate our approach, we apply the system, as explained in Section 5.1.3 in its three configurations to the web-video dataset discussed in Section 5.1.2 to create an index. We then evaluate the retrieval performance by issuing multiple search queries.

Transcription Quality First, to assess the correlation between transcription quality and retrieval performance, we evaluate the quality of the transcriptions generated by the three systems in terms of *WER*. Because there is no distinct alignment between the manually annotated captions and the segments generated by our system, we don't compute the error rate on the segment level but the whole videos. In some cases, this

[8] https://github.com/mchaput/whoosh
[9] https://xapian.org/docs/bm25.html

can lead to extreme error rates because the manual captions are missing parts of the video.

Known-Item Search To evaluate the systems' performance in terms of verbatim dialog search, we automatically evaluate them using known-item search. To this end, we randomly sample 1 000 of the 290 000 closed caption segments and use them as search queries for our system. We assume each of these queries has exactly one relevant result: the video segment the closed caption is associated with. Therefore we can use the *Mean Reciprocal Rank (MMR)* as described in Section 2.1.3 as a retrieval performance measure. We limit the number of results per query to 5 and set the reciprocals of not retrieved segments to 0. In addition to *MRR*, we report the success rate at ranks 1 and 5. More precisely, for each query, we check if the search results contain the correct video segment for the given maximal rank.

Open Ended Search We also evaluate the "baseline" and "denoiser" system using realistic user queries. Because finding realistic user queries is quite challenging, we chose 10 arbitrary queries that are subjectively realistic. As a result, this evaluation is explorative and serves to find issues with our approach more than it is a performance measurement. We issue five term-based queries and five phrasal queries and subjectively rate the relevance of the results to compute *Discounted Cumulative Gain (DCG)* for the top 5 search results. We measure the relevance of a segment as follows: 1 for video segments containing the query, 0 for semantically similar video segments, and -1 for irrelevant segments. The resulting DCG_5 values indicate which system configuration performs better. Their absolute values range from $-2{,}95$ for no relevant results, to 2,95 for every result is containing the query.

5.2 Results

Each of the three configurations of our system corresponds to one of the research questions. To answer them, we apply each configuration to our dataset to build an index of video segments that can be searched. We then evaluate the performance as described in Section 5.1.4, beginning with the transcription quality.

Transcription Quality Figure 5.2 shows the transcription quality in terms of WER. We observe that the "denoiser" system, with a median WER of 52%, performs slightly better than the "baseline" system. The "diarization" system performs much worse than the other two systems. Because diarization itself should not negatively affect transcription quality, we attribute this performance difference to the different voice activity detection algorithm used for this system. Unfortunately, our dataset is not annotated with speech regions, so we cannot automatically measure VAD performance. Manually inspecting some of the segments, we found that VAD did indeed miss certain parts of speech and identified breaks as speech.

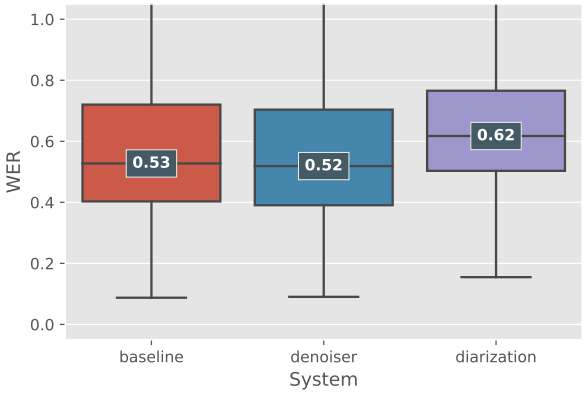

Figure 5.2: Overall word error rates for the three system configurations.

Known Item Search Table 5.1 shows the known item retrieval capability of the three system configurations. In addition to the three configurations to build the index, we evaluate three different matching functions. First, using the *BM25* matching function, the best system retrieves around 40% of the segments correctly at a rank at or below 5. In the next row, we show that using *Levenshtein Distance*-based fuzzy search to re-score the results increases retrieval performance for all systems. In the last row, we also count retrieved segments as correct if they appear in the same video but not at the correct position. The "denoiser" system consistently performs 2-3% better than the "baseline" in terms of success rate, meaning for the 1000 queries, it retrieves 20-30 more correct video segments than the "baseline".

	Baseline			**Denoiser**			**Diarization**		
	MRR	Best	Top 5	MRR	Best	Top 5	MRR	Best	Top 5
BM25	0.33	29.7%	37.3%	0.36	32.5%	40.3%	0.17	15.0%	19.5%
Fuzzy Search	0.41	38.7%	44.8%	**0.45**	**42.4%**	**47.8%**	0.20	17.6%	22.7%
Video Level Fuzzy Search	0.52	48.0%	57.5%	0.54	51.0%	59.0%	0.42	38.7%	48.7%

Table 5.1: Known item retrieval performance for the three system configurations.

Given these results, we're interested in whether the systems retrieve mostly the same segments or differ fundamentally. To this end, Figure 5.3 shows the retrieved segments of the "baseline" and "denoiser" system in a Venn diagram. Interestingly, the overlap of retrieved segments is smaller than expected, suggesting that combining the systems could greatly improve performance.

While we do not quantitatively evaluate the errors the system made for the not retrieved segments, we manually inspect a sample of the queries leading to negative results. Besides the expected ASR errors, we find three other common errors: First, *VAD* based segmentation is sometimes too aggressive, therefore some spoken words are missed. Sec-

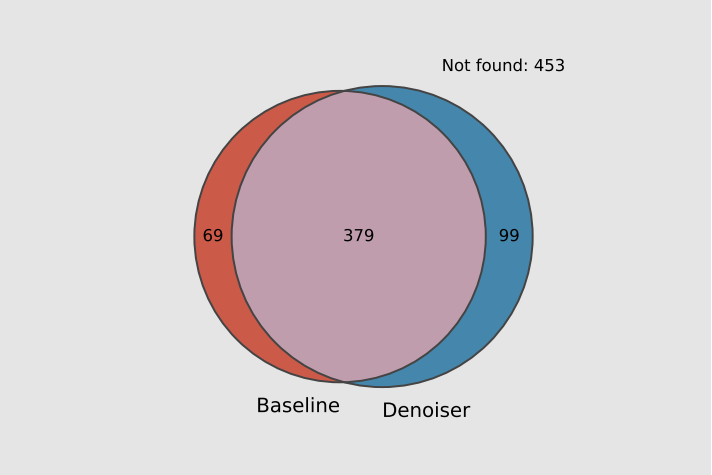

Figure 5.3: Known items retrieved in the top 5 by the Baseline and Denoiser systems using fuzzy matching.

ond, queries can contain specific words, which are not in the ASR systems vocabulary. The simple query "Minecraft" for example, does not produce any results. That is because the term does not have enough evidence in the ASR systems training data and produces "mine craft" or "mind craft" instead. Third, our dataset contains several wrong ground truth labels, which obviously cannot return the correct video segment when used as queries. As previously mentioned, captions manually created by humans can contain errors or be descriptive captions instead of actual speech transcriptions. In Chapter 6, we discuss possible approaches to address these challenges.

Open Ended Search Table 5.2 shows the queries used to qualitatively investigate the issues of the "baseline" and "denoiser" systems. We find that for the term based queries, the previously mentioned out-of-vocabulary (OOV) words are a major issue, while the queries not including OOV-terms returned mostly relevant results. We note that for the query "tornado", the "denoiser" system correctly retrieves an extremely noisy result, which the "baseline" misses.

The results for phrase-based queries are less clear. While some queries lead to mostly relevant results, others focus heavily on a single term. For "the cockpit is not answering their phone", most results either contain phone or answering, but never the two. As a consequence, we assume the queries might be too specific for our dataset.

Query	Topic	DCG_5 (higher is better)		OOV
		Baseline	Denoiser	
Terms				
easy cake recipe	baking	1.39	1.82	
lawnmower	gardning	-2.95	-2.95	✗
minecraft	gaming	-2.95	-2.95	✗
offside	soccer	-2.95	-2.95	✗
tornado	weather	1.75	1.95	
Phrases				
the cockpit is not answering their phone	news (9/11)	-2.95	-2.95	
my name is rob greenfield	specific person	2.13	1.74	
hands behind your back	crime	-1.95	-1.95	
I'm stuck in traffic	traffic	1.00	0.61	
breathe in and slowly breathe out	yoga	1.09	-0.54	

Table 5.2: Results for realistic user queries.

6

Limitations & Future Work

In this chapter, we discuss the limitations of this work and how future work can overcome them. We start with the general limitations of the whole thesis and then focus on individual components and improve them. We already discussed the limitations of audio-visual speech enhancement approaches in Chapter 4, so we do not focus on these again.

Pre-trained Models Due to the rather broad scope of this work as well as time and resource constraints, we mostly use pre-trained models for the various experiments. Fine-tuning the *ASR* model on the web-video domain, for example, on *LRS3-TED* [Afouras et al., 2018b], would probably increase the transcription performance. As *LRS3-TED* consists of TED talks, it is closer to the "web-video" domain, but still not sufficiently general enough. To our knowledge, there is no web-video dataset with high-quality transcriptions publicly available yet. Such a dataset would improve the transcription performance and enable other audio-visual speech processing research such as audio-visual speech enhancement and speaker identification.

Datasets We mainly use three datasets in this work: First, we use *CommonVoice*, a rather diverse dataset, to investigate the weaknesses of *ASR* systems. For the multi-modality study, we use *GRID*, a very narrowly scoped audio-visual dataset. Last, for speech-based video-retrieval, we use our own dataset based on manually captioned YouTube videos. While we could have used a single audio-visual dataset throughout this work, we chose to use them in accordance with the task. The two preliminary studies require a relatively clean reference to assess the influence of environmental noise, whereas the evaluation of speech-based video retrieval mainly needs a realistic dataset representing web-video "in the wild".

Segmentation In this work, we mainly focus on improving retrieval performance by improving *ASR* performance through pre-processing steps. We find that segmentation greatly impacts retrieval performance, an area in which future work is strongly needed. There are several possible approaches to do so, including VAD aggressiveness, segment overlap, and minimal segment length.

Matching Function We use *BM25* with fuzzy string matching as our matching function, which is common for queries to text-based document retrieval systems. In this context, often stemming and accent folding is used to improve robustness. To search for speech, other approaches to index the transcriptions by their sound might perform better. The most common phonetic algorithm to achieve this is *Soundex*[1], which aims for homophones to be encoded to the same representation.

Model Combination We use three different retrieval system configurations for indexing the speech of videos and find that retrieval improves when combined. While ensembles of multiple models are a common machine learning technique to improve performance, they might not be efficient enough for the purpose of retrieval. Another possible future approach is to adapt ASR decoding to return the top-k transcriptions instead of just the single best transcription and index the segment by all its transcriptions. Because normal beam search often leads to similar top-k hypotheses, *Diverse Beam Search* [Vijayakumar et al., 2018] could be employed to obtain a diverse set of transcriptions to index.

Towards End-to-end Retrieval Models Recently particularly popular in deep-learning became end-to-end models, which directly learn to predict from raw input and therefore replace manually engineered pipelines. In the context of this work, end-to-end audio-visual speech recognition models could potentially improve not only ASR performance but also retrieval performance. To take the end-to-end approach a step further, similar to *wav2vec* [Baevski et al., 2020], audio-visual speech embeddings could be computed. Such embeddings could then be fine-tuned to speech-based video retrieval, directly indexing the video segments by their embeddings and using a vector similarity measure for retrieval.

[1] https://www.archives.gov/research/census/soundex

7

Conclusions

The overall goal of this thesis was to identify and combine several state-of-the-art approaches for (audio-visual) speech enhancement, automatic speech recognition, and speaker diarization into a pipeline that is reliably able to determine what was said when given any video as input.

To this end, we first evaluated the performance of *Kaldi* and *Deep Speech* in the context of speech in web-video. We found that both are vulnerable to difficult speech recognition conditions common in web-video. In particular, environmental noise negatively influences automatic speech recognition performance despite data augmentation with noise during training. Additionally, transcription performance varies wildly between accents and age groups.

We then observed that to a certain extent, audio-visual speech enhancement can be used to improve ASR performance in noisy conditions. However, we could only show the improvement for lab-conditions, and the approach did not generalize well to web-video "in the wild". While there are many deep learning-based audio-visual speech enhancement approaches, most of them heavily rely on facial features.

Given the results of the two preliminary studies, we built a speech-based video retrieval prototype. To evaluate its performance, we collected a set of manually captioned YouTube videos. We compared three different configurations of the system to evaluate the effects of speech enhancement and speaker diarization. Our results showed that speech enhancement does improve known-item retrieval performance by 3%, leading to a retrieval rate of almost 50% on the very general dataset. By qualitatively evaluating the system, we identified several challenges for speech-based video retrieval and proposed possible approaches to solve them. Even though our results do not provide a fully reliable search for speech in videos, we identified several promising approaches that could greatly improve performance and provide a foundation for further research.

References

[Afouras et al., 2018a] Afouras, T., Chung, J. S., and Zisserman, A. (2018a). The Conversation: Deep Audio-Visual Speech Enhancement. In *Interspeech 2018*, pages 3244–3248. ISCA.

[Afouras et al., 2018b] Afouras, T., Chung, J. S., and Zisserman, A. (2018b). LRS3-TED: a large-scale dataset for visual speech recognition. In *arXiv preprint arXiv:1809.00496*.

[Afouras et al., 2019] Afouras, T., Chung, J. S., and Zisserman, A. (2019). My lips are concealed: Audio-visual speech enhancement through obstructions. In *INTERSPEECH*.

[Amodei et al., 2015] Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J., Fan, L., Fougner, C., Han, T., Hannun, A., Jun, B., LeGresley, P., Lin, L., Narang, S., Ng, A., Ozair, S., Prenger, R., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Wang, Y., Wang, Z., Wang, C., Xiao, B., Yogatama, D., Zhan, J., and Zhu, Z. (2015). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *arXiv:1512.02595 [cs]*. arXiv: 1512.02595.

[Ardila et al., 2020] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., and Weber, G. (2020). Common Voice: A Massively-Multilingual Speech Corpus. *arXiv:1912.06670 [cs]*. arXiv: 1912.06670.

[Baevski et al., 2020] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv:2006.11477 [cs, eess]*. arXiv: 2006.11477.

[Barker et al., 2018] Barker, J., Watanabe, S., Vincent, E., and Trmal, J. (2018). The fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines. *arXiv:1803.10609 [cs, eess]*. arXiv: 1803.10609.

[Bengio and Heigold, 2014] Bengio, S. and Heigold, G. (2014). Word Embeddings for Speech Recognition. In *Proceedings of the 15th Conference of the International Speech Communication Association, Interspeech*.

[Benoit, 1996] Benoit, C. (1996). Synthesis and automatic recognition of audio-visual speech. In *IEE Colloquium on Integrated Audio-Visual Processing for Recognition, Synthesis and Communication*, volume 1996, pages 1–1, London, UK. IEE.

[Bredin et al., 2019] Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., and Gill, M.-P. (2019). *pyannote.audio: neural building blocks for speaker diarization.* _eprint: 1911.01255.

[Chen et al., 1976] Chen, C., Recognition, J. W. o. P., Artificial Intelligence (1976, Hyannis, M., Recognition, W., and Intelligence, A. (1976). *Pattern Recognition and Artificial Intelligence: Proceedings of the Joint Workshop on Pattern Recognition and Artificial Intelligence, Held at Hyannis, Massachusetts, June 1-3, 1976.* Academic Press Rapid Manuscript Reproduction. Academic Press.

[Chuang et al., 2020] Chuang, S.-Y., Tsao, Y., Lo, C.-C., and Wang, H.-M. (2020). *Lite Audio-Visual Speech Enhancement.* _eprint: 2005.11769.

[Chung et al., 2018] Chung, J. S., Nagrani, A., and Zisserman, A. (2018). VoxCeleb2: Deep Speaker Recognition. In *INTERSPEECH*.

[Chung and Zisserman, 2016] Chung, J. S. and Zisserman, A. (2016). Lip Reading in the Wild. In *Asian Conference on Computer Vision*.

[Collobert et al., 2016] Collobert, R., Puhrsch, C., and Synnaeve, G. (2016). Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. *arXiv:1609.03193 [cs]*. arXiv: 1609.03193.

[Cooke et al., 2006] Cooke, M., Barker, J., Cunningham, S., and Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424.

[Craswell, 2009] Craswell, N. (2009). Mean Reciprocal Rank. In Liu, L. and Özsu, M. T., editors, *Encyclopedia of Database Systems*, pages 1703–1703. Springer US, Boston, MA.

[Defossez et al., 2020] Defossez, A., Synnaeve, G., and Adi, Y. (2020). *Real Time Speech Enhancement in the Waveform Domain.* _eprint: 2006.12847.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

[Ephrat et al., 2018] Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. T., and Rubinstein, M. (2018). Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics*, 37(4):1–11.

[Gabbay et al., 2018] Gabbay, A., Shamir, A., and Peleg, S. (2018). Visual Speech Enhancement. *arXiv:1711.08789 [cs, eess]*. arXiv: 1711.08789.

[Gao et al., 2018] Gao, R., Feris, R., and Grauman, K. (2018). Learning to Separate Object Sounds by Watching Unlabeled Video. In *ECCV*.

[Godfrey and Holliman, 1997] Godfrey, J. J. and Holliman, E. (1997). Switchboard-1 Release 2. type: dataset.

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 369–376, Pittsburgh, Pennsylvania. ACM Press.

[Gupta et al., 2017] Gupta, A., Miao, Y., Neves, L., and Metze, F. (2017). Visual Features for Context-Aware Speech Recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5020–5024. arXiv: 1712.00489.

[Hannun, 2017] Hannun, A. (2017). Sequence Modeling with CTC. *Distill*, 2(11):10.23915/distill.00008.

[Hannun et al., 2014] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014). Deep Speech: Scaling up end-to-end speech recognition. *arXiv:1412.5567 [cs]*. arXiv: 1412.5567.

[Hansen and Hasan, 2015] Hansen, J. H. L. and Hasan, T. (2015). Speaker Recognition by Machines and Humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99.

[Harte and Gillen, 2015] Harte, N. and Gillen, E. (2015). TCD-TIMIT: An Audio-Visual Corpus of Continuous Speech. *IEEE Transactions on Multimedia*, 17(5):603–615.

[Hauptmann, 2006] Hauptmann, A. (2006). Spoken Document Retrieval, Automatic. In Brown, K., editor, *Encyclopedia of Language & Linguistics (Second Edition)*, pages 95 – 103. Elsevier, Oxford, second edition edition.

[Hauptmann and Wactlar, 1997] Hauptmann, A. G. and Wactlar, H. D. (1997). Indexing and search of multimodal information. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 195–198 vol.1.

[Heafield, 2011] Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

[Heckmann et al., 2001] Heckmann, M., Berthommier, F., and Kroschel, K. (2001). A Hybrid ANN/HMM Audio-Visual Speech Recognition System. page 6.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*, 9:1735–80.

[Huang and Deng, 2010] Huang, X. and Deng, L. (2010). An Overview of Modern Speech Recognition. In *Handbook of Natural Language Processing, Second Edition, Chapter 15 (ISBN: 1420085921)*, pages 339–366. Chapman & Hall/CRC, handbook of natural language processing, second edition, chapter 15 (isbn: 1420085921) edition.

[Järvelin and Kekäläinen, 2002] Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446. Place: New York, NY, USA Publisher: Association for Computing Machinery.

[Lowerre, 1976] Lowerre, B. T. (1976). *The Harpy Speech Recognition System.* PhD Thesis, Carnegie Mellon University, USA.

[Mcgurk and Macdonald, 1976] Mcgurk, H. and Macdonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264(5588):746–748.

[Michelsanti et al., 2020] Michelsanti, D., Tan, Z.-H., Zhang, S.-X., Xu, Y., Yu, M., Yu, D., and Jensen, J. (2020). An Overview of Deep-Learning-Based Audio-Visual Speech Enhancement and Separation. *arXiv:2008.09586 [cs, eess].* arXiv: 2008.09586.

[Munteanu et al., 2006] Munteanu, C., Penn, G., Baecker, R., Toms, E., and James, D. (2006). Measuring the acceptable word error rate of machine-generated webcast transcripts. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 1.

[Owens and Efros, 2018] Owens, A. and Efros, A. A. (2018). Audio-Visual Scene Analysis with Self-Supervised Multisensory Features. *arXiv:1804.03641 [cs, eess].* arXiv: 1804.03641.

[Palaskar et al., 2018] Palaskar, S., Sanabria, R., and Metze, F. (2018). End-to-End Multimodal Speech Recognition. *arXiv:1804.09713 [cs, eess].* arXiv: 1804.09713.

[Panayotov et al., 2015] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

[Patel and Meshram, 2012] Patel, B. V. and Meshram, B. B. (2012). Content based video retrieval systems. *International Journal of UbiComp*, 3(2):13–30. arXiv: 1205.1641.

[Petajan et al., 1988] Petajan, E., Bischoff, B., Bodoff, D., and Brooke, N. M. (1988). An improved automatic lipreading system to enhance speech recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '88*, pages 19–25, Washington, D.C., United States. ACM Press.

[Potamianos et al., 2012] Potamianos, G., Neti, C., Luettin, J., and Matthews, I. (2012). Audiovisual automatic speech recognition. In Bailly, G., Perrier, P., and Vatikiotis-Bateson, E., editors, *Audiovisual Speech Processing*, pages 193–247. Cambridge University Press, Cambridge.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. event-place: Hilton Waikoloa Village, Big Island, Hawaii, US.

[Preethi, 2017] Preethi, J. (2017). Automatic Speech Recognition - An Overview.

[Rabiner and Juang, 1986] Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16.

[Radha, 2016] Radha, N. (2016). Video retrieval using speech and text in video. pages 1–6.

[Reddy et al., 2019] Reddy, C. K., Beyrami, E., Pool, J., Cutler, R., Srinivasan, S., and Gehrke, J. (2019). A Scalable Noisy Speech Dataset and Online Subjective Test Framework. *Proc. Interspeech 2019*, pages 1816–1820.

[Rix et al., 2001] Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 2, pages 749–752 vol.2.

[Rossetto et al., 2016] Rossetto, L., Giangreco, I., Tanase, C., and Schuldt, H. (2016). vitrivr: A Flexible Retrieval Stack Supporting Multiple Query Modes for Searching in Multimedia Collections. In *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pages 1183–1186, Amsterdam, The Netherlands. ACM Press.

[Rossetto and Schuldt, 2017] Rossetto, L. and Schuldt, H. (2017). Web Video in Numbers - An Analysis of Web-Video Metadata. *arXiv:1707.01340 [cs]*. arXiv: 1707.01340.

[Rossetto et al., 2018] Rossetto, L., Schuldt, H., Awad, G., and Butt, A. A. (2018). V3C - a Research Video Collection. *arXiv:1810.04401 [cs]*. arXiv: 1810.04401.

[Roth et al., 2020] Roth, J., Chaudhuri, S., Klejch, O., Marvin, R., Gallagher, A., Kaver, L., Ramaswamy, S., Stopczynski, A., Schmid, C., Xi, Z., and Pantofaru, C. (2020). Ava Active Speaker: An Audio-Visual Dataset for Active Speaker Detection. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4492–4496.

[Schneider et al., 2019] Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised Pre-training for Speech Recognition. *arXiv:1904.05862 [cs]*. arXiv: 1904.05862.

[Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[Smith, 2011] Smith, J. O. (2011). *Spectral audio signal processing*. W3K, Stanford, Calif. OCLC: 776892709.

[Snyder et al., 2018] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

[Sumby and Pollack, 1954] Sumby, W. H. and Pollack, I. (1954). Visual Contribution to Speech Intelligibility in Noise. *The Journal of the Acoustical Society of America*, 26(2):212–215.

[Vijayakumar et al., 2018] Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., and Batra, D. (2018). *Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models*. _eprint: 1610.02424.

[Yang and Meinel, 2014] Yang, H. and Meinel, C. (2014). Content Based Lecture Video Retrieval Using Speech and Video Text Information. *IEEE Transactions on Learning Technologies*, 7(2):142–154.

[Yin et al., 2018] Yin, R., Bredin, H., and Barras, C. (2018). Neural speech turn segmentation and affinity propagation for speaker diarization. In *Annual Conference of the International Speech Communication Association*, Hyderabad, India.

[Yu and Deng, 2015] Yu, D. and Deng, L. (2015). Introduction. In *Automatic Speech Recognition*, pages 1–9. Springer London, London. Series Title: Signals and Communication Technology.

[Zhang et al., 2019] Zhang, A., Wang, Q., Zhu, Z., Paisley, J., and Wang, C. (2019). Fully supervised speaker diarization. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6301–6305. IEEE.

[Zhang and Zhang, 2009] Zhang, E. and Zhang, Y. (2009). Average Precision. In LIU, L. and ÖZSU, M. T., editors, *Encyclopedia of Database Systems*, pages 192–193. Springer US, Boston, MA.

[Zhao et al., 2018] Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., and Torralba, A. (2018). The Sound of Pixels. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, volume 11205, pages 587–604. Springer International Publishing, Cham.

A

Common Voice Speaker Characteristics

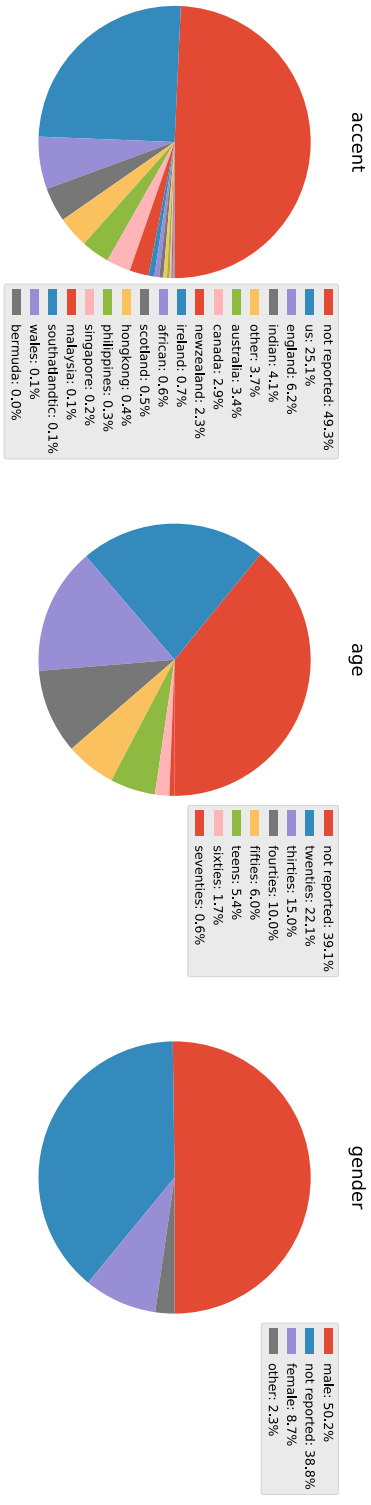

Figure A.1: Distribution of speaker characteristics in the subset of *Common Voice*.

B

Overview of Deep Learning-Based Audio-Visual Speech Enhancement Approaches

Approach	Code (Official)	Pretrained Model
Audio-Visual Speech Enhancement		
[Afouras et al., 2018a]	✗	✗
[Ephrat et al., 2018]	✗	✗
[Gabbay et al., 2018]	✓	✗
[Afouras et al., 2019]	✗	✗
[Chuang et al., 2020]	✓	✗
Related Approaches		
[Zhao et al., 2018]	✓	✗
[Gao et al., 2018]	✓	✗
[Owens and Efros, 2018]	✓	✓

Table B.1: Deep-learning based audio-visual speech enhancement approaches.

List of Figures

List of Tables