Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Bachelorarbeit (Informatik)

Evaluation of Automatic Speech Recognition Systems

Autoren	Fabian Germann Malgorzata Anna Ulasik
Hauptbetreuung	Prof. Dr. Mark Cieliebak
Datum	07.06.2019

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw School of Engineering

Erklärung betreffend das selbständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

……………………………………

Unterschriften:

…………………………………………………………...

……………………………………………………………

……………………………………………………………

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

1 Abstract

Powerful computing platforms, advances in deep learning and increased amount of data available have resulted in a rapid progress in automatic speech recognition (ASR) technology and an increase in the speech-to-text quality. The consequence of this development is increasing demand for ASR-based solutions. Tech companies developing ASR systems are in constant pursuit of reaching the best possible transcription quality. Our study provides an in-depth accuracy comparison of the current state-of-the-art solutions, including among others Google Speech-To-Text, IBM Watson and Microsoft Azure. The analysis is based on transcriptions of several well-known speech corpora such as Timit or Switchboard. Apart of the comparative system evaluation, we also perform an analysis of correlations between spoken language properties and the transcription accuracy and investigate how well the standard evaluation metric – WER – reflects the actual transcription quality. The evaluation results show, that the proprietary cloud solutions outperform the open source system in practically all domains, with the solutions of Google, Microsoft and Amazon in the lead. We also observe that the most challenging speech recognition tasks are related to translating spontaneous conversational non-native speech. Our fine-grained WER analysis provide evidence that this metric does not always reflect the proportion of information preserved in the transcription.

2 Zusammenfassung

Leistungsstarke Rechenplattformen, Fortschritte in Deep Learning und zunehmende Datenmengen führten zu grossen Fortschritten in automatischer Spracherkennung (ASR - Automatic Speech Recognition) und verbesserter Speech-to-Text-Qualität. Zunehmende Nachfrage für ASR-Lösungen sind die Folge. Technologiekonzerne die ASR-Systeme entwickeln sind stets bemüht, die bestmögliche Transkribierungsqualität zu erreichen. Unsere Studie bietet einen detaillierten Vergleich aktueller State-of-the-Art-Lösungen, darunter Google Speech-To-Text, IBM Watson und Microsoft Azure. Die Analyse basiert auf Transkriptionen mehrerer bekannter Sprachkorpora wie Timit oder Switchboard. Neben dem Systemvergleich führen wir eine Analyse der Korrelationen zwischen gesprochenen Spracheigenschaften und der Transkriptionsgenauigkeit durch und untersuchen, wie gut die Standardmetrik WER die tatsächliche Transkriptionsqualität widerspiegelt. Die Resultate zeigen, dass die proprietären Cloud-Lösungen die Open-Source-Systeme in praktisch allen Bereichen übertreffen, angeführt von Google, Microsoft und Amazon. Wir stellten ebenfalls fest, dass die schwierigsten Aufgaben der Spracherkennung mit der Übersetzung spontaner Konversationssprachen von nicht Muttersprachlern zusammenhängen. Unsere detaillierte WER-Analyse bestätigt die Annahme, dass diese Metrik nicht immer den Anteil der in der Transkription erhaltenen Informationen widerspiegelt.

3 Preface

Working on this thesis was an interesting experience and a challenging task. We would like to thank Prof. Dr. Mark Cieliebak for his support. We appreciate the numerous inspiring discussions we had throughout the project and the guidance we received from him.

Contents

4 Introduction

ASR has become an important technology to enable and improve the interactions between humans and between humans and machines. Applications, such as in-meeting assistants, voice search, virtual speech assistants, intelligent living room devices, or dictation tools facilitate our life and work. Due to advances in deep learning, powerful computing platforms and increased amount of data available, speech technology has made rapid progress recently and an essential improvement in the Speech-to-Text (STT) quality has been observed. This has resulted in increasing demands for ASR-based solutions.

ASR has been an intensive research area for decades and the need for improving speech recognition accuracy has driven researches and experiments all over the world. Tech companies developing ASR systems are in constant pursuit of achieving the lowest possible error rate.

In this thesis we conduct an in-depth comparison of the performance of existing speaker-independent Large-Vocabulary Continuous Speech Recognition (LVCSR) systems for English language. Taking various perspectives throughout our study, we hope to provide practitioners with a valuable input facilitating the selection process of an ASR tool. At the same time, we aim at identifying the correlations between transcription accuracy and spoken language variabilities, such as dialectal variations, accents or speaking rate. Further, we perform a fine-grained analysis of systems' error rates and this way try to understand how well Word Error Rate (WER) reflects the actual transcription accuracy.

4.1 Literature Review

The accuracy of automatic speech recognition has been one of the important research challenges.

There is a variety of research papers concerned with different aspects of ASR evaluation: from articles presenting evaluation techniques and metrics, throughout papers discussing factors affecting speech recognition accuracy to literature documenting comparative evaluations of ASR systems.

González et al. in [22] introduces a methodology to accomplish the evaluation of different speech recognition systems in several scenarios including the creation of new Speech Corpora of different types. [11] provides an overview of techniques for evaluating speech recognition systems, and explains subjective and objective evaluation methods. McCowan et al. in [37] introduces a definition of an evaluation metric and explains its desired properties.

Numerous papers are devoted to discussing existing and proposing alternative or additional speech recognition evaluation metrics. Much attention is focused on limitations of most commonly used metric: WER. Some articles explain the need for a more meaningful performance measure than WER, describing alternative metrics such as Keyword Error Rate (KER) [51], Relative Information Loss (RIL), Word Information Loss (WIL), Weighted Keyword Error Rate (WKER) [14], Command Success Rate (CSR) [6], or Word Information Preserved (WIP) [37]. However, none of the articles provides evidence that the alternative metrics are capable of distinctly outperforming the WER.

Widely investigated is the impact of speech properties on speech recognition accuracy. Numerous researches were conducted in order to understand how spoken language variability may impede ASR performance and how these impediments can be reduced by applying new or optimized STT technologies. Among others, there are studies concerned with speech disfluency. Goldwater et al. in [21] investigates which of disfluency aspects is most challenging for recognition by comparing the accuracy of two English ASR systems for conversational telephone speech. [32] introduces algorithms to automatically identify and remove repetitions and filled pauses and provides evidence that these algorithms significantly improve the recognition accuracy.

Accented speech and dialectal variations are also subject to multiple research papers. They provide evidence that accent and dialect have negative impact on ASR accuracy. [59] investigates how native and non-native speech can be distinguished, explores methods of detecting and adapting to non-native speech. [4] describes approaches for recognizing both the regional dialect and accent of a speaker. [58] and [62] present techniques for improving the recognition accuracy of accented speech. [46] investigates techniques for compensating for the effects of accents on performance of selected ASR technologies. Much attention is focused on overlapping speech, one of the largest challenges for ASR. [20] deals with overlapping speech detection, while [18] presents metrics extensions and algorithm adaptations for evaluating ASR in the presence of overlapping speech.

There are also studies focused on the correlation between the speaking rate and ASR accuracy. [56], [16] and [55] discuss the impact of fast and very slow speech on ASR accuracy, stating that increased error rates can be observed in both cases.

Apart from the studies focusing on particular speech properties and their impact on ASR, there are also evaluations aiming at comparing performance of multiple ASR systems. However, most of them were conducted more than three years ago, hence do not reflect current state-of-the-art.

[33] is a comparative evaluation of three systems: Microsoft API, Google API, and Sphinx-4. The testing set comprises selected utterances from TIMIT and International Telecommunication Union (ITU). The evaluation is based on the calculation of Word Error Rate and results in a ranking where Google API is in the lead with a WER of 9%, followed by Microsoft (18% WER) and Sphinx-4 (37% WER).

[17] presents a large-scale evaluation of open-source speech recognition toolkits. HTK in association with the decoders HDecode and Julius, CMU Sphinx with the decoders pocketsphinx and Sphinx-4, and the Kaldi toolkit are compared in terms of usability and expense of recognition accuracy. The evaluation basis is a Verbmobil 1 (VM1) corpus containing dialogue speech in three languages (English, Japanese and German) in the appointment scheduling task as well as Wall Street Journal 1 (WSJ1) corpus containing English read speech derived from Wall Street Journal news. The best result is achieved by Kaldi (12.7% and 6.5% WER on VM1 and WSJ1 acordingly), followed by HDecode, pocketsphinx, Sphinx-4 and Julius with the worst result of 27.2%on VM1 and 23.1% on WSJ1 .

There are further research papers documenting comparative studies, such as [34], [61] and [40]. These studies focus, however, on evaluating ASR systems performance on very specific recognition tasks or in very specific domains. [34] provides a comparison of the performance of automated transcription services (Google, IBM, Microsoft, Trint, YouTube) in the domain of dyadic medical teleconsultation. [61] focuses on an ASR evaluation in translation dictation and medical dictation workflows. It presents a performance comparison of a speaker-adapted ASR system installed on a laptop PC and a speaker-independent ASR system in a remote server accessible through a mobile device. [40] investigates ASR systems' suitability for use in different types of dialogue systems. The comparison is performed on 5 systems: Pocketsphinx, Apple, Google, AT&T and Otosense-Kaldi.

The scale of the evaluation documented in the present thesis is larger than the studies presented in the research papers mentioned above. We conduct a comparison of 7 ASR systems based on 9 standard corpora, which results in a set of 63 Corpus-System-Pairs. This allows a broader and more in-depth evaluation. By applying a large data set and by combining a praxis-oriented approach with solid theoretical foundations derived from the current state of research, we create a solid basis for ASR systems evaluation.

4.2 Outline

We organize the thesis into five main parts. We devote the introductory Chapter 4 to presenting the context of this thesis, explaining its goals and giving a brief overview of the current state of research. Chapter 5 introduces the theory of automatic speech recognition and ASR performance evaluation. Chapter 6 describes the experimental setup comprising evaluation scope, scenarios and process, while chapter 7 presents the results of the evaluation and discusses conclusions derived from the study. In Chapter 8, we summarize our observations related to ASR evaluation.

4.3 Terminology

There is a number of ASR-related terms and acronyms which are used throughout this thesis. This chapter introduces the most essential ones with their meanings relevant for this study. For definitions of other terms not included below, please refer to the Glossary at the end of the report.

Automatic Speech Recognition (ASR)	a technology enabling machines to process speech input and translate it into text. It is also known as Speech Recognition and Speech-to-text (STT).
LVCSR System	Large-Vocabulary Continuous Speech Recognition System is a system for automatically recognizing speech produced as a continuous stream. Its vocabulary is not limited to a particular domain. Large-Vocabulary Continuous Speech Recognition, if applied as a speaker-independent solution is the most challenging tasks in the field of speech recognition.
Speech Corpus	a collection of digital recordings of speech together with their annotations, meta data, and documentation.
Utterance	a unit of speech which is provided as input for speech recognition. It can be a single word, a phrase, a complete or incomplete sentence, or multiple sentences.
Reference	a manual transcription of an audio recording which serves as a reference for ASR evaluation.
Hypothesis	a transcription of an audio recording performed by an ASR system.
Alignment	the process of comparing the reference text with its transcript in order to identify confusion pairs (pairs of words where the transcription was incorrect and a substitution, deletion or insertion has occurred).
Evaluation Metric	a measure used for evaluation of an ASR system. It should be objective and clearly interpretable. Most well-known ASR evaluation metric is Word Error Rate.
Word Error Rate (WER)	the edit distance between a reference and its automatic transcription, normalised by the length of the reference.

5 Foundations

5.1 Automatic Speech Recognition

There have been attempts at automatically recognizing speech since the 1950's. The very first systems used simple pattern matching mechanisms that compared the acoustic input signal with reference sound patterns to identify spoken words in their entirety. Those systems did not have any knowledge apart from this pattern matching mechanism such as an understanding of syntax or semantics. This worked relatively well on very small vocabularies of a handful of words, because even if the pronunciation of the words was not consistent — the limited number of candidates were still sufficiently distinct. However, the words needed to be uttered clearly separated from each other in order that the system could identify where one word/pattern begins and another ends. Recognition of anything close to continuous speech was therefore far from reality. The first speech recognition system of this art was "Audrey". A system engineered at Bell Laboratories in 1952 that could understand digits with 90% accuracy — if spoken by its inventor HK Davis.

The next iteration were rule-based systems that did not only compare patterns but also applied rules to determine the spoken words — language-specific grammatical and syntactic rules for example. With this the systems started to incorporate real knowledge instead of just comparing patterns. Even though this did provide advancement, it didn't result in any major progress. Speaker variations and inconsistencies in the language make it hard to define rules that work for general use cases — vocabulary and speaker wise respectively.

The late 70s finally brought substantial progress to the field with the crucial introduction of statistical modelling. Before it was be lived that in order to build a good ASR system the system also needs to have thorough understanding of context — what is said, who says it etc. Jim and Janet Baker were the first to apply Hidden Markov Model (HMM) to the domain of speech recognition in 1975. This approach will dominate speech recognition systems from the 80's until the turn of the century and even today — even though the applied underlying algorithms have changed quite a bit — ASR systems are highly statistical. The following section will give a basic overview of fundamental probability theory of speech recognition.

5.1.1 Probability Theory of Speech Recognition

The overall biggest challenge in speech recognition is the variation n speech. The same sequence of words can be expressed through an infinite amount of different audio signals. These differences can stem from the speaker itself (accent, dialect, etc.) or noise introduced into the signal. For this reason the main paradigms of ASR take a statistical approach the 80's: The systems do not produce a definite transcript but rather one or more most likely transcriptions.

Equation 1 shows a more formal description of the problem of speech recognition[28]. For an acoustic input O with discrete observations $o_1, o_2, \ldots o_t$, the most likely symbol sequence $\widehat{W}$ out of all valid sequences $W = w_1, w_2, \ldots w_n$ in the language $\mathscr{L}$ can be described as:

$$\widehat{W} = \operatorname*{argmax}_{W \in \mathscr{L}} P(W|O) \tag{1}$$

By applying Bayes' rule we get probabilities that are, for the most part, easier to compute than $P(W|O)$ itself:

$$\widehat{W} = \operatorname*{argmax}_{W \in \mathscr{L}} \frac{P(O|W)P(W)}{P(O)} \tag{2}$$

$P(W)$ — the prior probability of the word sequence — can easily be estimated by examining the distribution of possible word sequences in the given language $\mathscr{L}$. $P(O|W)$ — the likelihood of the acoustic input O given the word sequence W — can be estimated using HMMs. The remaining $P(O)$ is actually not that trivial to calculate. But since $P(O)$ is dependent only on the acoustic input O, and therefore the same for every candidate sequence W, we can simply ignore it. This leaves us with:

$$\begin{aligned}\widehat{W} &= \operatorname*{argmax}_{W \in \mathscr{L}} \frac{P(O|W)P(W)}{P(O)} \\ &= \operatorname*{argmax}_{W \in \mathscr{L}} P(O|W)P(W) \qquad (3)\end{aligned}$$

In the next section we will cover the architecture of a traditional statistical ASR system and we will see how both parts of Equation 3 correspond to specific components of this architecture.

5.1.2 Architecture of a Typical Statistical Speech Recognition System

In this section we will introduce the architecture of a traditional statistical speech recognition system. The described methods are not necessarily used exactly like this anymore by state of the art systems, but most systems derive from this initial design and oftentimes still share various components with it. We will later discuss how newer systems evolved from this architecture and how they differ.

Parts of the system – namely its models – need to be created before a speech signal can be transcribed into text. We won't discuss how this training works here and concentrate on the process of transcription. See [28] for a more extensive introduction to this topic.

The transcription process and the components involved are visualized in Figure 1. It starts out with the acoustic analysis which takes the speech signal, splits it up and extracts features. Then the (pretrained) acoustic model uses these features and produces sequences of phones and corresponding probabilities. Now the decoder takes these phone sequences, translates them into lexical text using the phonetic model and predicts the most likely transcripts for the initial input. It also uses the language model to get language specific context (pretrained as well).

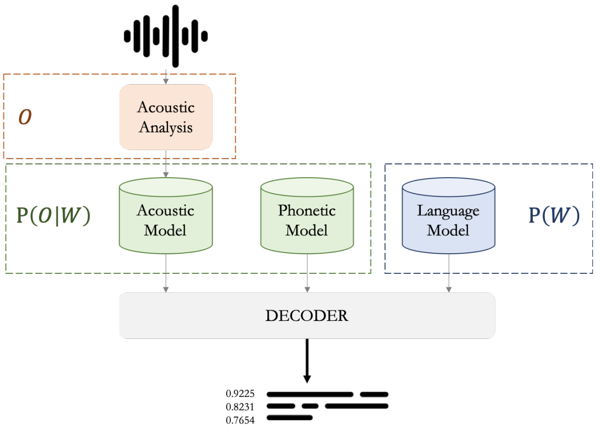


Figure 1: Typical statistical ASR decoding architecture

This was a very condensed overview of the whole process. The following paragraphs will go a little more into the details of each component.

Acoustic Analysis

The acoustic analysis component receives the raw speech signal and preprocesses it for the subsequent acoustic model. As a first step the audio signal is split into discrete samples or frames. Afterwards feature extraction is applied to each of those frames.

The size of the frames can vary, but usually ranges somewhere from 10 - 25 milliseconds[29]. When choosing the frame size, the aim is for the speech signal inside the frames to be stationary. This is necessary to effectively extract features that are representative of the frame.

The feature extraction process is very involved and there are many different approaches. Whatever features are chosen, they should exhibit the following properties[30]:

- Enable the system to distinguish between different but similar sounding speech sounds.
- Allow for the automatic creation of acoustic models without requiring excessive amounts of training data.
- The information gained from the features should be largely invariant across speakers and speaking environments respectively..

The most common acoustic feature representation in ASR is the Mel-frequency cepstral coefficients (MFCC). The MFCC feature extraction multi-step procedure that is modeling the human auditory perception system[42].

Acoustic Model

The acoustic model takes the extracted features from the acoustic analysis and maps them to a base unit. The choice of the right base unit is crucial for the performance of the system. Using words might seem intuitive at first glance. But this would introduce the same problems already mentioned when we discussed the early pattern matching approaches: Words can be pronounced in too many different ways for this to be effective with even moderately sized vocabularies — not to speak of continuous speech. Another problem with the choice of words is that the model would only be able to recognize words it has seen during training. New words, mispronounced words et cetera would not be recognized.

A much better choice for the base unit are phone or phoneme respectively. Phoneme are for speech what the alphabet is for text. But instead of covering the textual space, they covers the sound space[29]. One major advantage of phonemes over words is that the number of phonemes in a language is much smaller. Most languages have around 20 - 60 phonemes [29] and their number is set and does not change as with words. The problem that the trained model could encounter an unknown phoneme is therefore nonexistent. And because every word — existing, non-existing or even mispronounced — is made up of those same phonemes, an acoustic model would be able to process text it hasn't has seen before by inferring.

Now that the base unit has been set, the feature vectors extracted by the acoustic analysis need to be mapped to these phonemes. Since the 1980's statistical models based on HMMs have been the most popular choice by far for this task. We won't go into further detail of how HMMs themselves work as this is best explained thoroughly by itself. But we will discuss how HMMs are applied to speech — based on the excellent description in [28].

In speech the hidden states of the HMM are the phones and the observations are the extracted features. Figure 2 shows the schematic display of an HMM for the word *speech* which consists of the phones `[S]`, `[P]`, `[IY]` and `[CH]`. Each phone corresponds to a state s_i and for each transition from a state s_i to a state s_j there is a transition probability a_{ij}. Each feature vector (observation) from the acoustic analysis corresponds to an observation x_i and each state s_i has emission probabilities b_{ij} to emit an observation x_j.

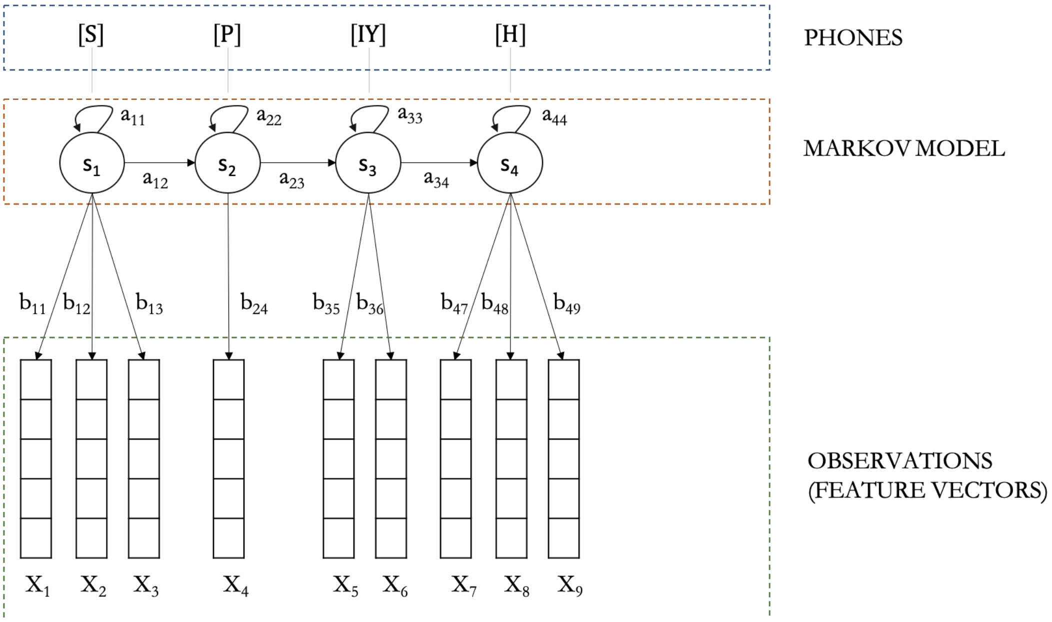


Figure 2: HMM model for ASR

The transmission probabilities a_{ij} from one state to another can be learned from training data. The emission probabilities for each state to specific observations (feature vectors) on the other hand is not quite so simple to caculate. As the feature vectors consist of real valued numbers it's not as easy as to just count occurences. There are different approaches to solve this problem, but a very popular one is the use of Gaussian Mixture Models (GMMs)[28]. The combination of HMMs and GMMs for acoustic modelling is often called a *GMM-HMM Model*.

It is possible, even very likely, that a single phone/state emits multiple observations. This is because the (speech) duration of a phone can be much larger than the sample size chosen in the acoustic analysis. [28] has observed `[z]` phones that vary in length from 7 milliseconds to more than 1.3 seconds. Or 1 to 130 frames (and therefore feature vectors) with a sample rate of 10 milliseconds.

It's important to note that this is strongly simplified summarization. For example: Phones are not actually always pronounced the same way. The pronunciation very much depends on the context and other variables. In practice therefore each phone has it's own HMM consisting of usually 3 states: A start state, a middle state and an end state respectively[28]. Furthermore Figure 2 only shows transitions to itself or a single subsequent state. In reality each state can of course transition into various other states (with different transition probabilities). Unlike other HMM applications the HMM models used for speech recognition do not allow arbitrary transitions between states but have rather strong constraints. They do not allow to transition to earlier states but only to themselves or successive ones. These types of HMM are also called Bakis network[28].

After transforming the raw speech signal into features, and calculating phone sequences of different likelihoods, we somehow need to turn this phone sequences into actual lexical text. This is done using the pronunciation model which is discussed in the next section.

Pronunciation Model

The pronunciation model is used to transform the sequence of phones from the acoustic model into lexical text. It usually is a simple mapping from words to one or more pronunciations. This is the only model in a typical ASR system that is not learned from data, but rather created by linguistic experts[29]. While the model is simple in its structure, the creation of large pronunciation models is a highly labour intensive task and requires a lot of expertise. It is one of the biggest challenges to overcome, when trying to apply an existing ASR system to a new language, as good pronunciation models aren't that freely available for every language.

A very good resource for the English language is the Carnegie Mellon University Pronouncing Dictionary (CMUdict). It is an open source dictionary providing pronunciations for over 134000 North American English words. Listing 1 shows a little excerpt from the CMUdict.

```
BIOTECH  B AY1 OW0 T EH2 K
BIOTECHNICA  B AY2 OW0 T EH1 K N IH0 K AH0
BIOTECHNICA'S  B AY2 OW0 T EH1 K N IH0 K AH0 Z
BIOTECHNOLOGICAL  B AY2 OW0 T EH2 K N AH0 L AA1 JH IH0 K AH0 L
BIOTECHNOLOGIES  B AY2 OW0 T EH2 K N AA1 L AH0 JH IY0 Z
BIOTECHNOLOGY  B AY2 OW0 T EH2 K N AA1 L AH0 JH IY0
BIOTECHNOLOGY'S  B AY2 OW0 T EH2 K N AA1 L AH0 JH IY0 Z
```

Listing 1: Excerpt from the CMUdict

Translating the phoneme sequence into words is of course not as straightforwards as just taking phonemes and looking up the word. Each sequence could multiple results.To combat this problem a language model is used which we discuss in the next section.

Language Model

The job of the language model is to provide context. The acoustic model might score multiple phone sequences very similar. And that is probably correct because for every acoustic signal there are multiple possible phone sequences that would likely result in a very similar pronunciation. Also the succession of the phones of all these candidate sequences might be very likely to occur in the acoustic space. However, this does not necessarily mean that all of these phone sequences can be translated into intelligible text. Take the following two phone sequences for example:

```
R EH K AH G N AY Z S P IY CH
R EH K AH   N AY   S B IY CH
```

Both sequences are very similar and would therefore be pronounced very similar. These phone sequences are actually the phonetic spelling of the following texts:

```
Recognize Speech
Wreck a nice beach
```

Now humans can instantly identify the first example as a much more likely transcription because we know that the second one doesn't make sense. This is because we understand the context. We know what these words actually mean and can with almost certainty say, that the first transcription is the correct one. At least when the choice is between those two. The ASR system does not have this knowledge though. It does not really understand the meaning of those words. It therefore needs the language model to obtain this context. To understand what succession of words make sense — are likely to occur.

Just like the acoustic model, the language model needs to be trained in advance for later use in the transcription process. This is done on very large text data of the required language. Of course this means that the language model too is specific to the language it was trained on and new language models are required for transcribing different languages.

Language models are not specific to the field of speech recognition but have many more application areas in the area of Natural Language Processing (NLP) such as Machine Translation, Optical character recognition (OCR), Spelling Correction, Dialog Generation etc[29]. Consequently there are many existing tools solving this problem. Three prominent toolkits for building language models are the widely used SRI Language Modeling (SRILM) Toolkit, the newer, for large data quantities optimized KenLM Toolkit and the OpenGrm NGram Library which uses finite-state transducers (FSTs).

The next and final component of the system — the decoder — uses the now introduced models to find best transcription for the raw acoustic signal provided at the beginning.

Decoder

The decoder is the final component of the system that brings everything together. It uses the acoustic, phonetic and language model respectively to find the word sequence that best matches the input signal. In subsubsection 5.1.1 we defined the following equation for the speech recognition problem:

$$\widehat{W} = \underset{W \in \mathscr{L}}{\operatorname{argmax}}\ P(O|W)P(W) \tag{3 revisited}$$

The decoder is the component that solves this equation and determines $\widehat{W}$ using the acoustic and pronunciation model ($P(O|W)$) as well as the language model ($P(W)$).

Combining the information of all these models results in an enormous search graph: The language model gives probabilities for combination of words, each word itself consists of phonemes which have their own probabilities for being combined and then each phone has its own HMM. Even for medium sized vocabularies of 40'000 words — medium sized for LVCSR that is — the search graph amounts to a size of tens of millions of states[29]. Performing an exact search on this graph is impossible — even for todays state of the art systems. The transcription is therefore determined by using approximate search techniques. Common examples for such techniques are Viterbi decoding using Viterbi approximation and Weighted Finite State Transducers (WFSTs)[38], [28].

This concludes the exploration of the traditional stastistical ASR system architecture. In the next section a short overview of newer approaches will be presented.

5.1.3 Hybrid and end-to-end Systems

The traditional HMM based ASR systems discussed in the previous sections has been the state of the art for a long time in LVCSR. Even today some systems still use this approach such as the CMUSphinx sphinx4 system included in this evaluation. However, more often than not today's systems differ quite a bit from this approach. While now ASR systems are dominated by Deep Neural Networks (DNNs), this is far from a new idea. Already back in the 90s research was done regarding application of Artificial Neural Networks (ANNs) in speech recognition and it continued to gradually take over more and more components of the traditional ASR architecture[13]. Today every component we discussed can be replaced with an neural network approach: N-gram language models can be replaced with natural language models. The GMM-HMM acoustic model can be replaced by DNN-HMMs or -HMMs. During acoustic analysis convolutional neural networks can be used for raw signal processing and even pronunciation models have neural network based alternatives today.

Most state of the art systems today use this hybrid approach. However, for a few years now a new paradigm has emerged: The so called end-to-end systems. The idea is that instead of using all these different neural networks as individual components that need individual training, one neural network should replace the whole process. The motivation behind this is that it would remove a lot of complexity and such systems would facilitate the application to a new language as neither a pronunciation model nor a language model would be needed. The probably most famous architecture of this kind is Baidu's DeepSpeech[23], [2]. Mozillas DeepSpeech system is based on Baidu's architecture.

Now, while there is a lot of talk about end-to-end systems and there are systems that claim to be end-to-end, thus far there isn't any system that is truly end-to-end as we have specified above. All end-to-end systems that perform similarly well as Systems with a more traditional architecture still need at lest language model. So at the time they are more striving towards the end-to-end system than already being one.

5.2 Recognition Tasks Classification

Speech recognition tasks can be classified into four categories according to two criteria. Firstly, it can be distinguished between recognition targeting human to human or human to computer utterances. Secondly, the utterances to be recognized can either have a dialogue or a monologue style [8]. Table 1 lists typical tasks that are representative for each category.

	Dialogue	**Monologue**
Human to human	Category I: Meeting minutes, interviews, calls.	Category II: broadcast news, news programs, lectures, presentations, and voice mails
Human to machine	Category III: voice-command tools	Category IV: dictation

Table 1: Categorization of speech recognition tasks derived from [8]

The Category I targets human-to-human dialogues. The recognition task refers to transcriptions of meetings minutes, interviews or calls, which requires processing human–human conversational speech under unpredictable recording conditions and vocabularies, sometimes containing also emotional speech. This task presents a large challenge for spoken language processing.

Tasks belonging to the Category II, which aim at recognizing human monologues for human audience, are represented by transcriptions of broadcast news, news programs, lectures, presentations, and voice mails. Since in the Category II, the speakers need to ensure that they are understood by the audience in one-way communication, the task of recognizing their speech is relatively easier than transcribing utterances of the spontaneous dialogue speech.

Most of the practical application systems widely used now are classified into the Category III, which relates to recognizing utterances in dialogues between human and computer. Voice-command tools are a representative of this class. Unlike other categories, the design and development of systems in Category III is usually preceded by a clear definition of the system application. The set of human messages to be recognized is finite in number, each being associated with a particular action (e.g., route a call to a proper destination).

Category IV targets the recognition of monologues performed when people are talking to computers. One of the typical tasks belonging to this category is dictation. Since the utterances to be recognized in this case are made with the expectation that the speech will be converted exactly into texts with correct characters, its spontaneity is much lower than those in Category III.

Various researches have provided evidence that utterances spoken by people talking to computers, especially when the people are conscious of computers, are acoustically, as well as linguistically, very different from utterances directed towards people, such as those in Categories I and II. Among the four categories, spontaneity is considered to be the highest in Category I and the lowest in Category IV.

Apart from the classification based on distinction between human and machine actors and dialogue and monologue speech, speech recognition tasks can also be classified according to speech mode (isolated word recognition or continuous speech recognition), vocabulary size (large, medium or small), speaker mode (speaker-dependent or speaker-independent ASR) and speaking style (spontaneous and dictation). Figure 3 provides an overview of the speech recognition tasks according to the classification in [31].

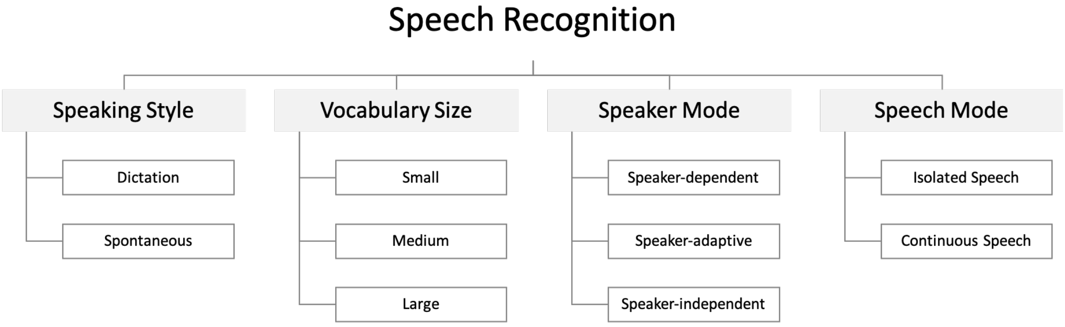


Figure 3: Speech recognition tasks according to the classification in [31]

In isolated word recognition, the word boundary detection (segmentation) and recognition is much easier than if the words are connected. The beginning and the end of each word can be detected directly from the energy of the signal.

Speaker-independent recognition is more difficult than speaker-dependent recognition, since the speech model must be general enough to cover all types of voices and all possible ways of word pronunciations, and yet specific enough to discriminate between individual words. For a speaker-dependent system, training or adaptation of speech models is carried out by using utterances of each speaker. In speaker adaptation, the system is bootstrapped with speaker-independent models, and then gradually adapts to the specific aspects of the speaker.

From among the recognition task categories depicted above the speaker-independent LVCSR is the most challenging tasks in the field [14]. It is due to a number of factors, including unlimited vocabulary, speakers' articulation variety, dialects, accents, variable speaking rates, and high degree of acoustic variability caused by acoustic environment diversity and recording device variety. LVCSR systems are subject to the following comparative evaluation.

5.3 Corpora

Speech corpora constitute the prime source of data for research and development in the area of automatic speech recognition. They are collections of digital recordings of speech together with their annotations, meta data, and documentation. [54] There are various applications of speech corpora: some build a basis for training language models, some are used specifically for acoustic modelling, some are applied for developing and evaluating ASR systems. Corpora are usually split into two data sets: one for the training phase and one for the testing phase. The training material is used to set the model parameters of the recognition system. The testing material is used to determine the performance quality of the trained system. The differentiation between training and testing speech data is essential for ensuring a fair evaluation of the system performance.

5.3.1 Speech Corpus Selection

Criteria for corpus selection are strongly related to the recognition task of the ASR system (see subsection 5.2 for details on recognition task categories). The degree to which a corpus exhibits required properties determines its utility as a development or evaluation corpus for a particular system. It is due to the fact that the presence of irrelevant variations may degrade recognition performance. When a system is designed for isolated word recognition, an isolated words corpus should be applied. Similarly, when a system is designed for telephone speech, it should be developed with the use of a telephone speech corpus. On the other hand, speaker-independent LVCSR systems require high diversity of the speech material. A large number of utterances with variability in parameters like recording conditions, speakers' gender, age, accent or dialect are necessary to cover a wide range of real speech conditions that may occur. [45] These requirements provide the reason, why good quality documentation and meta data is essential for successful corpora selection and application. On one hand, the corpus must be provided with a documentation of the applied transcriptions and annotations rules. On the other hand, it must contain speech material meta data, such as speakers' profile information comprising among others speaker's mother tongue, dialect, age, gender and education level and recording details, such as environmental conditions, room acoustics, sources of noise or recording device. [54]

5.3.2 Standard Speech Corpora

The availability of speech corpora is a central issue due to the difficulties and the cost of collecting and manually annotating a corpus with transcriptions. It is a commonly applied practice to use standard speech corpora for both development and evaluation of speech recognition technologies. [7] Several data sets, available through the Linguistics Data Consortium (LDC) and European Language Resources Association (ELRA), have become standards for the development and evaluation of ASR algorithms. [50]

English corpora most widely distributed by LDC used for speech recognition purposes are: TIMIT (acoustic-phonetic a continuous speech corpus for development and evaluation of ASR systems), Web 1T 5-gram Version 1 (for statistical language modelling), CELEX2 (for pronunciation modelling), TIDIGITS, (for designing and evaluating algorithms for speaker-independent recognition of connected digit sequences), Switchboard-1 Release 2 (large vocabulary conversational speech corpus), English Gigaword Fifth Edition (for language modelling) [36] and WSJ1 (containing English read speech comprising Wall Street Journal news). Corpora available through ELRA applicable for ASR systems development are among others Accented English GlobalPhone (utterance-segmented read speech from the newspaper domain) and several versions of English British SpeechDat (recordings of British English speakers from different regions, recorded over fixed and mobile telephone network). Multiple other speech corpora can be acquired from other sources, such as: Common Voice (a data set consisting of read-aloud speech data), AMI (a multi-modal data set consisting of meeting recordings), LibriSpeech (large-scale corpus of read English speech), SWC (Spoken Wikipedia Corpora, data set comprising read-aloud Wikipedia articles), Tedlium (English speech corpus comprising TED talks recordings), Voxforge (read English speech corpus) and WSJCAMO (a British English speech corpus for large vocabulary continuous speaker-independent speech recognition systems).

The following evaluation is based on a set of corpora selected from the data sets listed above. The selection criteria together with an overview of the selected corpora are presented in section 6. For corpora detailed documentation please refer to the Appendix A Corpora Documentation.

5.4 ASR Systems Evaluation

The purpose of evaluating ASR systems is to measure the systems' performance on recognition task and to judge the usefulness of the systems' output. Evaluation methods aim at providing a comparison criterion between different systems or ASR technologies. [14]

5.4.1 Evaluation Techniques and Metrics

Techniques for evaluating speech recognition can be classified into subjective or objective methods. Subjective methods directly involve humans during measurement. They are more suited to evaluating applications with higher semantic or dialogue content. However, their downside is the fact that human subjects cannot reliably perform quality measurement and that they cannot handle fine-grained measurement scales. Objective methods do not directly involve human subjects and have the advantage of producing reproducible results. Moreover, their essential advantage is their automation. However, on the other hand it is difficult to create methods with the capacity to cope easily with the complex processes required for evaluating speech understanding or end application usability. [11] In the following we focus on the challenges related to the objective evaluation metrics.

According to McCowan et al. [37] an evaluation metric should ideally have four properties. First, it must measure directly and merely the performance of the ASR component, even if the analysis is performed with respect to the overall end usability. Second, the measure should be calculated in an objective, automated manner. Third, the measure must be clearly interpretable in the context of end application usability. The absolute value of the measure must reflect system performance. Finally, while the evaluation measure should be general, it should be modular to allow a thorough application-dependent analysis. Different end applications place different relative importance on particular words and associate different costs with different types of errors. For instance, an application in which an alarm is generated if one of a set of keywords is spoken. In this case, only a small subset of all spoken words is important, and the balance between false alarms and missed alarms will depend on the relative costs of reacting to, or missing, the alarm. In a spoken document retrieval system, the relative importance of each word will depend on the information it carries with respect to the particular application context, and usability may be hampered more by missing information than by erroneous insertions. In a dictation application, all words may be considered to be of equal importance, and missed words be just as costly as falsely inserted words. Any application-oriented evaluation framework should therefore allow the performance to be analyzed in terms of individual words or particular types of errors.

The performance of an ASR system can be investigated in terms of speed, which is measured with the real time factor [31] or can be measured in terms of error rate related to accuracy. The latter measure is the most frequent one and is discussed below in detail.

5.4.2 Word Error Rate

The standard approach to evaluate the performance of a speech recognition system in terms of accuracy is the WER. The word sequence provided by the ASR system is aligned with a reference text, and the number of errors is computed as the sum of substitutions (S), insertions (I), and deletions (D). [1] The alignment is performed based on the Levenshtein distance, or edit distance. The edit distance between two strings is the minimum number (or weighted sum) of insertions, deletions and substitutions required to transform one string into the other. So, the WER is the edit distance between a reference and its automatic transcription, normalized by the length of the reference. This normalization is applied to allow comparison between different systems on different tasks, it results from the fact that the magnitude of the edit distance depends on the string length. [37] The WER is computed as follows:

$$WER = \frac{I + D + S}{N} \cdot 100 \tag{4}$$

where I is the number of insertions, D is the number of deletions, S is the number of substitutions and N is the total number of words in the reference transcription.

When reporting the performance of a speech recognition system, sometimes the Word Recognition Rate (WRR) is used instead [31]:

$$WRR = 1 - WER = \frac{N - I - D - S}{N} \tag{5}$$

There are two major shortcomings of the WER. One is that the word error rate is difficult to interpret. It is due to the normalization by N. While the edit distance (the numerator of the WER) itself has clear interpretation as an accumulated cost, normalizing by the reference sequence length is problematic as the numerator is not bounded by N due to the inclusion of insertions. This means that in practice the word error rate may exceed unity, or equivalently that the WRR mentioned above may be negative. This property means that it is often difficult to interpret the meaning of the absolute value of the WER, or to make relative comparisons between two different rates. [37]

Another disadvantage of WER is that it is sometimes not easily interpretable in terms of the overall ASR-based application usability. [37] It is clear why WER is important for evaluating the quality of a transcript and that no better alternative likely exists if the task is speech transcription for its own sake, but there are doubts around its applicability to downstream uses of ASR. Speech Recognition is widely used inside spoken utterance retrieval systems, spoken language understanding, speech summarizers, and speech-to-speech translation systems, where human users never see the transcripts themselves. So a question occurs: what is the harm in transcribing some words wrongly if the user never sees the transcript and the performance of the task remains unchanged. [15] In certain domains, at least, relatively high-WER transcripts have been shown to be perfectly usable [44].

Hence, there is interest in the speech recognition community to consider evaluation measures that allow a better understanding of system performance than WER, particularly in terms of end application usability. McCowan et al. in [37] provide an interesting example derived from [60], which shows that improvements in spoken language understanding can be obtained while a significant increase in the WER can be observed. Similarly, in spoken document retrieval applications, it has been shown that high WER do not necessarily lead to any significant degradation in retrieval performance [19].

5.5 ASR Alternative Evaluation Metrics

A number of metrics alternative to or extending WER can be found in the literature. This section provides an overview of these measures and provides some conclusions drawn from their comparison with WER.

There are alternative measures proposed as response to the first limitation of WER: the difficulty to interpret the meaning of WER absolute value. One of them is Word Correct Rate (WCR), applicable particularly for isolated word recognition systems. It does not consider insertion errors, hence eliminates the WER first disadvantage. [37] The lack of a lower bound in WER, and the consequent asymmetry with respect to insertions and deletions is also discussed in [41], where a WIP metric is proposed. It is an approximate measure of mutual information between the reference and automatic transcriptions. It has comparable simplicity to WER as it is a simple function of hits, substitutions, deletions and insertions. But unlike WER, WIP is a true percentage, and it approximates directly the proportion of information preserved.

The proportion of information preserved is also subject to several other research papers. Hence Park et al. in [51] discuss the concept of KER in the context of applying ASR systems for speech analytics. As many speech analytics applications rely on identifying keywords in the transcripts, their performance should be more sensitive to keyword errors than regular word errors. The conclusion of the comparison between KER and WER is a statement that values for both metrics are similar in higher accuracy transcripts, but KER increases more rapidly than WER as the transcription accuracy deteriorates.

In [48], Nanjo et al. propose WKER as an evaluation metric for keyword-based open-domain speech understanding. It is based on Term Frequency-Inverse Document Frequency (TF-IDF) criterion and gives a weight on errors from a viewpoint of Information Retrieval.

Another measure inspired by information retrieval is precision and recall proposed by McCowan et al. in [51] The speech recognition evaluation is perceived as an Information Retrieval task, in which each word occurrence is treated as a unit of information, and which aims at retrieving the relevant information from the reference in the automatic transcription. In the proposed framework there are only two types of errors, insertions (false alarms) and deletions (false rejections). A substitution error is a co-occurrence of these two. In terms of information content, a substitution error represents both a loss of relevant information as well the retrieval of erroneous information, and thus is considered as both a deletion and an insertion error. This leads to counting substitutions twice which according to the authors is the proper approach, if the information content of the words is considered in the context of an end application.

Morris et al. in [28] introduce two measures for evaluating applications in which the proportion of word information communicated is more meaningful than edit cost. These metrics are MER (match error rate) and WIL (word information lost). MER is the probability of a given match being incorrect, while WIL is an approximation to the proportion of word information lost. Both metrics provide a measure which vary from 0 when there are no errors to 1 when there are no hits. At low error rates the two measures and the WER give similar scores, however, in case, where significant error rates are common, the rankings given by each of the three measures start to differ significantly.

All of the metrics are attempts to overcome the disadvantages of WER and find a more accurate measure for applications where proportion of word information communicated is more relevant than edit cost. In two of the studies, it has been shown that in areas which typically involve high error rates, measures alternative to WER seem more suitable. However, the WER score is very well established and as the last decades of research has shown, it cannot be easily replaced by a new measure. Despite its downsides and multiple alternatives discussed in the literature, it remains a standard applied in studies and papers in the research area and a measure to compare performance of existing WER systems in the industry.

5.6 ASR Systems Evaluation Tools

One of the most well-known applications to evaluate speech recognition is sclite, an open-source tool integrated in the Scoring Toolkit (SCTK) developed by National Institute of Standards and Technologies (NIST). The metric used by sclite for evaluating speech recognition is WER. The tool first performs an alignment of a manual transcription with the automatic transcription obtained from the ASR, which is followed by scoring, consisting in gathering statistics such as number speakers, words, hits, insertions, deletions and substitutions and calculating the WER. The performance of the system is then summarized in a variety of reports. [49]

Apart of sclite, there are other evaluation tools available as open-source solutions, such as jiwer and asr-evaluation. jiwer is a Python package to approximate the WER of a transcript. It computes the minimum-edit distance between the reference sentence and the hypothesis sentence of an ASR system. The minimum-edit distance is calculated using the Wagner-Fisher algorithm, which computes it on the character-level. [26]

asr-evaluation is another Python package for evaluating ASR. It uses the editdistance (Python module for computing edit distances and alignments between sequences), for computing the edit distance between the reference and the hypothesis. The program measures the ASR performance by calculating WER, WRR and Sentence Error Rate (SER), which is a ratio of incorrect sentences to total number of sentences. [3]

6 Experimental Setup

6.1 Evaluation Objectives

The evaluation presented in this thesis follows three main goals. The first objective is praxis-oriented: We investigate the performance of several existing speaker-independent large-vocabulary continuous speech recognition systems in order to provide practitioners with inputs facilitating the selection process of an ASR tool. When evaluating, we consider four recognition tasks according to the classification provided in subsection 5.2: human-human dialogue (transcribing of meetings or interviews) and human-human monologue (transcribing of lectures and presentations), human-machine dialogue (voice-command tools) and human-machine monologue (dictation tools). The second goal of the evaluation is to analyse correlations between various spoken language variabilities such as accented speech, dialectical variation and disfluencies and the transcription accuracy. We seek to understand which from among the investigated factors present the greatest challenges for ASR technology. Finally, we perform a fine-grained analysis of the error rates of the systems under investigation in order to better understand the accuracy level of the transcripts and at the same time discover how well WER reflects the actual transcription quality.

6.2 Corpora Selection and Integration

The corpora selection was based on public availability and applicability for evaluating speaker-independent LVCSR systems. The selection process was performed in consideration of the speech recognition tasks classification provided in subsection 5.2, and with regard to the diversity of spoken language properties required for speaker-independent LVCSR evaluation. Following these two goals, we focused in our selection process on factors such as variety of speaking styles (spontaneous speech, semi-spontaneous and read-aloud, monologues and dialogues), and variety of speakers (with accented speech and dialectal variations). The selection was preceded by literature research in order to identify data sets commonly applied in ASR evaluations.

Eventually nine English corpora were selected to build a basis for the evaluation: three corpora comprising spontaneous dialogue speech, one consisting of semi-spontaneous monologue utterances and five containing read-aloud monologue speech. The evaluation was performed on the test sets of the corpora. Most of the corpora are provided with a default split into a training set and a test set. In case of corpora without a pre-defined test set, a random selection of utterances of total duration of around 5 hours was performed. Table 12 provides an overview of the main corpora properties.

The final data set contains 70’875 utterances with a duration of 75.4 hours in total. The utterances come from 1131 speakers, native and non-native, female and male and speaking various dialects. For details on the corpora properties see A Corpora Documentation.

6.3 Systems Selection and Integration

6.3.1 Systems Selection

The objective of the system selection is to enable the integration of a good number of systems in the given time constraints, while still ensuring a variation of systems suitable for different use cases. We decided to include both proprietary cloud services as well as offline open source systems.

The cloud services provide ease of use and enable us to integrate multiple systems in reasonable a time frame. The services also provide pretrained models out of the box without the need to invest a lot of time into training. In this category we decided on the cloud services of Google,

Corpus Name	**Test Set**	**Test Set Duration**	**Speaking Style**	**Number Speakers**	**Accented Speech**	**Dialectal Variation**	**Filled Pauses**	**Overlapping Speech**
AMI	Random selection	5h	Dialogue spontaneous speech	38	Yes	Yes	Yes	Yes
Common Voice	Default test set	5h	Monologue read-aloud speech	Unknown	Unknown	Yes	Yes	No
LibriSpeech Clean	Default test set	5.4h	Monologue read-aloud speech	49	Unknown	No	No	No
LibriSpeech Other	Default test set	5.3h	Monologue read-aloud speech	33	Unknown	No	No	No
RT	Random selection	3.6h	Dialogue spontaneous speech	30	Yes	Unknown	Yes	Yes
ST	Random selection	4.7h	Monologue read-aloud speech	5	Unknown	No	No	No
Switchboard	Random selection		Dialogue spontaneous speech					
Tedlium	Default test set	2.6h	Monologue semi-spontaneous speech	11	Unknown	No	Yes	No
Timit	Default test set	1.4h	Monologue read-aloud speech	168	No	No	Yes	No
Voxforge	Default test set	3.9h	Monologue	171	Unknown	Yes	No	No

Table 2: Overview of corpus properties.

Microsoft, IBM and Amazon. Using the services of these technology heavyweights has a few advantages: The products have excellent documentation and provide client libraries in multiple languages which facilitates implementation. Furthermore these companies' services are tried and true and it will be interesting to see how their state of the art systems compare.

At the same time the usage of a cloud based service might not always be desirable or even possible due to privacy concerns and data protection laws. For these cases we want to evaluate some open source systems that can be setup and operated on private infrastructure without the need for the data leaving the premises. These systems also bring a lot of flexibility because they enable the training and adaptation of models for very specific use cases while the models of cloud services are more of a generic nature. This of course makes it a little difficult for our use case as we are unable to train our own models due to time constraints. We are therefore dependent on open source systems that provide pretrained models. We decided to incorporate CMUSphinx sphinx4, Kaldi and Mozilla Deepseech into our evaluation. These systems are popular open source systems with established communities. They also fall nicely into the different categories of ASR systems discussed earlier in subsection 5.1. CMUSphinx sphinx4 is a traditional statistical ASR system not using any . Kaldi on the other hand — actually more of a toolkit than a finished system — provides usage of a variety of algorithms including . And Mozilla DeepSpeech is an end-to-end system that implements the Baidu's DeepSpeech architecture. Last but not least all three systems provide models that are ready to use for decoding.

6.3.2 Systems Integration

As all of the systems are provided by different companies, the integration of each system is obviously different. We will first describe implementation specifics all systems have in common. Afterwards we will go into a little more detail for each system.

Audio Preprocessing

During examination of all the systems documentation two audio formats were predominantly requested or recommended: Lossless audio encodings — RIFF WAVE (WAV) or Free Lossless Audio Codec (FLAC) — either sampled at 16 kHz with a bit depth of 16 bit or sampled at 8 kHz with a bit depth of 8 bit. As all but one of our selected speech corpora provide audio files with 16 kHz / 8 bit or higher we decided to convert all audio into this format. We made 2 exceptions to this:

- Kaldi's ASpIRE Model requires audio files with a 8 kHz sample rate and bit depth of 16 bit. Therefore this was used for this configuration.
- The switchboard corpus is based on telephone recordings which are provided with 8 kHz and 8 bit. To be consistent we re-sampled the audio to 16 kHz / 8 bit just like before. However, it turned out to be working well for some systems and to be a very bad choice for others as they severely under-performed. We have therefore decided to test a small sample of the switchboard corpus on both 16 kHz / 16 bit and 8 kHz / 8 bit respectively and chose the best performing option for each system configuration.

The audio preprocessing was done with the sox command line utility.

Result Processing

The amount and structure of information returned from the different systems varies substantially. Some return just a simple textual representation of the audio. Others return additional alternative transcriptions and time stamps for every token/word. Some return a single result for the whole speech signal, others return the transcription in multiple parts. Each with its own alternatives, token lists etc.

To have a single result for every audio file we processed the returned results as follows:

- For each result we only considered the best alternative — meaning the one with the highest confidence value.
- If a transcription was returned in multiple parts, we merged them to a single transcription. In case multiple alternatives were returned we chose the ones with highest confidence.

Transcript Postprocessing

Just like the result structures the returned transcription texts themselves were also very different. Some only contained the lexical representation of the audio, others included punctuation and capitalization, others again even normalized specific detected entities like phone numbers, dates, digits etc. Certain systems even returned markers for non-speech such as silence, laughter or background noise.

Again, to have comparable transcriptions, we applied some used simple lexical transcriptions. For systems that did not provide this we applied post-processing to the transcripts to receive the required representation. The postprocessing applied encompassed:

- Removal of punctuation. Punctuation marks considered were periods, commas, question marks, exclamation marks and semicolons. We did not remove dashes and hyphens.
- Removal of non-speech markup (silence, noise & laughter).
- Reversing normalization of numerical values (using the num2words python package):
  - Decimals: `1.78` → `one point seven eight`
  - Time: `11:45` → `eleven forty-five`
  - Ordinals: `32nd` → `thirty-second`
  - Decades: `1980s` → `nineteen eighties`
  - Years: `1984` → `nineteen eighty-four`
  - And finally all remaining numbers: `999` → `nine hundred and ninety-nine`

Note that we did not normalize capitalization because the scoring tool was configured to ignore it. For more details on which systems needed this kind of post-processing see the more detailed system descriptions below.

Error Handling

With the amount of audio files to transcribe it's unavoidable that some transcriptions fail. This happened due to a few different Reasons: Some systems couldn't handle utterances with only noise, others just failed inexplicably. As we needed transcriptions from every system for every utterance we decided to assign empty transcriptions in such cases because that's what ultimately happened — the system did not recognize anything.

Transcription Language

The scope of our evaluation only encompasses speech in English. However, the proprietary systems offer specific dialects of English as transcription language. Because this option is not available for the open source systems we have decided to use American English as transcription language for most of the evaluation. However, within the scope of scenario we evaluate whether using the language/dialect of the speaker for transcribing improves accuracy. Therefor all corpora providing dialect information were also transcribed this way.

After these more general points that affected all systems we will provide additional system-specific implementation details in the following sections.

Google Cloud

The adapter for Google Cloud STT was implemented using the official `google-cloud-speech`, `google-cloud-storage` and google-cloud-core Pyhton packages. Because of the longer audio duration the longrunning recognize endpoint was used.

Google allows to choose from 3 different models when transcribing:

- `command_and_search`: A special model for short/single word utterances for voice commands/search commands or voice search.
- `phone_call` Best for audio that originated from a phone call
- `video`: A premium mode best suited for videos and recordings with more than one person speaking
- `default` Best for audio that does not fit any of the other models

Out of these models we used `phone_call`, `video` and `default` for our evaluation. See Table 3 for details on the configuration.

Amazon Transcribe

The adapter for Amazon Transcribe was implemented using the official *boto* Python package. The same package was use for upload all audio files to Amazon S3 Storage (this war required). See 3 for details on system configuration.

Microsoft Azure

The adapater for Microsoft Azure was implemented using the official azure-cognitiveservices-speech Python package based on the sample from the github repository [35]. Due to the duration of the utterances (>15 s) the continuous recognition interface was used. If the service only recognizes noise or silence inside a specific timeout it does not return a result. It also returns an error for 0-length utterances. See Table 3 for all configurations used.

IBM Watson

The adapter for IBM Watson was implemented using the official python sdk based on the synchronous example from the ocumentation[25]. There are multiple models to choose from but that's mainly to choose the transcription language and samplerate. We use the `en-US_Narrowband` model for Switchboard and the `en-US_Boradband` model for every other corpus. For the dialect Experiment in 13 dialect specific *en-**_ Broadband* models were used where available.

Kaldi

Unlike the other ASR systems incorporated in the scope of this evaluation Kaldi is a speech recognition toolkit rather than a finished system. It is very powerful and flexible but starting from scratch can be challenging. Luckily a myriad of recipes are provided with the project repository. Out of these recipes we chose the following two for which pretrained models are available:

- The **ASpIRE** recipe was developed as a submission to the ASpIRE challenge and is trained on the Fisher-English corpus, which is a corpus of telephone recordings.
- The **LibriSpeech** recipe trains a model on the LibriSpeech corpus which we also use in this evaluation.

For decoding we used the built in decoder binaries `online2-wav-nnet3-latgen-faster` and `online2-wav-nnet2-latgen-faster` for the ASpIRE and the LibriSpeech models respectively. See Table 3 for all configurations used. The Kaldi version used in this evaluation is 5.5, which was the most current at the time of writing.

Mozilla DeepSpeech

Mozilla DeepSpeech is a relatively new system that implements the Baidu DeepSpeech architecture presented in [23] and [2]. It ships with pre-trained acoustic and language model.

The version used in this evaluation is 0.4.1, which is the latest at the time of writing. There are v0.5.0-alpha pre-releases but these don't come with ready-to-use models. The model is reportedly trained on American English data from the Fisher, LibriSpeech and Switchboard training corpora as well as a pre-release snapshot of the English Common Voice training corpus[10]. It is provided with or without rounded weights which increase computation. We included both versions into our evaluation.

For decoding we used the official python bindings provided with the `deepspeech` python package. See Table 3 for all the configurations used.

CMUSphinx sphinx4

Sphinx 4 is the oldest system used in our Evaluation with the first version of it being released over 20 years ago. It is also the only system that does not use DNNs but rather a tradition HMM-based paradigm. And according to its timeline and posts in the forum there are no plans to change that.

As sphinx4 is implemented completely in java we had to create a simple java cli for decoding which we based on an example from the code repository[57]. This cli received one or multiple audio files and returned the transcriptions to the console where it was parsed from python program. There are multiple models available for download on the website[9]. The most current one for English comes in three version. A semi-continuous, a PTM and a continuous model respectively. The difference between these models is the number of gaussians that are used to calculate the score. We chose to evaluate on the PTM model which is the golden middle and the recommended version. Additionally we choose the continuous model which uses the highest number of gaussians and is therefore slower. See Table 3 for more details on the used configurations.

6.4 Metrics and Evaluation Tool

As discussed in chapter subsection 5.4 the most common metric for ASR evaluation is Word Error Rate. Although shortcomings of this metric are widely discussed in the literature, there have not been found any alternative metric that would essentially outperform WER. For this reason, WER is used in the following evaluation. However, we take an attempt to investigate how accurately WER reflects the transcription accuracy and hence a more fine-grained evaluation of the transcriptions is performed. Metrics applied in this WER qualitative analysis are:

For alignment of the reference text with the hypothesis text and for calculating the WER based on this alignment, we use sclite, one of most commonly applied evaluation tools (see chapter subsection 5.4 for details). The reference and hypothesis files are provided to sclite in the trn format. As a result of the alignment, a report in the sgml format is generated. It contains confusion pairs together with their classification in the categories: substitution, deletion and insertion. The correctly recognized words are also marked accordingly. Finally, the report is parsed in python to be used for further processing in the evaluation.

6.5 Scenarios

The foundation for evaluation scenarios were four previously mentioned speech recognition task categories and ASR use cases related to these tasks. The categories were used as a basis for determining utterance properties potentially affecting ASR systems performance. Table 4 provides an overview of the use cases combined with most utterance properties values perceived as most probable for particular use case. The properties values, which seem to be most challenging per use case, are marked in bold. The properties comprise spoken language properties on one hand and properties related to recording setup on the other.

Based on the utterance properties, a set of evaluation scenarios was defined. The attention was focused on the properties related to spoken language variabilities. The scenarios are presented in the Table 12.

The objective of the five scenarios is to measure the performance of each system for each of the pre-defined property values.

6.6 WER Analysis

The objective of the WER-Analysis is to investigate the WER in more detail. As discussed before — it can be ambiguous as it does not differentiate between different kinds of errors. A noun in the plural form transcribed to the singular form weighs just as much as if it was a completely different word. Open compounds (written as 2 words) transcribed to hyphenated or closed compounds (written together) even result in 2 errors: An insertion and a substitution.

System	Model	Audio	Language	Merged Transcription	Applied Transcription Postprocessing
Google Cloud	video	FLAC / 16 kHz / 16 bit	en-US	✓	Removal of punctuation, reversion of numeric normalization back to lexical representation.
Google Cloud	default	FLAC / 16 kHz / 16 bit	en-**	✓	Removal of punctuation, reversion of numeric normalization back to lexical representation.
Google Cloud	phone_call	FLAC / 16 kHz / 16 bit	en-US	✓	Removal of punctuation, reversion of numeric normalization back to lexical representation.
Amazon Transcribe	Default	WAV / 16 kHz / 16 bit	en-**	✓	-
Amazon Transcribe	Default	WAV / 8 kHz / 8 bit	en-**	✓	-
MS Azure	Default	WAV / 16 kHz / 16 bit	en-**	✓	Reverted numeric normalization back to lexical representation
MS Azure	Default	WAV / 8 kHz / 8 bit	en-**	✓	Reverted numeric normalization back to lexical representation
IBM Watson	en-**_BroadbandModel	FLAC / 16 kHz / 16 bit	en-**	✓	Removal of `%HESITATION`
IBM Watson	en-**_NarrowbandModel	FLAC / 98 kHz / 8 bit	en-**	✓	Removal of `%HESITATION`
Kaldi	ASpIRE	WAV / 8 kHz / 16 bit	en	✗	Removal of `[noise]`, `[laughter]`, `[vocalized-noise]` and `<unk>`
Kaldi	LibriSpeech	WAV / 16 kHz / 16 bit	en	✗	Removal of `[noise]`, `[laughter]`, `[vocalized-noise]` and `<unk>`
Mozilla Deepspeech	Included	WAV / 16 kHz / 16 bit	en-US	✗	-
Mozilla Deepspeech	Included, Rounded	WAV / 16 kHz / 16 bit	en-US	✗	-
CMUSphinx sphinx4	Included, PTM	WAV / 16 kHz / 16 bit	en-US	✗	-
CMUSphinx sphinx4	Included, Continuous	WAV / 16 kHz / 16 bit	en-US	✗	-

Table 3: ASR system configurations

Recognition task category	**Human to human dialogue**	**Human to human monologue**	**Human to machine dialogue**	**Human to machine monologue**
	In-meeting virtual assistant, call and interview transcriptions	Lecture transcriptions	Voice-activated tools	Dictation tool
Utterance duration	**Short to medium**	Long	Long	Medium to long
Speaking rate	Medium	Fast	Slow	Slow
Dialect	Any	Any	Any	Any
Accent	Any	Any	Any	Any
Overlapping speech	High probability of occurrence	Low probability of occurrence	Low probability of occurrence	Low probability of occurrence
Filled pauses	High probability of occurrence	High probability of occurrence	Medium probability of occurrence	Medium probability of occurrence
Acoustic environment	Any indoor space, but outdoor also probable	Any indoor space, usually lecture hall or class room	Any, also outdoor	Any indoor space
Recording device	Lapel or distant microphone, headset, microphone array, phone-integrated or computer-integrated microphone, microphone array	Lapel or distant microphone, headset, microphone array	Phone-integrated, lapel or distant microphone, headset	Phone-integrated or computer-integrated microphone, dictation device

Table 4: Overview of the ASR use cases.

Utterance property	**Property values**					
Utterance duration (UD)	1sec ≤ UD < 2sec	2sec ≤ UD < 4sec	4sec ≤ UD < 6sec	6sec ≤ UD < 8sec	8sec ≤ UD < 10sec		
Speaking rate	SP < 100	100 ≤ SP < 160	160 ≤ SP < 200	200 ≤ SP < 300			
Dialect	en-US	en-GB	en-AU	en-CA	en-NZ	en-IN	en-Other
Accent	Native	Non-native					
Filled pauses	No filled pauses in utterance	Filled pauses removed from reference	Filled pauses in both reference and audio	Utterances containing merely filled pauses			

Table 5: Overview of evaluation scenarios.

We therefore want to explore what kind of errors constitute the WERs resulted from our evaluation. We will do this in two steps.

At first we will look at the actual character difference of the wrong words. How big the differences actually are and how they are distribution throughout specific WERs obtained. The difference between the words will be calculate the same way as the WER on reference and transcription, but on character level instead of word level. This is also known as Character Error Rate (CER). But usually it is used to determine the error rate on the whole transcription, just like with the WER. We want to use it to determine how big the errors already scored with the WER are.

$$\begin{array}{rcl} & \rightarrow & \text{four} \\ \text{fourleaf} & \rightarrow & \text{four} \\ & \Downarrow & \\ \text{fourleaf} & \rightarrow & \text{fourleaf} \end{array}$$

(a) Correction of a compound word error from `DELETION` and `SUBSTITUTION` mutations to a `CORRECT` mutation.

$$\begin{array}{rcl} \text{authorises} & \rightarrow & \text{authorizes} \\ & \Downarrow & \\ \text{authorises} & \rightarrow & \text{authorises} \end{array}$$

(b) Correction of a spelling variation error from a `SUBSTITUTION` mutation to a `CORRECT` mutation.

Figure 4: Examples for error corrections.

In a second step we want to take a more linguistic approach and investigate what type of errors the WER is made up of. Depending on the use case certain kinds of error are much worse than others. And it would be interesting to know how the proportions of these error types are. Specifically we want to look at the following:

- Compound words
- Homonyms
- Contractions
- Plurals
- Spelling variations (American vs British)
- Filled pauses

To calculate how much of the WER can be attributed to each individual error type, we use the alignment from the WER calculation. We subsequently — for one error type after another — check the mutations (Deletions, Substitutions & Insertions) for each utterance and identify the ones that can be attributed to the error type. For every detected instance of an error we adjust the alignment such that the error is removed. See Figure 4 for examples of a corrected compound word error (4a) and a corrected spelling variation error (4b). After all alignments have been processed for a specific error type we save the alignment and pass it on for the analysis of the next error type.

In the end we have the initial alignment and an additional alignments for each error type. Each one a little better than the one before. For all these alignments the WER can be calculated and by calculating the subsequent differences between the WERs we can determine the share an error type has in the word error.

6.7 Process

The evaluation process implemented for the purpose of this thesis comprises six steps: corpus pre-processing, transcription, transcripts' post-processing, alignment, scoring and reporting. Figure 5 gives an overview of the complete process flow.

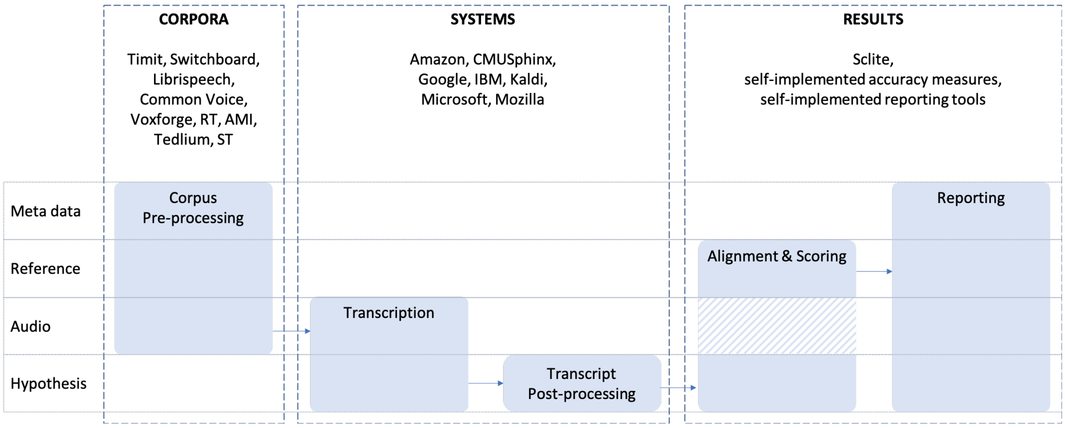

CORPORA
Timit, Switchboard, Librispeech, Common Voice, Voxforge, RT, AMI, Tedlium, ST
SYSTEMS
Amazon, CMUSphinx, Google, IBM, Kaldi, Microsoft, Mozilla
RESULTS
Sclite, self-implemented accuracy measures, self-implemented reporting tools
Meta data
Reference
Audio
Hypothesis
Corpus Pre-processing
Transcription
Transcript Post-processing
Alignment & Scoring
Reporting

Figure 5: Evaluation process flow

The corpus pre-processing step consists in extracting required information from the provided corpus and transforming it into uniform data structure needed for further processing. The basic data structure is an utterance which is gradually enriched with information in every of the six process steps listed above.

The process starts with corpus pre-processing. After retrieving the corpus, a quality check of the corpus is performed. Its aim is to identify incomplete utterance data. If the data missing is the reference text or the audio file or if the start and end time of the utterance are incorrect the utterance is removed from the data set. Afterwards, the available data is normalized.

A large part of the pre-processing is performed on the reference text. In some cases, the reference file needs to be segmented as the reference is provided as one large file containing multiple utterances. On the other hand, there are long utterances, that require splitting so that their duration does not exceed several seconds. Furthermore, if the text contains any tags, such as ones describing speaker noise (e.g. laughing or coughing), non-speaker noise or speaker disfluencies (restarting, partial words etc.), these tags are removed.

In order to perform possibly reliable comparison of performances between the systems, the reference text and the transcript (hypothesis text) must have possibly similar format. Reference and hypothesis normalization comprises removing punctuation marks, transforming all letters to lower-case, spelling out numbers and applying the same formatting of integers, decimal values, time, year and decades. We tried to discover and eliminate as much discrepancies of this kind as possible, however a complete consistency between the texts in terms of formatting cannot be guaranteed.

After the pre-processing step the utterance data structure comprises clean and normalized utterance data: reference text, path to the audio file, meta data such as language, dialect, accent, information if the utterance contains overlapping speech or speaker noise and more. For full specification of the utterance data structure please refer to the Appendix B Utterance Data Structure.

In the next step, the utterances get prepared for the transcription. This step consists in extracting audio information and trimming the audio file, if necessary. Similarly to reference file length, also the duration of provided audio files varies essentially between the corpora. Some provide a set of audios of several seconds, each containing one short utterance, while others comprise long audios with long utterances of several minutes. In some cases, there are long audios provided containing multiple short utterances. These audios are segmented accordingly to ensure consistent segmentation throughout all corpora. In the final setup, each corpus has a variant, where both reference text and audio are segmented. As a result s, the duration of all utterances lies below 10 seconds. Appendix A Corpora Documentation and D Corpora Variants provide an overview of all corpora variants with detailed information about reference and audio segmentation.

The next step consists in transcribing the utterances by all systems. The audios are first converted into the audio encoding and audio signal required by the particular system. The utterance data structure is extended with new information such as the hypothesis text in the original and normalized version, the language model that was used for the transcription as well as audio file details such as sample rate, bit depth and number channels. The transcriptions are performed with various parameter settings for each system. For the details of the system parameter settings, please refer to subsection 6.3

As soon as the reference and hypothesis texts are available, the alignment can be performed. Sclite takes reference and hypothesis as inputs and calculates the number of insertions, deletions and substitutions. The alignment results, which is a full list of confusion pairs and words correctly transcribed together with the final WER score, are appended to the utterance data structure.

This way the complete utterance data set is in place, ready to be applied as foundation for the systems comparison and further analysis.

7 Evaluation Results

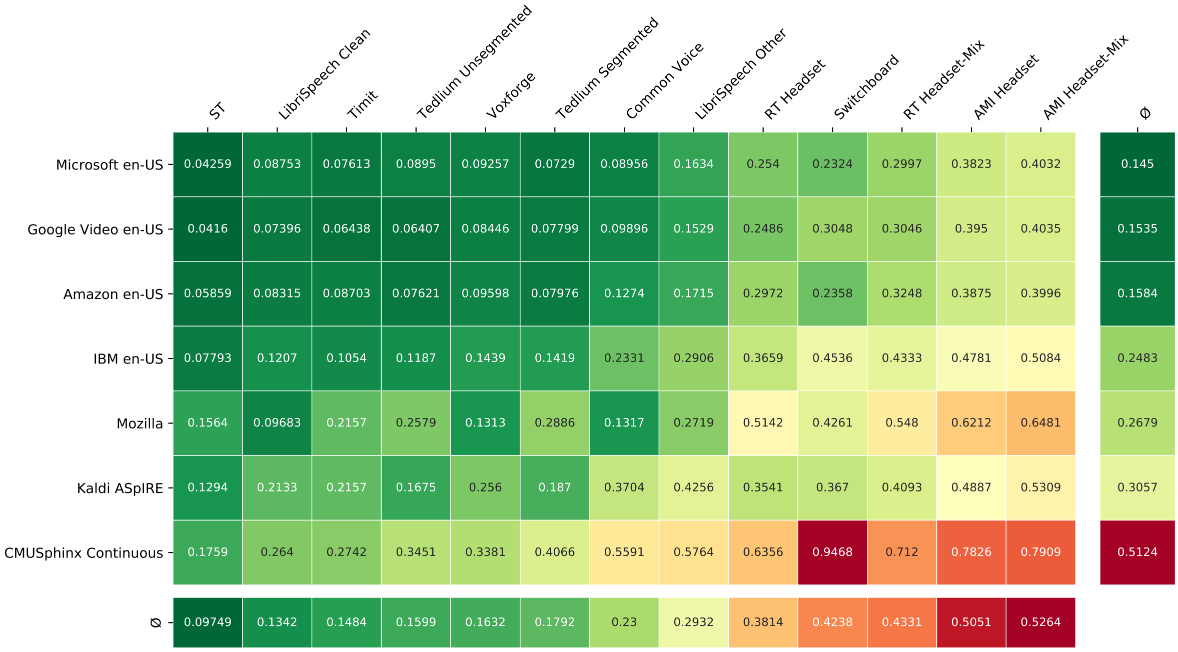
The evaluation was performed with three different objectives and hence involved three different evaluation approaches.

1. In order to compare the performance of the seven systems, we ran transcription of nine corpora on each of the systems. We tested various system configurations.
2. Following the goal of analyzing correlations between spoken language variabilities and transcription accuracy, we conducted several experiments on carefully selected utterance samples with various spoken-language-related properties.
3. Finally, in order to perform a fine-grained analysis of WER, we investigated in detail the confusion pairs provided as result of alignment between references and hypothesis.

The results of each approach are presented in the following chapters.

7.1 Overall Performance Comparison

Figure 6 provides an overview of WER values for all System - Speech Corpus pairs which were subject to this evaluation. For each system the best performing configuration has been selected.



	ST	LibriSpeech Clean	Timit	Tedlium Unsegmented	Voxforge	Tedlium Segmented	Common Voice	LibriSpeech Other	RT Headset	Switchboard	RT Headset-Mix	AMI Headset	AMI Headset-Mix	Ø
Microsoft en-US	0.04259	0.08753	0.07613	0.0895	0.09257	0.0729	0.08956	0.1634	0.254	0.2324	0.2997	0.3823	0.4032	0.145
Google Video en-US	0.0416	0.07396	0.06438	0.06407	0.08446	0.07799	0.09896	0.1529	0.2486	0.3048	0.3046	0.395	0.4035	0.1535
Amazon en-US	0.05859	0.08315	0.08703	0.07621	0.09598	0.07976	0.1274	0.1715	0.2972	0.2358	0.3248	0.3875	0.3996	0.1584
IBM en-US	0.07793	0.1207	0.1054	0.1187	0.1439	0.1419	0.2331	0.2906	0.3659	0.4536	0.4333	0.4781	0.5084	0.2483
Mozilla	0.1564	0.09683	0.2157	0.2579	0.1313	0.2886	0.1317	0.2719	0.5142	0.4261	0.548	0.6212	0.6481	0.2679
Kaldi ASpIRE	0.1294	0.2133	0.2157	0.1675	0.256	0.187	0.3704	0.4256	0.3541	0.367	0.4093	0.4887	0.5309	0.3057
CMUSphinx Continuous	0.1759	0.264	0.2742	0.3451	0.3381	0.4066	0.5591	0.5764	0.6356	0.9468	0.712	0.7826	0.7909	0.5124
Ø	0.09749	0.1342	0.1484	0.1599	0.1632	0.1792	0.23	0.2932	0.3814	0.4238	0.4331	0.5051	0.5264	

Figure 6: General overview of WER results for best performing system configurations

It can be observed in Figure 6 that the top two systems (Google and Microsoft) are performing very consistently. Against expectations, the IBM solution, is only in the fifth position, while Amazon Transcribe, a system which is not that frequently mentioned in the literature is performing almost as well as the two leaders.

It can be also derived from the overview that the spontaneous dialogue speech recognition is substantially more challenging for the systems than recognizing non-spontaneous or semi-spontaneous speech. In the following, we will take a closer look at the performance of the systems in these two categories: non-spontaneous and semi-spontaneous monologue speech recognition versus spontaneous dialogue speech recognition.

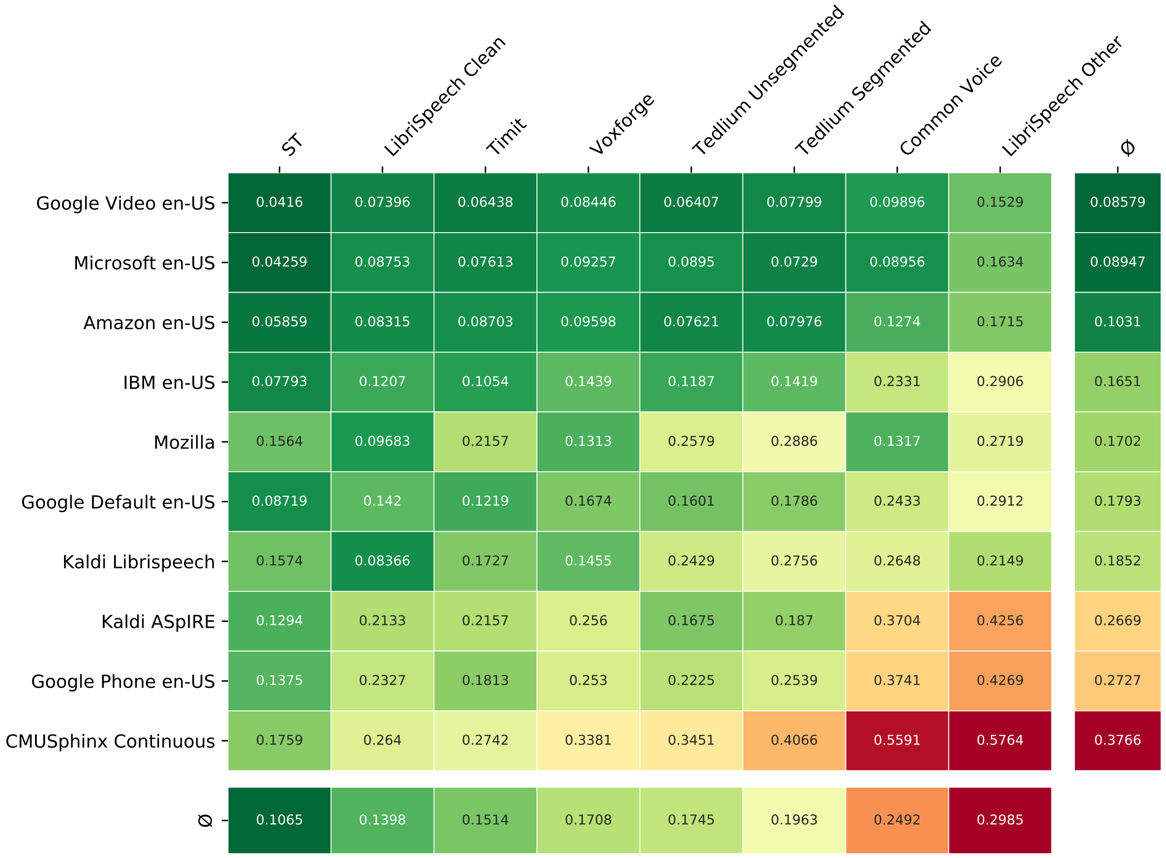


Figure 7: Overview of WER results for non-spontaneous and semi-spontaneous speech corpora

7.2 Non-spontaneous and Semi-spontaneous Monologue Speech Recognition

Figure 7 provides an overview of the results for all systems in all tested configurations for five non-spontaneous (read-aloud) speech corpora Common Voice, Librispeech, VoxForge, Timit, ST and one semi-spontaneous speech corpus TedLium (a set of lectures recordings).

7.3 Spontaneous Dialogue Speech Recognition

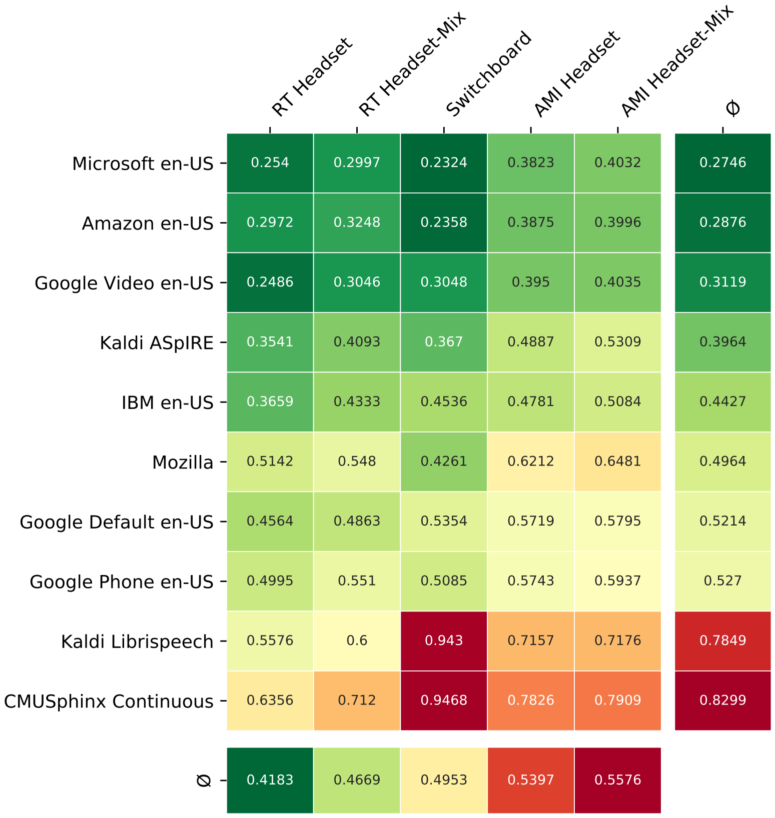
The transcription of three spontaneous dialogue speech corpora resulted in the scoring provided in the figure Figure 8.

The average WER for all systems on all spontaneous dialogue corpora is essentially higher than the scoring for the first category (see subsection 7.2). This is due to the fact that spontaneous speech is significantly different from read speech both acoustically and linguistically, in a way which makes it more difficult to recognize. [47]

Dialogue speech is challenging due to non-canonical pronunciations, acoustic and prosodic variability, and high levels of disfluency, resulting from occurrence of filler words, repetitions, false starts, repaired utterances and stuttering). [21] [24] Spontaneous speech is additionally characterized by accelerated speaking rate and higher out-of-vocabulary rate. [47]

Another problem that arises in dialogue speech, especially meeting speech, refers to multiple concurrent speakers. It was observed that around 50% of speech segments in a meeting or telephone conversation contain some degree of overlapping speech. These overlapped speech segments are problematic for speech recognition, producing an increase in WER. [39]

In this evaluation, the overlapping utterances have been excluded due to missing reference



	RT Headset	RT Headset-Mix	Switchboard	AMI Headset	AMI Headset-Mix	Ø
Microsoft en-US	0.254	0.2997	0.2324	0.3823	0.4032	0.2746
Amazon en-US	0.2972	0.3248	0.2358	0.3875	0.3996	0.2876
Google Video en-US	0.2486	0.3046	0.3048	0.395	0.4035	0.3119
Kaldi ASpIRE	0.3541	0.4093	0.367	0.4887	0.5309	0.3964
IBM en-US	0.3659	0.4333	0.4536	0.4781	0.5084	0.4427
Mozilla	0.5142	0.548	0.4261	0.6212	0.6481	0.4964
Google Default en-US	0.4564	0.4863	0.5354	0.5719	0.5795	0.5214
Google Phone en-US	0.4995	0.551	0.5085	0.5743	0.5937	0.527
Kaldi Librispeech	0.5576	0.6	0.943	0.7157	0.7176	0.7849
CMUSphinx Continuous	0.6356	0.712	0.9468	0.7826	0.7909	0.8299
Ø	0.4183	0.4669	0.4953	0.5397	0.5576	

Figure 8: Overview of WER results for spontaneous speech corpora

transcriptions reflecting overlapped utterances. The corpora are provided with transcriptions of single speaker utterances, and these transcriptions do not reflect what speech can be actually heard in the audio recording. Such transcriptions could be created by merging single speaker utterance transcriptions based on the word time stamps, however, due to time constraint of this project, this solution has not been implemented.

7.4 Scenario-Based Evaluation

The sample selection and category definition for the experiments described below followed several principles. The essential factor considered during the sample selection process was the available sample size. If the total duration of a category sample was equal or below 1 minute the category was excluded from an experiment. It was agreed that utterances from each category must come from the same set of corpora or must have the same properties. By properties, we mean utterance duration, speaking rate, dialect, accent, occurrence of overlapping speech and filled pauses. These properties were essential selection criteria. Most common speaking rate throughout all corpora is in range between 100 and 200 words per minute. For this reason, this group was selected for most evaluations. In some cases, if filtering by speaking rate resulted in a too small sample, this filter was left out. Utterances of duration between 2 and 6 seconds build the largest data set, and hence this group was selected as a basis for most experiments. The categories used for samples partitioning were defined based on statistics generated for the corpora. Various data partitions were considered and the most representative ones were selected. From among various systems configurations with various model, the configuration with the best WER score was selected and this score is provided in the results table.

7.4.1 Utterance duration

The objective of the following experiment is to evaluate the performance of each system depending on the utterance duration (UD).

For speaker recognition systems, the impact of utterance duration on the system performance is widely investigated. Multiple studies provide evidence that speaker recognition accuracy degrade with decreasing utterance length, such as [52] or [53]. However, following the results of the literature research conducted for the purposes of this thesis, the correlation between utterance duration and speech recognition performance appears to be less investigated. No studies have been found that could build the theoretical foundation for the following experiment. However, an impact of utterance duration variability on the speech recognition accuracy seems probable. In the following experiment, we investigate if any correlation can be detected.

UD Experiment A: Short Utterances below 10 Seconds

This experiment is based on three read-aloud corpora: Common Voice, Timit and VoxForge. These corpora comprise merely short utterances of duration of several seconds. Utterances below one second and above ten seconds were excluded from the evaluation due to insufficient amount of data (only two utterances in total in all three corpora).

Sample Selection

Source corpora:	Common Voice, Timit, VoxForge
Sample size:	207.7 min (3099 utterances)
Utterance duration:	see section “Results”
Speaking rate:	between 100 and 200 words per minute
Dialect:	en-US
Accent:	native
Overlapping speech:	no
Non-lexical filler words:	no

Results

Sample duration	**4 min**	**84 min**	**82 min**	**31 min**	**6 min**
Category (UD range)	**[1 sec, 2 sec)**	**[2 sec, 4 sec)**	**[4 sec, 6 sec)**	**[6 sec, 8 sec)**	**[8 sec, 10 sec)**
Amazon en-US	0.1	0.09	0.09	0.09	**0.09**
Google Video en-US	**0.09**	**0.07**	**0.07**	**0.08**	**0.09**
IBM en-US	0.14	0.12	0.12	0.14	0.14
Kaldi Librispeech	0.18	0.17	0.13	0.12	0.13
Mozilla	0.19	0.18	0.11	0.1	**0.09**
Microsoft en-US	0.1	0.08	0.08	0.09	0.1
CMUSphinx Continuous	0.31	0.3	0.33	0.33	0.3
Avg WER	0.172	0.159	**0.148**	0.153	0.149

Table 6: WER results for various utterance durations (below 10 seconds)

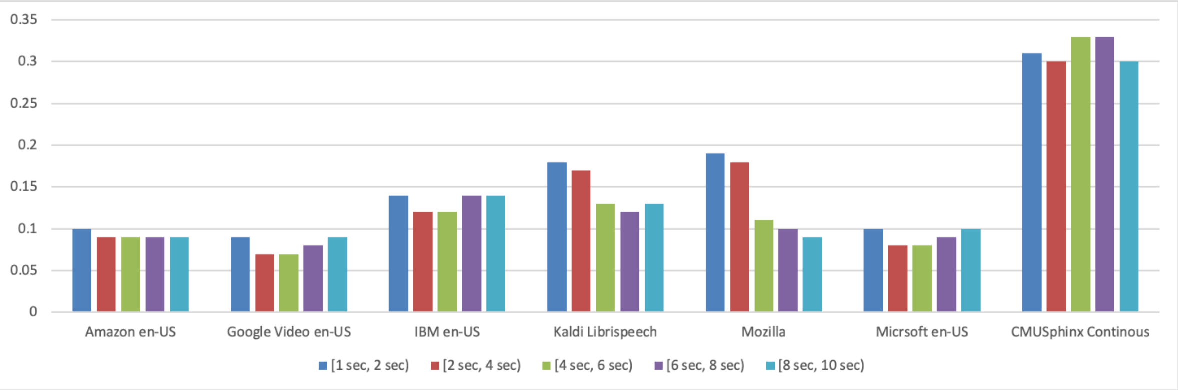


Figure 9: WER results for various utterance durations (below 10 seconds)

Conclusions

In case of all systems with the exception of CMUSphinx, the WER value is the highest in the shortest utterances transcriptions. However, for longer utterances no clear correlation can be detected. The lowest average WER is achieved when the utterances have the length of four to six seconds, which is the medium length in the investigated data set. As the duration difference between the categories is relatively small, another experiment was performed in order to investigate the correlation between utterance duration and transcription accuracy (see Experiment 2).

The best performing system in all categories is Google. In case of the longest utterances, the same result as Google's was achieved by Amazon and Mozilla.

UD Experiment B: Several Seconds versus Several Minutes Utterances

In this experiment, the ASR performance is measured in presence of substantial utterance duration variability. The transcription accuracy for utterances of several seconds is compared with the accuracy for utterances of several minutes. The basis of the experiment is one semi-spontaneous corpus: TedLium. The idea of the experiment is to compare two segmentation variants of lectures from the TedLium corpus: segmented version with utterance of less than ten seconds and unsegmented version comprising utterances of more than five minutes.

Sample Selection

Source corpora:	TedLium
Sample size:	322.9 minutes (1166 utterances)
Utterance duration:	see section “Results”
Speaking rate:	between 100 and 200 words per minute
Dialect:	en-US
Accent:	native
Overlapping speech:	no
Non-lexical filler words:	no

Results

Sample duration	**157 min**	**165.9 min**
Category	**UD $<$ 10 seconds**	**UD $>$ 5 minutes**
Amazon en-US	0.08	0.08
Google Video en-US	0.08	**0.06**
IBM en-US	0.14	0.12
Kaldi Aspire	0.19	0.17
Mozilla	0.29	0.26
Microsoft en-US	**0.07**	0.09
CMUSphinx Continuous	0.41	0.35
Avg WER	0.199	**0.179**

Table 7: WER results for various utterance durations (several seconds versus several minutes)

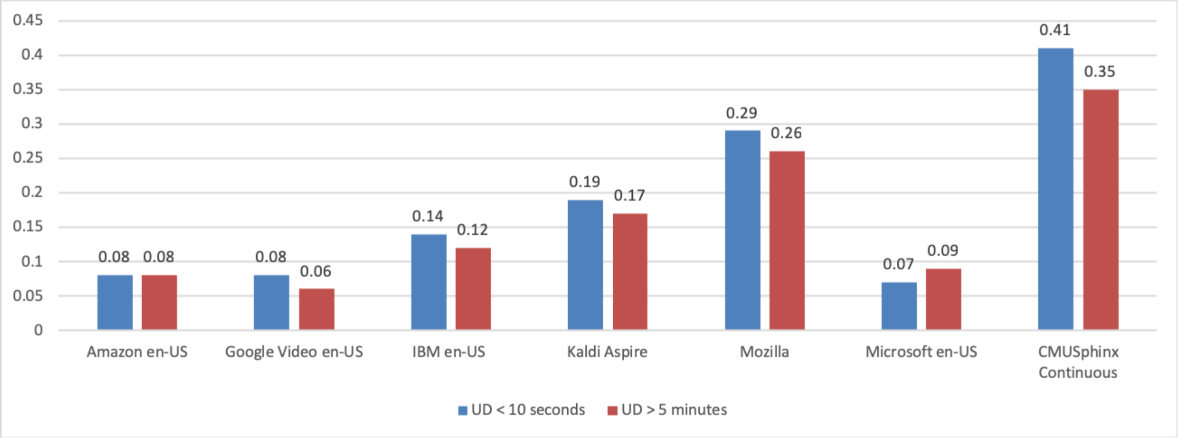


Figure 10: WER results for various utterance durations (several seconds versus several minutes)

Conclusions

In case of this experiment, the correlation between the utterance length and the transcription quality can be observed. With the exception of Microsoft system, the accuracy is always higher for longer utterances, which leads to a higher average WER value in this category.

The best performing system in the short utterances category is Microsoft. In case of long utterances the highest accuracy is achieved by Google.

7.4.2 Speaking rate

The scope of the following experiment is to investigate the performance of the systems in the context of the speaking rate (SR). We seek to discover if any correlation between the speaking rate and the WER value can be observed.

The speaking rate (also called speech rate) is the speed of speech, that can be measured as word rate (number words per minute), phone rate (number phones per second) [56] or vowel rate (number vowels per second)[43]. Speaking rate has been shown to have a significant effect on speech recognition. [56] Fast speech is associated with less accuracy. But very slow speech has also been found to correlate with higher error rates. [21] Figure 11 shows recognition error rates for subsets of 100 utterances from the Wallstreet Journal corpus (WSJ1) grouped by word rate. The bell-shaped curve shows the distribution of utterances in the entire corpus. [56]

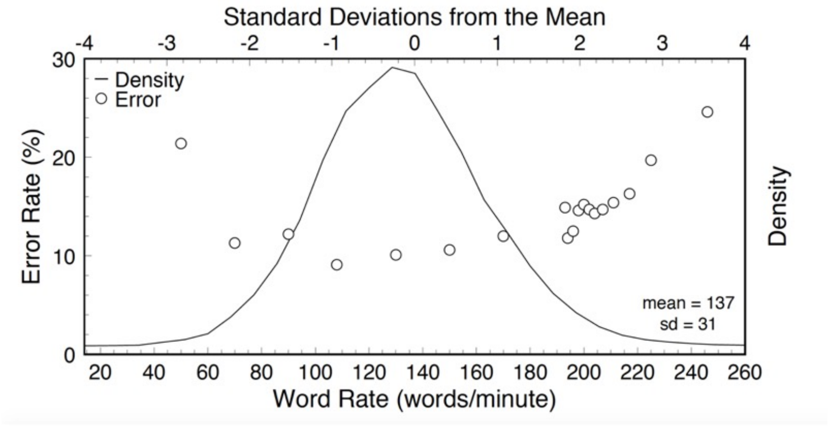


Figure 11: Recognition error rates for subsets of 100 utterances from the WSJ1 corpus grouped by word rate [56]

Most utterances in the corpus presented in Figure 11 have a speaking rate between 100 and 160 words per minute. This is also the range, where the Error Rate has the lowest value. These observations were considered while determining the approach for the speaking rate evaluation below.

The basis of the experiment are three read-aloud corpora: Common Voice, Timit and VoxForge. The categories used for samples partitioning were defined based on the literature research and statistics generated for the corpora under investigation. Metric applied in the experiment is word rate, hence the speaking rate is calculated as follows:

$$SR = \frac{\text{number of words in an utterance}}{\text{utterance duration in minutes}} \tag{6}$$

The data is partitioned into four categories. Each category is a speaking rate range. Utterances with a speaking rate above 300 words per minute were excluded from the evaluation due to insufficient amount of data (only 1 utterance in total in all three corpora).

Sample Selection

Source corpora:	Common Voice, Timit, VoxForge
Sample size:	239.85 minutes (3867 utterances)
Utterance duration:	between 2 and 6 seconds
Speaking rate:	see section “Results”
Dialect:	en-US
Accent:	native
Overlapping speech:	no
Non-lexical filler words:	no

Results

Sample duration	**48 min**	**107 min**	**59 min**	**26 min**
Category (SP range)	**[40, 100)**	**[100, 160)**	**[160, 200)**	**[200, 300)**
Amazon en-US	0.06	0.09	0.08	0.07
Google Video en-US	**0.04**	**0.07**	**0.07**	**0.06**
IBM en-US	0.09	0.13	0.11	0.08
Kaldi Librispeech	0.15	0.16	0.15	0.15
Mozilla	0.1	0.16	0.15	0.15
Microsoft en-US	0.05	0.08	**0.07**	**0.06**
CMUSphinx Continuous	0.22	0.32	0.3	0.28
Avg WER	**0.11**	0.159	0.146	0.136

Table 8: WER results for various speaking rates

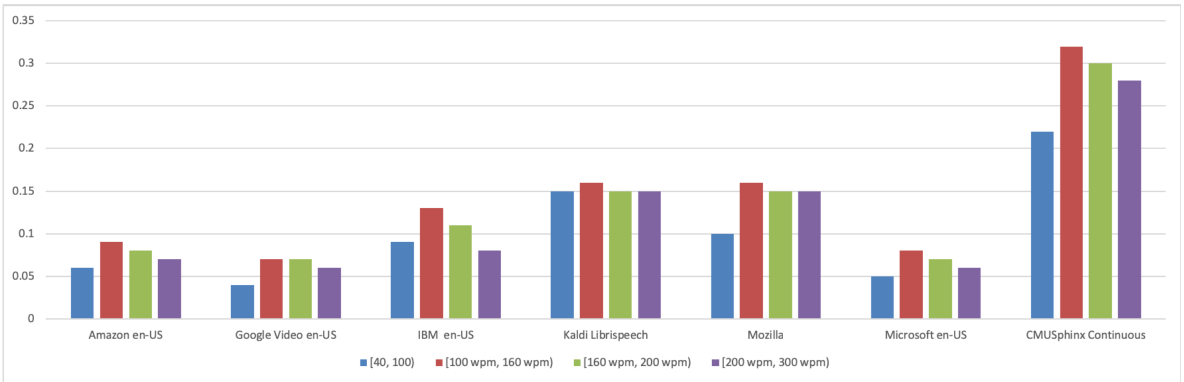


Figure 12: WER results for various speaking rates

Conclusions

The best result was achieved on utterances with in the slowest speaking rate ranged between 40 and 100 wpm. The second best result can be observed in the fast speech category (between 200 and 300 wpm). When compared to the results described in [56], these are exactly the opposite phenomena. The difference may result from the disproportion between the data set sizes of the particular categories. Another probable reason may be the technology development. The papers mentioned in the introduction to the experiment are dated back to 1995 and 1998. It has not been further investigated.

Google achieved the best result in all categories. On utterances with the fastest speaking rates, however, Microsoft's performance was the same as Google's.

7.4.3 Dialect

In the following experiment we investigate if the speaker's dialect may affect the transcription accuracy.

In this thesis, the term "dialect" refers to a regional variety of a language. As we focus merely on English language, the dialect list comprises of varieties of English spoken as a first language in different countries, such as United States of America, Great Britain, Canada, Australia and more (see C Dialects List of English Language for a complete list of English language dialects). Although dialects of the same language share many similarities, they are often differentiated at linguistic levels such as phonological and grammatical, and very often have different vocabularies. [12] Hence, modelling dialectal variation is a challenge for ASR technology. Research has shown that ASR performance typically decreases when evaluated on a dialect of the same language that was not used for training its models. Similarly, models simultaneously trained on a group of dialects tend to underperform when compared to dialect-specific models. The Google AI team conducted a comparison of the two approaches: dialect-specific system versus system combining several dialects (in this case these were Arabic dialects). The conclusion was that the combined system performs consistently worse than the dialect-specific systems across all dialects under investigation, although the combined system was trained on about 5 times the training data. [5] Therefore, in many state-of-the-art speech recognition systems there are different recognizers per dialect. That is, each recognizer is trained both acoustically and linguistically on dialect-specific data. When trying to decide which dialect-specific model to use to decode an utterance, possible strategies include automatically detecting the spoken dialect or following predefined language settings. [12]

The basis of the experiment are two read-aloud corpora: Common Voice and VoxForge. Out of all dialects occurring in the corpora under investigation, the following were selected: en-US (United States), en-CA (Canada), en-GB (Great Britain), en-IN (India) and en-NZ (New Zealand). The experiment is divided into 2 parts:

1. Evaluation of all systems with their default models on utterances spoken in dialects listed above (experiment A).
2. Evaluation of systems with dialect-specific models. Dialect-specific models different than en-US are available in Google, Microsoft, IBM and Amazon (experiment B).

Both experiments are conducted on the same data set.

Sample Selection

Source corpora:	Common Voice, VoxForge
Sample size:	133.7 minutes (1944 utterances)
Utterance duration:	between 2 and 6 seconds
Speaking rate:	between 100 and 200 words per minute
Dialect:	see section “Results”
Accent:	native
Overlapping speech:	no
Non-lexical filler words:	no

Results

In the following, the results of both experiments are provided and some conclusions from the experiments are drawn.

Dialect Experiment A: Transcription with en-US model (default model)

Sample duration	**105 min**	**10 min**	**8 min**	**4 min**	**4 min**
Category (Dialect)	**en-US**	**en-GB**	**en-CA**	**en-NZ**	**en-IN**
Amazon en-US	0.09	0.1	0.08	**0.08**	0.24
Google Default	0.16	0.20	0.15	0.27	0.43
IBM en-US	0.13	0.15	0.11	0.21	0.47
Kaldi Librispeech	0.14	0.12	0.12	0.14	0.52
Mozilla	0.1	0.13	0.1	0.12	0.26
Microsoft en-US	**0.08**	**0.08**	**0.07**	**0.08**	**0.14**
CMUSphinx Continous	0.34	0.37	0.38	0.42	0.84
Avg WER	0.14	0.15	0.13	0.16	0.38

Table 9: WER results for various dialects

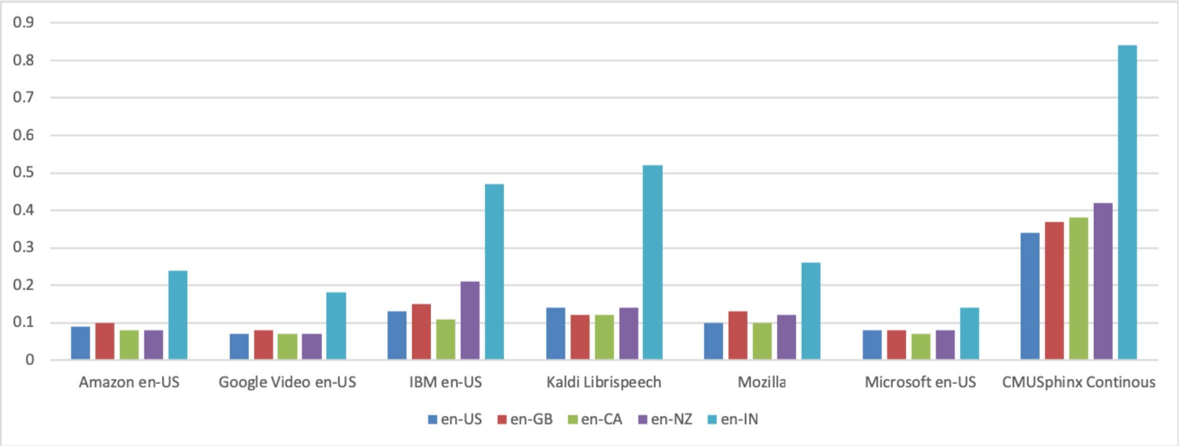


Figure 13: WER results for various dialects

Conclusions

The best result achieved with en-US model was for the Canadian dialect. Most challenging dialect appears to be Indian English.

The best performing system is Microsoft. The results for Google are worse than in previous scenarios because the default model, not the video model, was applied. For New Zealand English the same result as on Microsoft system was achieved by Amazon.

Dialect Experiment B: Transcription with corresponding dialect model

The dialect specific models are available in configurations of Google, Microsoft, IBM and Amazon:

- Google, Microsoft, Amazon and IBM for en-GB
- Google, Microsoft and Amazon for en-IN
- Google and Microsoft for en-CA and en-NZ

Sample duration	**105 min**	**10 min**	**8 min**	**4 min**	**4 min**
Category (Dialect)	**en-US**	**en-GB**	**en-CA**	**en-NZ**	**en-IN**
Amazon	0.09	0.09	-	-	0.33
IBM	0.13	0.20	-	-	-
Kaldi Librispeech	0.14	-	-	-	-
Google Default	0.16	0.12	0.15	0.16	0.56
Microsoft	0.08	0.09	0.09	0.18	0.41
Mozilla	0.10	-	-	-	-
Avg WER	0.12	0.13	0.12	0.17	0.43

Table 10: WER results for various dialects

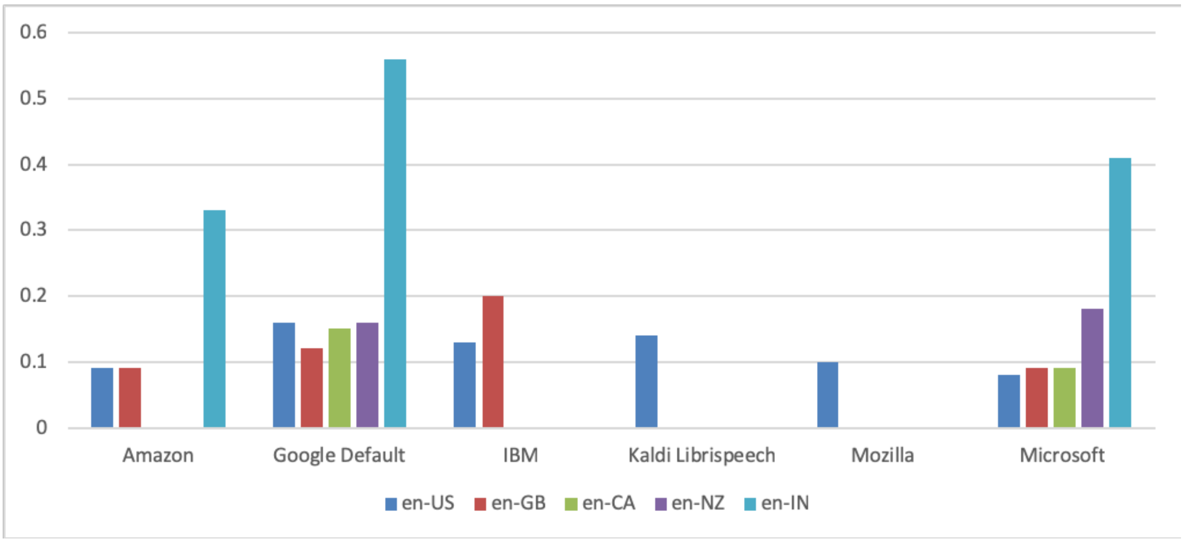


Figure 14: WER results for various dialects

Conclusions

It can be observed in this experiment that the most challenging dialect is the Indian one. An interesting observation is that the average WER value achieved on the default model is for all systems lower than when a dialect-specific model is applied. This tendency is clearly visible in Figure 15.

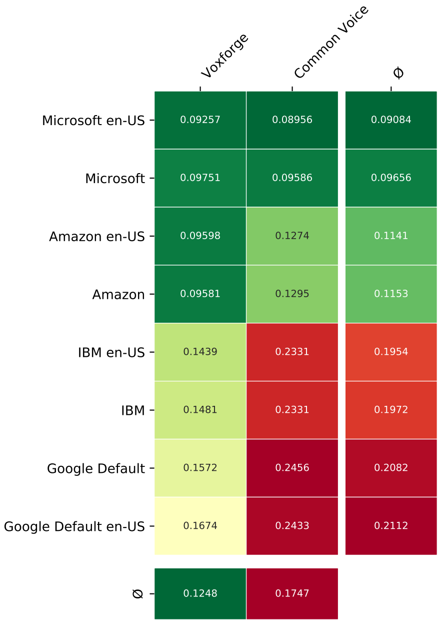


Figure 15: WER values depending on applied models

7.4.4 Accent

The goal of this experiment is to investigate the impact of native and non-native speaker accent on the ASR systems performance.

Speech recognition system when applied with utterances of non-native speakers suffer from higher error rates, compared to native speech. Some of main challenges which non-native speech presents for speech recognition are disfluency, accented pronunciation, pronunciation errors due to unfamiliarity with a word, errors in syntax, and syntax that is unusual but not incorrect.

Other difficulties include overemphasis of word boundaries. The irregularity of these deviations makes the speech recognition task even more challenging. [58] Various techniques, such as acoustic model adaptation and pronunciation adaptation, have been reported to improve the recognition of non-native or accented speech. [62] However, the ASR systems are mostly trained with native speech, which leads to a discrepancy between WER of native versus non-native speech recognition.

In the data set used for this evaluation only two corpora contain non-native speech utterances: AMI and RT. A sample of native speakers' and non-native speakers' utterances from the two corpora is used for the experiment.

Sample Selection

Source corpora:	AMI, RT
Sample size:	29.5 min (541 utterances)
Utterance duration:	between 2 and 6 seconds
Speaking rate:	all
Dialect:	all
Accent:	see section "Results"
Overlapping speech:	no
Non-lexical filler words:	no

Results

Sample duration	**10 min**	**19.4 min**
Category	**Native Speech**	**Non-native speech**
Amazon en-US	0.17	0.23
Google Video en-US	0.12	0.19
IBM / IBM en-US	0.21	0.33
Kaldi ASpIRE	0.21	0.38
Mozilla Rounded	0.34	0.49
Microsoft en-US	0.16	0.19
CMUSphinx Continuous	0.52	0.6
Avg WER	0.271	0.369

Table 11: WER results for native and non-native speech

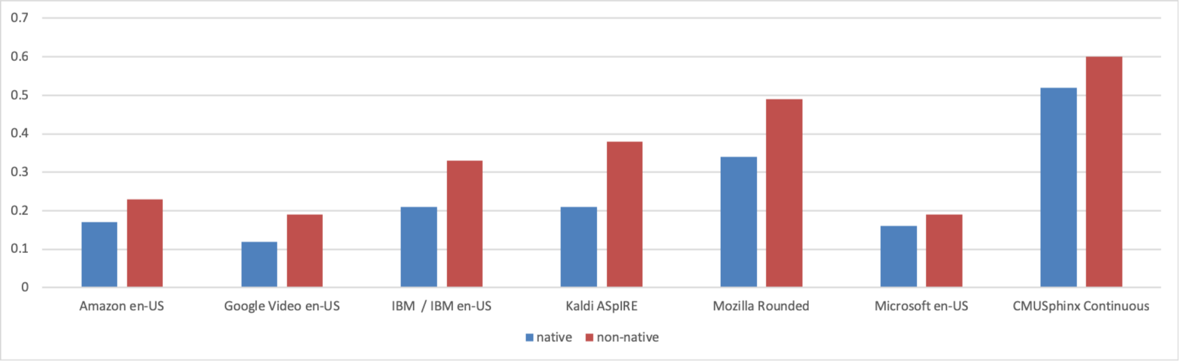


Figure 16: WER results for native and non-native speech

Conclusions

The experiment provides evidence for the observations presented in [58]: Non-native speech recognition is essentially more challenging than native speech. This can be observed for all systems.

Google achieved in this experiment the best result in the native speech category. On non-native speech utterances, the accuracy of Google and Microsoft systems was the highest.

7.4.5 Non-lexical filler words

The goal of this experiment is to investigate how the occurrence of filled pauses (also called filler words) in an utterance affect the ASR systems performance.

Disfluencies, such as repetitions, filled pauses, and hesitations can seriously affect the word recognition accuracy of an ASR system. Firstly, they make utterances longer without adding semantic information. An ASR tries to interpret a disfluency, which results in inaccurate transcripts being produced. Secondly, ASR systems are trained on well-structured sentences, but as the disfluencies add spurious content, they cause a mismatch between training and evaluation data that result in poor transcription. Since an ASR system makes predictions based on its language model, disfluencies affect correct recognition of words that are likely to follow each other. All of this may lead to a higher number of substitution, insertion and deletion errors. [32] [27]

We selected three sets of utterances from two conversational corpora: AMI and RT. The first set comprises utterances which do not contain any filled pauses. The second set consists of utterances containing only filled pauses, while the third set is a collection of utterances where filled pauses occur between words. We classified the following fillers as filled pauses: “mm”, “oh”, “uhm”, “hmm”, “uh-huh”, “ah”, “eh”, “er”, “uh”, “um”, “mhm”.

Sample Selection

Source corpora:	AMI, RT
Sample size:	31 min (491 utterances)
Utterance duration:	between 2 and 6 seconds
Speaking rate:	between 100 and 200 words per minute
Dialect:	en-US
Accent:	native
Overlapping speech:	no
Non-lexical filler words:	see section “Results”

Results

Sample duration	**14 min**	**14 min**	**17 min**	**0.16**
Category	**Originally contains fp**	**Fp removed from original**	**Originally contains no fp**	**Originally contains only fp**
Amazon en-US	0.34	0.35	0.21	0.61
Google Video en-US	0.36	0.23	0.19	0.75
IBM en-US	0.42	0.28	0.28	0.77
Kaldi ASpIRE	0.31	0.44	0.28	0.55
Mozilla / Mozilla Rounded	0.57	0.38	0.39	1
Micrsoft en-US	0.33	0.31	0.19	0.52
CMUSphinx Continous	0.67	0.57	0.55	0.93
Avg WER	0.45	0.37	0.31	0.77

Table 12: WER results for experiment with filled pauses

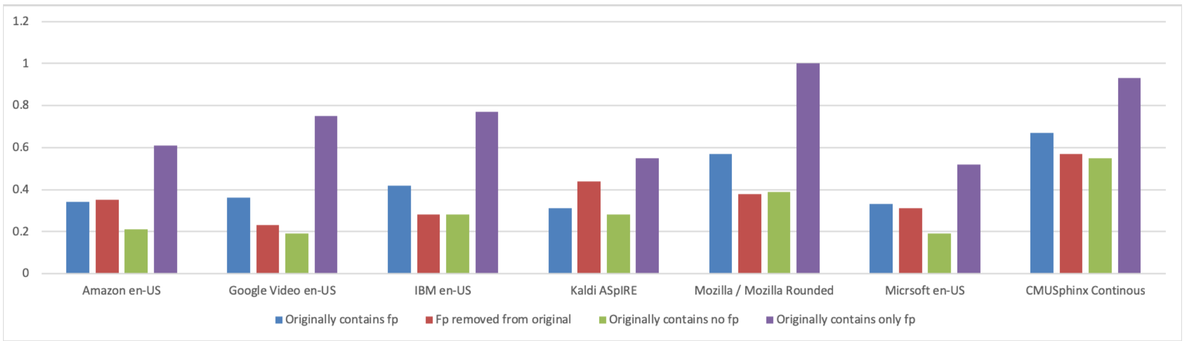


Figure 17: WER results for experiment with filled pauses

Conclusions

WER is by far the highest in the category where the utterance contains only filled pauses. This shows that filled pauses recognition is a challenging task. This may have several reasons, such as missing commonly agreed spelling of filled pauses or a high level of acoustic similarity between filler words. Filler words that are often confused are for instance hum-um, mhm-hmm and uh-um. **??** provides examples for these three phenomena.

When investigating the first two categories, it can be observed that various systems deal with filled pauses in a different way. The two categories contain exactly the same utterances. In the first category there are filled pauses in each of the utterance. In the second category these filled pauses are removed. Google's result clearly improves after the filled pauses were removed, while Kaldi's performance deteriorates. When manually analyzing some hypothesis examples, it could be seen that the filled pauses are frequently not contained in Google's transcriptions, while Kaldi includes them in the transcriptions (see Table 13).

Reference	**Hypothesis Google**	**Hypothesis Kaldi**
it's very **um** wide spectrum on scope and it requires a lot of technical ability so actually for me the hardest thing	it's very wide spectrum on scope and it requires a lot of technical ability so actually for me the hardest thing	it's very um wide spectrum an scope and it requires a lot of technical ability so actually for me the hardest thing
um for maryland distinguished scholars	four maryland distinguished scholars	**um** for maryland distinguished scholars

Table 13: WER results for experiment with filled pauses

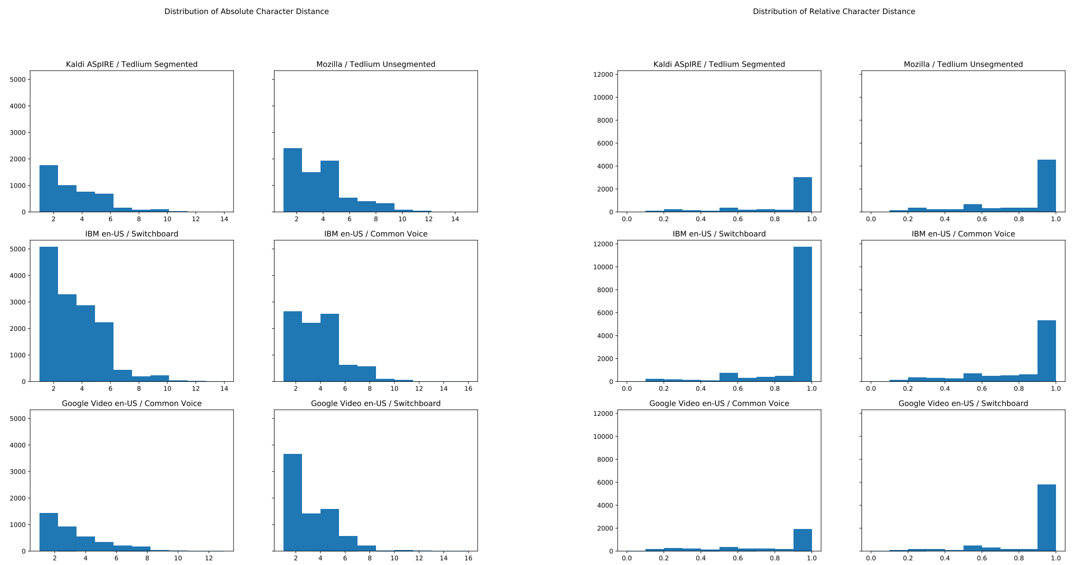
7.5 Fine grained WER Analysis

As was described before in subsubsection 5.4.2 the WER has its drawbacks. In this section, we will attempt a more detailed analysis of this metric.

7.5.1 Character Distance Analysis

In the scope of the character analysis we wanted to investigate how many characters of a word classified as wrong are actually incorrect. The hypothesis — based on observation — was, that a lot of the errors are actually just one or two letters missing or exchanged. And that the main part of the word remains unchanged and therefore still legible. When looking at the distribution of the absolute character distances in Figure 18b that seems to be the case at first glance. Most errors are on the left-hand side of the distribution — meaning only on or a few characters were wrong.

But taking a look at Figure 18a suggests something else: Almost all of the errors are on the very right hand side of the distribution meaning 90 %+ of the word is actually wrong.



(a) Distribution of absolute character distance.

(b) Distribution of relative character distance. The last column to to right contains all errors with a relative distance of 100% or higher. The character distance can be higher than 100% when more characters were inserted in the hypothesis than present in the reference.

Figure 18: Absolute and relative character distance distribution

Conclusion

The distributions from both Figure 18a and Figure 18b suggest that most errors happen on relatively short words. That's why the absolute character distances are on the lower side while the relative distributions are on the high side of the distribution. Of course the character distance on it's own is not sufficient to judge the correctness of the WER metric. Not every word the same importance and some words can even be omitted without losing information and context. In the next section we will look at specific error types to understand better what kinds of error the WER encompasses.

7.5.2 Linguistic Analysis

We will review the results of this experiment by looking at some specific examples. While we did calculate the results for every system and corpus configurations, the outcome was similar in such a way that reviewing them all one by one wouldn't bring any added value. Figure 19 shows the results for the Google Video en-US configuration. We grouped the corpora into the previously already seen categories of *read speech*, *semi-spontaneous speech* and *spontaneous speech* to get a more general idea based on speechj characteristics instead of corpus specific results. It's instantly apparent that spontaneous speech once more gives very different result from the rest. A very large chunk of the error — 22.88 % here — can be attributed to wrongly transcribed filled pauses. Depending on the use case these are probably not errors that are critical as filled pauses do not contain any information whatsoever.

Looking at the other categories we can see that filled pauses do not contribute much to the overall error. This is to be expected as filled pauses are a characteristic for spontaneous speech and occur much less in semi-spontaneous and even less in read speech. There are two other noticeable error types here though: Compound words and homonyms. The former is — much like filled pauses — in most cases probably not as big a problem as the 10 % share of the overall error may signal as it's mostly a syntactical tool which does not change meaning.

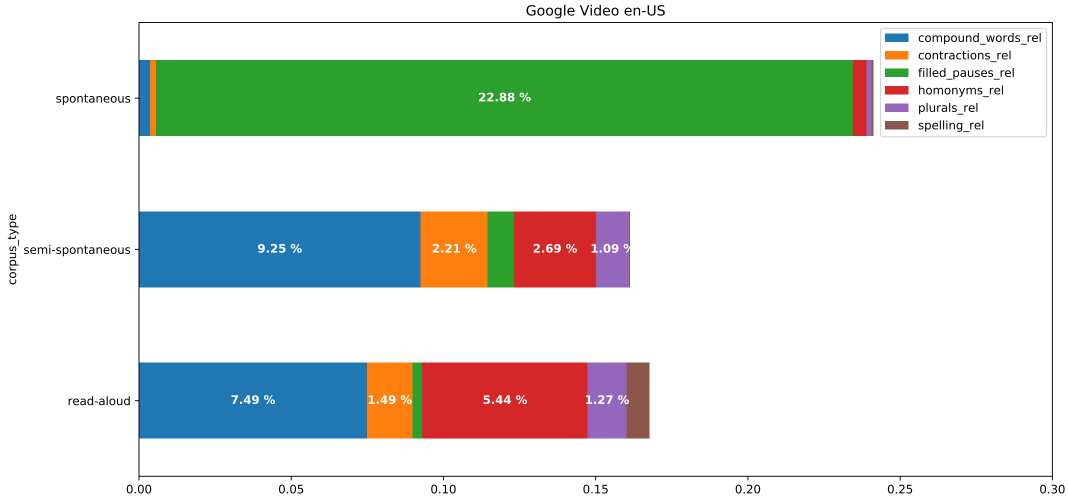


Figure 19: Linguistic Analysis for Google Video en-US

This is not the case for the homonyms though. While they sound the same when articulated, the spelling can be quite different and when reading a wrongly transcribed homonym can lead to confusion. A reader that's familiar with the challenges of speech recognition might realize quite quickly what the problem is, but it certainly disturbs the flow of reading. As homonyms are acoustically indiscernible these kind of errors can only be fixed with language model.

While the remaining investigated error types occur, they do so in much smaller numbers. This is mostly consistent over all systems and corpora.

Conclusion

Even though the number of error types investigated remained limited due to time constraints, we were able to observe that the WER really does classify errors of very different gravity. Especially when the use case is not a verbatim transcription but rather a comprehensible text representation.

7.6 Conclusions

The overall results of the evaluation showed, that the proprietary cloud solutions did outperform the open source system in practically all domains, with the solutions of Google, Microsoft and Amazon in the lead. The performance of these systems was very consistent over all speech corpora. However, it needs to be mentioned at this point, that is obviously unclear on what data the proprietary systems were trained and that theoretically, there could be an overlap between the systems training data and the evaluation data applied in our study.

The IBM system surprisingly lagged behind the three front runners and was occasionally placed closer to the good open-source systems than the other cloud services. We have made different attempts of improving the performance of the IBM transcriptions including trying out various models, testing different audio encodings, checking alternative transcriptions and even comparing our own results with the ones returned from the demo application on IBM's website. We could not find any issue in our implementeation and resolved to use the transcriptions as they were.

Despite the worse performance of the open source systems we did not conclude that these systems' performance is necessarily worse. Due to time constraints, we were forced to use pre-trained models. And while those were not able to perform as well as the cloud solutions, it was clearly recognizable that these systems can give good results too, if trained for the required domain or recognition task. This was especially evident in case of a surprisingly good performance of the Kaldi ASpIRE model on spontaneous speech. Its performance was very close to the top proprietary systems. The same observation is valid for Mozilla DeepSpeech whose Model is reportedly trained on Fisher, LibriSpeech, Switchboard and parts of Common Voice. DeepSpeech's performance is more balanced than the one of Kaldi, whose models perform very different across our evaluation corpora. DeepSpeech also achieves comparatively better results on the corpora it was trained on. The only system which, throughout the whole evaluation, performed worse than almost any other system was CMUSphinx sphinx4. However, this was not that surprising as it uses a very traditional setup and is the only system not using DNNs. It is worth mentioning, however, that CMUSphinx is also the only system that provides a solution applicable offline on a mobile device: CMUSphinx pocketsphinx. All other solutions are far to computation-intensive for lower powered devices. We did not include pocketsphinx in this evaluation because it oes not support LVCSR which is the focus of our evaluation.

The investigation of the correlations between spoken language properties and the transcription accuracy was concluded with a few observations. The most evident one was the substantial difference in the transcription accuracy between spontaneous dialogue speech and non-spontaneous monologue speech. The discrepancy between the average results from the two categories were in some cases even threefold. According to expectations, the experiments provided evidence that non-native speech contributes to a higher error rate that native speech. Interestingly, we could also see that applying dialect-specific models for transcribing utterances spoken in the corresponding dialect does not lead to transcription quality enhancement. On the contrary, better results on non-american utterances were achieved with the defauld en-US model for all systems. The reason being most probably, the size of the data set that the en-US model was trained on. No unambiguous correlation between the utterance duration and the accuracy, neither between speaking rate and accuracy could have been detected. An interesting observation was that the fast- and the slow-paced speech seemed to be transcribed with better quality than the middle-pace ones, however may have the reason in the disproportion of the samples size for different speaking rates.

The fine-grained WER analysis provided evidence that this metric does not always reflect the proportion of information preserved in the transcription. Two of the examined error types – filled pauses and compound words – made up surprisingly large portions of the error rate. And while transcripts including such errors might not be verbatim, these errors have arguably very little impact on readability and comprehensibility.

8 Discussion and Outlook

Performing an evaluation with over 60 system – corpus pairs is a challenging task. One of the main reasons being, the diversity of corpora and systems implementations. In case of corpora this diversity relates to their structure, the way the meta data is stored, the segmentation of the audio and reference files, reference and audio file formats. The challenge with the systems refers to getting familiar with their implementations and their various configuration options as well as trying these configurations out. Investigating both corpora and systems is a time and effort consuming phase, which should not be underestimated.

As a result of applying several corpora, another challenge occurs: handling large amounts of data. This was not evident at the beginning of the implementation but as time went by and the number of transcriptions increased, the duration of the end-to-end evaluation process, we implemented, essentially increased. Caching of the results from particular stages of the evaluation solved the problem to some extent. However, when planning an undertaking such as a relatively large-scale ASR evaluation, much attention should be focused on the performance aspect.

The amount of data is clearly crucial for the reliability of such an evaluation. However, not only the amount but also the diversity of data is essential in case of LVCSR systems. It is worth analyzing in detail beforehand, what data sets should be selected for an evaluation with regards to utterance properties. Although the corpora set used for our evaluation was relatively large, the diversity of the corpora eventually turned out to be limited. The issue with corpora selection is the frequently incomplete information about the available meta data in corpora documentation. Only after downloading and investigating in every detail the corpora structure, some first conclusions can be drawn about the diversity of the utterances. Another aspect is the quality and completeness of the data. In some cases, only after integrating the corpus in an evaluation framework, the quality of the corpus can be investigated. This is obviously time consuming and despite the effort invested, the corpus may be eventually be excluded from the evaluation. This was the experience we made with Wikipedia corpus SWC, which was finally not part of this evaluation.

When planning an evaluation of multiple systems, enough time must be planned to extensively test various implementations and implement appropriate error handling. Our experience shows that some errors can be discovered only when larger quantities of transcriptions are performed.

An important issue in the general discussion about ASR evaluation, is its reliability. There are several aspects which may limit the reliability. One of them is the reference processing, especially in case of conversational speech. It must be decided what elements from the reference need to be removed, and for instance how to handle false starts: when a speaker starts saying a word but stops before the word is completely articulated. The question arises, if this part of word should remain in the reference or should be removed. Another aspect is spelling of numbers, dates or acronyms. A number spelled versus a number transcribed as a digit lead to a substitution detection in the alignment of reference and hypothesis, even though the word was recognized correctly.

Another aspect of reliability refers to data sets that are applied for evaluation versus data sets that the system was trained on. It is hard to judge two systems based on utterances from a corpus, if one was trained on the training set of this corpus and the other one not. And obviously, this information is not public for most system, hence no conscious decision can be made with regard to corpora selection.

Comparative study of several systems with multiple corpora provides an opportunity to investigate a vast variety of aspects related to ASR. It would be interesting to investigate in more detail the other evaluation metrics and compare the results for the same set of corpus – system pairs. Comparing the metrics measuring information preserved in the transcription with the WER results would allow to better understand the relation between WER value and semantic accuracy. The correlation between spoken language properties and transcription accuracy could also be further investigated. Especially interesting and worth investigating seem the challenges of overlapping speech. Furthermore, the aspects related to recording setup, such as noise, acoustic environment and recording device, which were not part of this evaluation, could provide some interesting insights into the challenges of ASR.

References

[1] Ahmed Ali and Steve Renals. "Word error rate estimation for speech recognition: e-WER". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 20–24.

[2] Dario Amodei et al. "Deep speech 2: End-to-end speech recognition in english and mandarin". In: *International conference on machine learning*. 2016, pp. 173–182.

[3] *asr-evaluation Package Documentation*. URL: `https://github.com/belambert/asr-evaluation` (visited on June 1, 2019).

[4] Fadi Biadsy. "Automatic dialect and accent recognition and its application to speech recognition". PhD thesis. Columbia University, 2011.

[5] Fadi Biadsy, Pedro J Moreno, and Martin Jansche. "Google's cross-dialect Arabic voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 4441–4444.

[6] Joan Cabestany, Ignacio Rojas, and Gonzalo Joya. *Advances in Computational Intelligence: 11th International Work-Conference on Artificial Neural Networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings*. Vol. 6692. Springer, 2011.

[7] Joseph P Campbell and Douglas A Reynolds. "Corpora for the evaluation of speaker recognition systems". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 2. IEEE. 1999, pp. 829–832.

[8] Wu Chou and Biing-Hwang Juang. "Toward spontaneous speech recognition and understanding". In: *Pattern recognition in speech and language processing*. CRC Press, 2003, pp. 188–223.

[9] *CMUSphinx US English Models*. 2016. URL: `https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English/` (visited on Mar. 3, 2019).

[10] Kelly Davis. *Deep Speech 0.4.1 - Release Notes*. Mozilla. 2019. URL: `https://github.com/mozilla/DeepSpeech/releases/tag/v0.4.1`.

[11] Laila Dybkjær, Holmer Hemsen, and Wolfgang Minker. *Evaluation of text and speech systems*. Vol. 38. Springer Science & Business Media, 2007.

[12] Mohamed G Elfeky, Pedro Moreno, and Victor Soto. "Multi-dialectical languages effect on speech recognition: Too much choice can hurt". In: *Procedia Computer Science* 128 (2018), pp. 1–8.

[13] *End-to-End Models for Speech Processing*. 2017. URL: `https://www.youtube.com/watch?v=3MjIkWxXigM` (visited on June 2, 2019).

[14] Rahhal Errattahi, Asmaa El Hannani, and Hassan Ouahmane. "Automatic speech recognition errors detection and correction: A review". In: *Procedia Computer Science* 128 (2018), pp. 32–37.

[15] Benoit Favre et al. "Automatic human utility evaluation of ASR systems: Does WER really predict performance?" In: *INTERSPEECH*. 2013, pp. 3463–3467.

[16] Eric Fosler-Lussier and Nelson Morgan. "Effects of speaking rate and word frequency on pronunciations in convertional speech". In: *Speech Communication* 29.2-4 (1999), pp. 137–158.

[17] Christian Gaida et al. "Comparing open-source speech recognition toolkits". In: *Tech. Rep., DHBW Stuttgart* (2014).

[18] Olivier Galibert. "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech." In: *INTERSPEECH*. 2013, pp. 1131–1134.

[19] John S Garofolo, Cedric GP Auzanne, and Ellen M Voorhees. "The TREC spoken document retrieval track: A success story". In: *Content-Based Multimedia Information Access-Volume 1*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. 2000, pp. 1–20.

[20] Jürgen T Geiger et al. “Using linguistic information to detect overlapping speech”. In: *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France.* 2013.

[21] Sharon Goldwater, Dan Jurafsky, and Christopher D Manning. “Which words are hard to recognize? Prosodic, lexical, and disfluency factors that increase speech recognition error rates”. In: *Speech Communication* 52.3 (2010), pp. 181–200.

[22] Marıa González et al. “An illustrated methodology for evaluating ASR systems”. In: *International Workshop on Adaptive Multimedia Retrieval.* Springer. 2011, pp. 33–42.

[23] Awni Hannun et al. “Deep speech: Scaling up end-to-end speech recognition”. In: *arXiv preprint arXiv:1412.5567* (2014).

[24] Hany Hassan et al. “Segmentation and disfluency removal for conversational speech translation”. In: *Fifteenth Annual Conference of the International Speech Communication Association.* 2014.

[25] *IBM Watson STT Documentation.* 2019. URL: `https://cloud.ibm.com/apidocs/speech-to-text?code=python` (visited on Apr. 11, 2019).

[26] *jiwer Package Documentation.* URL: `https://pypi.org/project/jiwer/` (visited on June 1, 2019).

[27] Douglas A Jones et al. “Measuring the readability of automatic speech-to-text transcripts”. In: *Eighth European Conference on Speech Communication and Technology.* 2003.

[28] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2Nd Edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009. ISBN: 0131873210.

[29] Preethi Jyothi. *Automatic Speech Recognition - An Overview.* Microsoft Research. 2017. URL: `https://www.microsoft.com/en-us/research/video/automatic-speech-recognition-overview/#!related_info`.

[30] S Karpagavalli and E Chandra. “A review on automatic speech recognition architecture and approaches”. In: *International Journal of Signal Processing, Image Processing and Pattern Recognition* 9.4 (2016), pp. 393–404.

[31] S Karpagavalli et al. “Automatic Speech Recognition: Architecture, Methodologies and Challenges-A Review”. In: *International Journal of Advanced Research in Computer Science* 2.6 (2011).

[32] Mayank Kaushik, Matthew Trinkle, and Ahmad Hashemi-Sakhtsari. “Automatic detection and removal of disfluencies from spontaneous speech”. In: *Proceedings of the Australasian International Conference on Speech Science and Technology (SST).* 2010.

[33] Veton Këpuska and Gamal Bohouta. “Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx)”. In: *Int. J. Eng. Res. Appl* 7.03 (2017), pp. 20–24.

[34] Joshua Y Kim et al. “A Comparison of Online Automatic Speech Recognition Systems and the Nonverbal Responses to Unintelligible Speech”. In: *arXiv preprint arXiv:1904.12403* (2019).

[35] Christian Landsiedel and Mark Hillebrand. *Azure Speech Continous Sample.* 2019. URL: `https://github.com/Azure-Samples/cognitive-services-speech-sdk/blob/master/samples/python/console/speech_sample.py#L193` (visited on May 4, 2019).

[36] *LDC Top Ten Corpora.* URL: `https://catalog.ldc.upenn.edu/topten` (visited on June 1, 2019).

[37] Iain A McCowan et al. *On the use of information retrieval measures for speech recognition evaluation.* Tech. rep. IDIAP, 2004.

[38] Mehryar Mohri, Fernando Pereira, and Michael Riley. “Speech recognition with weighted finite-state transducers”. In: *Springer Handbook of Speech Processing.* Springer, 2008, pp. 559–584.

[39] Darren C Moore and Iain A McCowan. “Microphone array speech recognition: Experiments on overlapping speech in meetings”. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03).* Vol. 5. IEEE. 2003, pp. V–497.

[40] Fabrizio Morbini et al. “Which ASR should I choose for my dialogue system?” In: *Proceedings of the SIGDIAL 2013 Conference.* 2013, pp. 394–403.

[41] Andrew Morris. *An information theoretic measure of sequence recognition performance.* Tech. rep. IDIAP, 2002.

[42] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques". In: *arXiv preprint arXiv:1003.4083* (2010).

[43] Murray J Munro and Tracey M Derwing. "The effects of speaking rate on listener evaluations of native and foreign-accented speech". In: *Language Learning* 48.2 (1998), pp. 159–182.

[44] Cosmin Munteanu et al. "The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM. 2006, pp. 493–502.

[45] AB Nagorski, LWJ Boves, and Herman Steeneken. "Optimal selection of speech data for automatic speech recognition systems". In: (2002).

[46] Maryam Najafian. "Modeling accents for automatic speech recognition". In: (2013).

[47] Masanobu Nakamura, Koji Iwano, and Sadaoki Furui. "Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance". In: *Computer Speech & Language* 22.2 (2008), pp. 171–184.

[48] Hiroaki Nanjo and Tatsuya Kawahara. "A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding". In: *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 1. IEEE. 2005, pp. I–1053.

[49] *NIST SCLITE Scoring Package Documentation.* URL: `http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm` (visited on June 1, 2019).

[50] Douglas O'Shaughnessy. "Automatic speech recognition: History, methods and challenges". In: *Pattern Recognition* 41.10 (2008), pp. 2965–2979.

[51] Youngja Park et al. "An empirical analysis of word error rate and keyword error rate". In: *Ninth Annual Conference of the International Speech Communication Association.* 2008.

[52] Arnab Poddar, Md Sahidullah, and Goutam Saha. "Performance comparison of speaker recognition systems in presence of duration variability". In: *2015 Annual IEEE India Conference (INDICON).* IEEE. 2015, pp. 1–6.

[53] Arnab Poddar, Md Sahidullah, and Goutam Saha. "Quality measures for speaker verification with short utterances". In: *Digital Signal Processing* 88 (2019), pp. 66–79.

[54] Florian Schiel et al. *The production of speech corpora.* 2012.

[55] Takahiro Shinozaki and Sadaoki Furui. "Error Analysis Using Decision Trees in Spontaneous Presentation". In: *Book name IEEE Workshop on Automatic Speech Recognition and Understanding.*

[56] Matthew A Siegler and Richard M Stern. "On the effects of speech rate in large vocabulary speech recognition systems". In: *1995 international conference on acoustics, speech, and signal processing.* Vol. 1. IEEE. 1995, pp. 612–615.

[57] Alexander Solovets, nshmyrev, and bic-user. *Sphinx4 Transcriber Demo.* 2015. URL: `https://github.com/cmusphinx/sphinx4/blob/master/sphinx4-samples/src/main/java/edu/cmu/sphinx/demo/transcriber/TranscriberDemo.java` (visited on Mar. 3, 2019).

[58] Laura Mayfield Tomokiyo. "Handling non-native speech in lvcsr: A preliminary study". In: *Proceedings of the EUROCALL/CALICO/ISCA workshop on Integrating Speech Technology in (Language) Learning (InSTIL).* 2000.

[59] Laura Mayfield Tomokiyo. "Recognizing non-native speech: characterizing and adapting to non-native usage in LVCSR". PhD thesis. PhD Thesis, Carnegie Mellon University, 2001.

[60] Ye-Yi Wang, Alex Acero, and Ciprian Chelba. "Is word error rate a good indicator for spoken language understanding accuracy". In: *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721).* IEEE. 2003, pp. 577–582.

[61] Julián Zapata and Andreas Søeborg Kirkedal. "Assessing the Performance of Automatic Speech Recognition Systems When Used by Native and Non-Native Speakers of Three

Major Languages in Dictation Workflows". In: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*. 2015, pp. 201–210.

[62] Yanli Zheng et al. "Accent detection and speech recognition for shanghai-accented mandarin". In: *Ninth European Conference on Speech Communication and Technology*. 2005.

Glossary

acoustic environment refers in this report to the environment, where the recording of the speech has taken place.. 27

acoustic model is used in automatic speech recognition to represent the relationship between an audio signal and the phonemes or other linguistic units that make up speech. The model is learned from a set of audio recordings and their corresponding transcripts.. 9

alignment refers in this report to the process of comparing the reference text with its transcript.. 7

audio encoding is the process of changing digital audio from one format to another. 29

audio signal is a representation of sound, typically using a level of electrical voltage for analog signals, and a series of binary numbers for digital signals.. 29

bit depth is the number of bits of information per signal sample. A higher audio bit depth indicates a more detailed sound recording.. 29

channel is a representation of sound coming from a single device, such as a microphone or going to a single device such as a speaker. A digital audio file can contain multiple channels of data.. 29

confusion pair is a pair of words, one coming from the reference text and one being its incorrectly transcribed equivalent in the transcription. 7

continuous speech is speech which is produced as a continuous stream.. 10, 20

decoder is a component of an ASR system which applies acoustic, phonetic and language model to find the word sequence that best matches the signal provided in the input audio.. 6

deep learning a class of machine learning algorithms applying artificial neural networks. A neural network is a composition of a large number of highly interconnected processing elements (neurones) working in parallel to solve a specific problem. Neural networks take examples as inputs and learn from them to solve the problem.. 5

edit distance is a way of quantifying how dissimilar two strings are to one another by counting the minimum number of operations required to transform one string into the other. 17

feature extraction refers in this report to a technique for analyzing and characterizing audio content. It consists in transforming the original audio data to a data set with a reduced number of features meant to contain most information and be non-redundant.. 10

filled pause is a form of speech disfluency. It is aspoken sound or word used to fill gaps in speech, such as for instance "uh" and "um".. 5

hit is a word recognized correctly by an ASR system. 18

hypothesis is an automatic transcription of an audio recording which is used as a basis for ASR evaluation.. 7

in-meeting assistants is a system designed to transcribe meetings and processing the meeting information for instance by identifying action items and decisions.. 5

Information Retrieval comprises methods for organizing large amounts of structured and unstructured data to enable the efficient retrieval of relevant information coressponding to a particular query. 18

isolated word recognition relates to recognizing words in isolation not as part of continuous speech. The beginning and the end of each word can be detected directly from the energy of the signal.. 14

language model provides context for an ASR system to distinguish between words and phrases that sound similar. It estimates the relative likelihood of different phrases.. 12

Large-Vocabulary Continuous Speech Recognition System is a system for automatically recognizing continuous speech. Its vocabulary is not limited to a particular domain. 5, 56

Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. It is named after the Soviet mathematician Vladimir Levenshtein.. 17

Machine Translation is an automated translation of text from one language to another without human involvement.. 12

overlapping speech occurs if two or more speakers speak simultaneously.. 6

phone is the smallest identifiable unit found in a stream of speech. In case of phones, it is irrelevant if the sound is critical to the meanings of words.. 10, 35

phoneme is the smallest unit of language that may change the meaning of a word, for instance /p/ – /b/ in pig - big .. 10

phonetic model is used for translating phone sequences into lexical text. Creating a phonetic model requires involvement of linguists, it is not trained by the machines.. 9

precision is a measure of relevance used in Information Retrieval. It is computed as the fraction of retrieved documents that are relevant to the query.. 18

pronunciation model is a component of an ASR systems which determines how sound units can be combined to build words.. 11

recall is a measure of relevance used in Information Retrieval. It is computed as the fraction of the relevant documents that are successfully retrieved.. 18

reference is a manual transcription of an audio recording which serves as a reference for ASR evaluation.. 7

sample rate measures how frequent a signal is sampled per time unit. It is usually expressed in kiloHertz (kHz), which means 1,000 times per second.. 29

speaker-adapted system initially uses speaker-independent models, and then gradually adapts to the specific aspects of the speaker using the system.. 6

speaker-independent system is applied for recognizing speech of any speaker. The speech model must be general enough to cover all types of voices and all possible ways of word pronunciations.. 5, 6

speaking rate is the speech tempo. It is most often expressed in words per minute (wpm). 5, 6, 35, 37

speech analytics is the process of analyzing recorded calls to gather customer information to improve communication and future interaction. It is primarily used by customer contact centers to extract information buried in client interactions with an enterprises. It can include among others automatic transcription of conversations and analysis of calls for any positive or negative sentiments.. 18

speech corpus is a collection of digital recordings of speech together with their annotations, meta data, and documentation.. 5

speech rate is the speech tempo. It is most often expressed in words per minute (wpm). 35

speech-to-text is a process of translating natural language into text by a machine. 5, 56

statistical modelling 8

transcript is the result of a translation of speech input into text.. 8

utterance refers in this report to a unit of speech which is provided as input for speech recognition. It can be a single word, a phrase, a complete or incomplete sentence, or multiple sentences.. 6, 7

virtual speech assistants is an application enabling the user to interact with a device by providing it with spoken commands.. 5

voice search is a solution for processing a search query provided as speech and retrieve relevant results.. 5

voice-command tool is an application enabling the user to interact with a device by providing it with voice commands.. 14

Word Error Rate is a metric for evaluating automatic speech recognition accuracy.. 5, 56

Acronyms

ANN Artificial Neural Network. 13

ASR Automatic Speech Recognition. 3, 5–20, 22, 24–26, 34

CER Character Error Rate. 27

CMUdict Carnegie Mellon University Pronouncing Dictionary. 12

CSR Command Success Rate. 5

DNN Deep Neural Network. 13, 24, 46

ELRA European Language Resources Association. 16

FLAC Free Lossless Audio Codec. 22, 26

FST finite-state transducer. 13

GMM Gaussian Mixture Model. 11, 13

HMM Hidden Markov Model. 8–11, 13, 24

ITU International Telecommunication Union. 6

KER Keyword Error Rate. 5, 18

LDC Linguistics Data Consortium. 16

LVCSR Large-Vocabulary Continuous Speech Recognition. 5, 13, 15, 20, 46, *Glossary:* Word Error Rate

MFCC Mel-frequency cepstral coefficients. 10

NIST National Institute of Standards and Technologies. 19

NLP Natural Language Processing. 12

OCR Optical character recognition. 12

RIL Relative Information Loss. 5

SCTK Scoring Toolkit. 19

SER Sentence Error Rate. 19

SR speaking rate. 35

SRILM SRI Language Modeling. 12

STT Speech-to-Text. 5, *Glossary:* speech-to-text

TF-IDF Term Frequency-Inverse Document Frequency. 18

UD utterance duration. 33

VM1 Verbmobil 1. 6

WAV RIFF WAVE. 22, 26

WCR Word Correct Rate. 18

WER Word Error Rate. 5, 6, 17–20, 25, 27, 28, 30–32, 34, 35, 43–45, *Glossary:* Word Error Rate

WFST Weighted Finite State Transducer. 13

WIL Word Information Loss. 5

WIP Word Information Preserved. 5, 18

WKER Weighted Keyword Error Rate. 5, 18

WRR Word Recognition Rate. 17–19

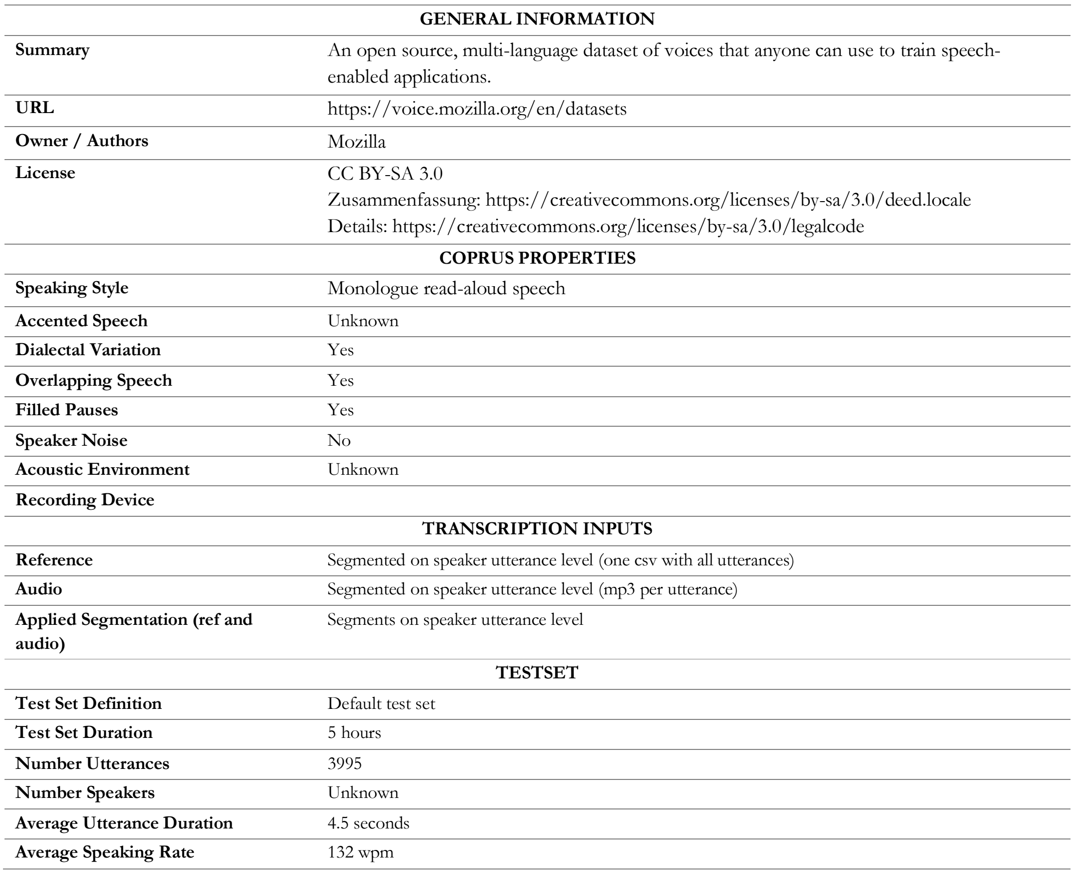
WSJ1 Wall Street Journal 1. 6, 36

A Corpora Documentation

A.1 AMI

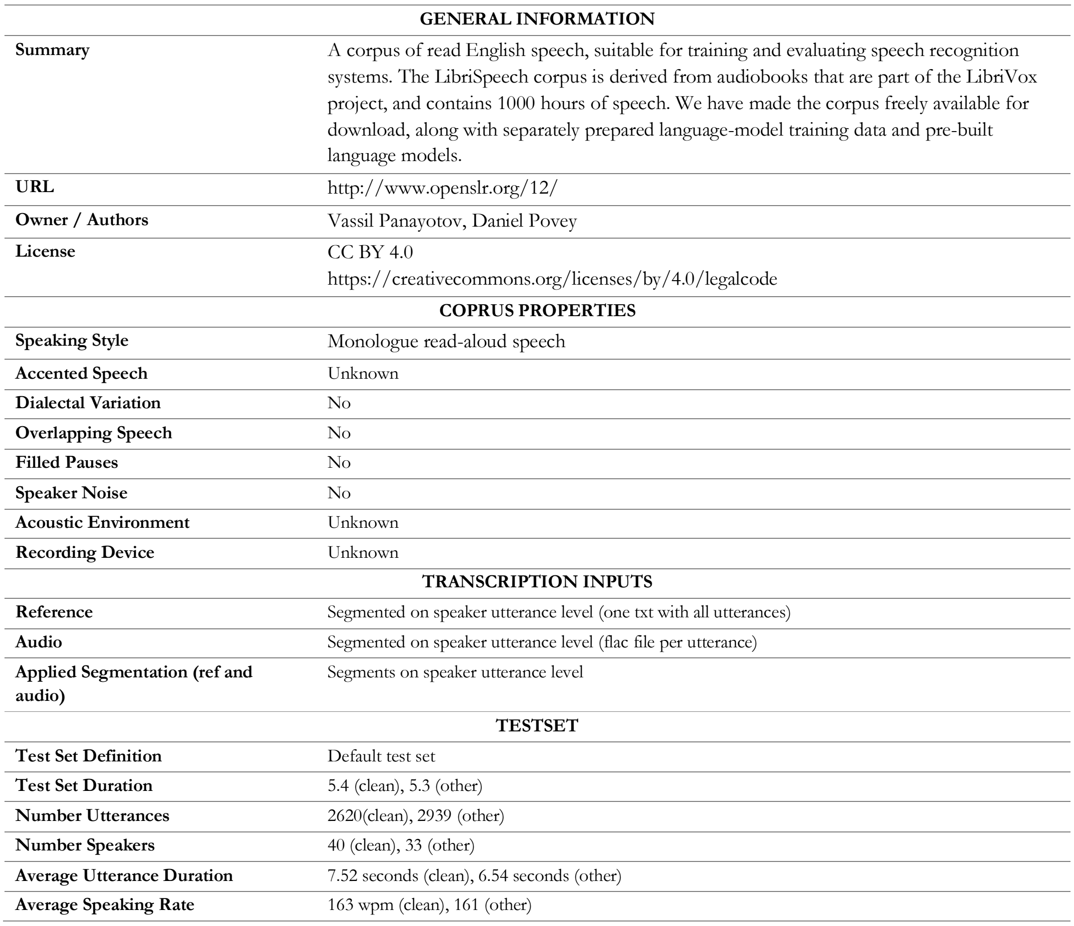
GENERAL INFORMATION	
Summary	The AMI Meeting Corpus is a multi-modal data set consisting of 100 hours of meeting recordings. Around two-thirds of the data has been elicited using a scenario in which the participants play different roles in a design team, taking a design project from kick-off to completion over the course of a day. The rest consists of naturally occurring meetings in a range of domains. Although the AMI Meeting Corpus was created for the uses of a consortium that is developing meeting browsing technology, it is designed to be useful for a wide range of research areas.
URL	http://groups.inf.ed.ac.uk/ami/corpus/
Owner / Authors	University of Edinburgh
License	Creative Commons Attribution 4.0 International Public License http://creativecommons.org/licenses/by/4.0/legalcode
COPRUS PROPERTIES	
Speaking Style	Dialog spontaneous speech
Accented Speech	Yes
Dialectal Variation	Yes
Overlapping Speech	Yes
Filled Pauses	Yes
Speaker Noise	Yes
Acoustic Environment	Meeting room
Recording Device	RealMedia audio mix, Headset mix, Lapel mix, Individual lapels, Individual headsets, Microphone array
TRANSCRIPTION INPUTS	
Reference	Segmented on word level (.xml file per speaker per meeting)
Audio	Unsegmented (one .wav file per meeting)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
TESTSET	
Test Set Definition	Random selection
Test Set Duration	5 hours
Number Utterances	4563
Number Speakers	38
Average Utterance Duration	4 seconds
Average Speaking Rate	141 wpm

A.2 Common Voice



GENERAL INFORMATION	
Summary	An open source, multi-language dataset of voices that anyone can use to train speech-enabled applications.
URL	https://voice.mozilla.org/en/datasets
Owner / Authors	Mozilla
License	CC BY-SA 3.0 Zusammenfassung: https://creativecommons.org/licenses/by-sa/3.0/deed.locale Details: https://creativecommons.org/licenses/by-sa/3.0/legalcode
COPRUS PROPERTIES	
Speaking Style	Monologue read-aloud speech
Accented Speech	Unknown
Dialectal Variation	Yes
Overlapping Speech	Yes
Filled Pauses	Yes
Speaker Noise	No
Acoustic Environment	Unknown
Recording Device	
TRANSCRIPTION INPUTS	
Reference	Segmented on speaker utterance level (one csv with all utterances)
Audio	Segmented on speaker utterance level (mp3 per utterance)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
TESTSET	
Test Set Definition	Default test set
Test Set Duration	5 hours
Number Utterances	3995
Number Speakers	Unknown
Average Utterance Duration	4.5 seconds
Average Speaking Rate	132 wpm

A.3 LibriSpeech



	GENERAL INFORMATION
Summary	A corpus of read English speech, suitable for training and evaluating speech recognition systems. The LibriSpeech corpus is derived from audiobooks that are part of the LibriVox project, and contains 1000 hours of speech. We have made the corpus freely available for download, along with separately prepared language-model training data and pre-built language models.
URL	http://www.openslr.org/12/
Owner / Authors	Vassil Panayotov, Daniel Povey
License	CC BY 4.0 https://creativecommons.org/licenses/by/4.0/legalcode
	COPRUS PROPERTIES
Speaking Style	Monologue read-aloud speech
Accented Speech	Unknown
Dialectal Variation	No
Overlapping Speech	No
Filled Pauses	No
Speaker Noise	No
Acoustic Environment	Unknown
Recording Device	Unknown
	TRANSCRIPTION INPUTS
Reference	Segmented on speaker utterance level (one txt with all utterances)
Audio	Segmented on speaker utterance level (flac file per utterance)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
	TESTSET
Test Set Definition	Default test set
Test Set Duration	5.4 (clean), 5.3 (other)
Number Utterances	2620(clean), 2939 (other)
Number Speakers	40 (clean), 33 (other)
Average Utterance Duration	7.52 seconds (clean), 6.54 seconds (other)
Average Speaking Rate	163 wpm (clean), 161 (other)

A.4 RT

GENERAL INFORMATION	
Summary	The evaluation data consists of an approximately 180-minute multi-site test set containing 7 meeting excerpts from 7 meetings. The test data was collected at EDI, IDI, and NIST. Each meeting excerpt contains a head-mic recording for each subject and one or more distant microphone recordings (whatever the data collection sites provided to NIST).
URL	https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation
Owner / Authors	Prepared by the National Institute of Standards and Technology (NIST) Multimodal Information Group and distributed by the Linguistic Data Consortium (LDC)
License	Linguistic Data Consortium (LDC) license
COPRUS PROPERTIES	
Speaking Style	Dialog spontaneous speech
Accented Speech	Yes
Dialectal Variation	Unknown
Overlapping Speech	Yes
Filled Pauses	Yes
Speaker Noise	Yes
Acoustic Environment	Meeting room
Recording Device	individual lapels, individual headsets, headset sum, microphone array, distant microphone, summed distant microphone, source localization arrays, KEMAR mannequin
TRANSCRIPTION INPUTS	
Reference	Segmented on speaker utterance level (one .tdf file with all speaker utterances per meeting)
Audio	Segmented on meeting level (one .sph file per meeting)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
TESTSET	
Test Set Definition	Random selection
Test Set Duration	3.6 hours
Number Utterances	6334
Number Speakers	30
Average Utterance Duration	2 seconds
Average Speaking Rate	195 wpm

A.5 Switchboard

GENERAL INFORMATION	
Summary	The Switchboard-1 Telephone Speech Corpus (LDC97S62) consists of approximately 260 hours of speech and was originally collected by Texas Instruments in 1990-1, under DARPA sponsorship. The first release of the corpus was published by NIST and distributed by the LDC in 1992-3. Since that release, a number of corrections have been made to the data files as presented on the original CD-ROM set and all copies of the first pressing have been distributed. Switchboard is a collection of about 2,400 two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States. A computer-driven robot operator system handled the calls, giving the caller appropriate recorded prompts, selecting and dialing another person (the callee) to take part in a conversation, introducing a topic for discussion and recording the speech from the two subjects into separate channels until the conversation was finished. About 70 topics were provided, of which about 50 were used frequently. Selection of topics and callees was constrained so that: (1) no two speakers would converse together more than once and (2) no one spoke more than once on a given topic.
URL	https://catalog.ldc.upenn.edu/LDC97S62
Owner / Authors	Collected by Texas Instruments in 1990-1, under DARPA sponsorship Authors: John J. Godfrey, Edward Holliman
License	Linguistic Data Consortium (LDC) license
COPRUS PROPERTIES	
Speaking Style	Dialog spontaneous speech
Accented Speech	Yes
Dialectal Variation	Unknown
Overlapping Speech	Yes
Filled Pauses	Yes
Speaker Noise	Yes
Acoustic Environment	Unknown
Recording Device	Phone
TRANSCRIPTION INPUTS	
Reference	Segemented on speaker utterance level (one .txt file with all utterances) Transcripts created by Mississippi State transcripts
Audio	Segemented on call level (one audio file per call)
Applied Segmentation (ref and audio)	Segements on speaker utterance level
TESTSET	
Test Set Definition	Random selection
Test Set Duration	5 hours
Number Utterances	4105
Number Speakers	483
Average Utterance Duration	4.39 seconds
Average Speaking Rate	128 wpm

A.6 ST

	GENERAL INFORMATION
Summary	A free American English corpus by Surfingtech (www.surfing.ai), containing utterances from 10 speakers, Each speaker has about 350 utterances. The data set is a subset of a much bigger data set (about 1000hours) which was recorded in the same environment as this open source data. This corpus were recorded in silence in-door environment using cellphone. It has 10 speakers. Each speaker has about 350 utterances. All utterances were carefully transcribed and checked by human. Transcription accuracy is guaranteed.
URL	http://www.openslr.org/45/
Owner / Authors	Surfingtech (www.surfing.ai)
License	Creative Common BY-NC-ND 4.0 https://creativecommons.org/licenses/by-nc-nd/4.0/deed.de
	COPRUS PROPERTIES
Speaking Style	Monologue read-aloud speech
Accented Speech	Unknown
Dialectal Variation	No
Overlapping Speech	No
Filled Pauses	No
Speaker Noise	No
Acoustic Environment	Silence in-door environment
Recording Device	Cellphone
	TRANSCRIPTION INPUTS
Reference	Segmented on speaker utterance level (one .txt with all utterances)
Audio	Segmented on speaker utterance level (one .wav file per utterance)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
	TESTSET
Test Set Definition	Random selection
Test Set Duration	4.7 hours
Number Utterances	3842
Number Speakers	5
Average Utterance Duration	4.44 seconds
Average Speaking Rate	109 wpm

A.7 TedLium

GENERAL INFORMATION	
Summary	The TED-LIUM corpus is English-language TED talks, with transcriptions, sampled at 16kHz. It contains about 452 hours of speech.
URL	https://www.openslr.org/51/
Owner / Authors	Created through a collaboration between the Ubiqus company and the LIUM (University of Le Mans, France)
License	Creative Commons BY-NC-ND 3.0
COPRUS PROPERTIES	
Speaking Style	Monologue semi-spontaneous speech
Accented Speech	Unknown
Dialectal Variation	No
Overlapping Speech	No
Filled Pauses	Yes
Speaker Noise	No
Acoustic Environment	Unknown (most probably audience hall)
Recording Device	Unknown
TRANSCRIPTION INPUTS	
Reference	Segmented on utterance level (one .stm file split in short utterances with time stamps)
Audio	Segemented on lecture level (one .sph file with the whole lecture recording)
Applied Segmentation (ref and audio)	Segments on speaker utterance level (segmented) Segments on lecture utterance level (unsegmented)
TESTSET	
Test Set Definition	Default test set
Test Set Duration	2.6 hours (segmented),
Number Utterances	1155 (segmented)
Number Speakers	11
Average Utterance Duration	8.15 seconds (segmented)
Average Speaking Rate	172 wpm

A.8 Timit

GENERAL INFORMATION	
Summary	The TIMIT corpus of read speech is designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI). The speech was recorded at TI, transcribed at MIT and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST). The TIMIT corpus transcriptions have been hand verified. Test and training subsets, balanced for phonetic and dialectal coverage, are specified. Tabular computer-searchable information is included as well as written documentation.
URL	https://catalog.ldc.upenn.edu/LDC93S1
Owner / Authors	Authors: John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, Victor Zue
License	Linguistic Data Consortium (LDC) license
COPRUS PROPERTIES	
Speaking Style	Monologue read-aloud speech
Accented Speech	No
Dialectal Variation	No
Overlapping Speech	No
Filled Pauses	Yes
Speaker Noise	No
Acoustic Environment	Unknown
Recording Device	Unknown
TRANSCRIPTION INPUTS	
Reference	Segmented on speaker utterance level (one .txt file per speaker utterance)
Audio	Segmented on speaker utterance level (one .wav file per speaker utterance)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
TESTSET	
Test Set Definition	Default test set
Test Set Duration	1.4 hours
Number Utterances	1680
Number Speakers	168
Average Utterance Duration	3.09 seconds
Average Speaking Rate	172 wpm

A.9 VoxForge

	GENERAL INFORMATION
Summary	The volunteer-supported speech-gathering effort Voxforge3, on which the acoustic models we used for alignment were trained, contains a certain amount of LibriVox audio, but the dataset is much smaller than the one we present here, with around 100 hours of English speech, and suffers from major gender and per- speaker duration imbalances.
URL	http://www.voxforge.org
Owner / Authors	VoxForge
License	GNU General Public License: http://www.gnu.org/copyleft/gpl.html
	COPRUS PROPERTIES
Speaking Style	Monologue read-aloud speech
Accented Speech	Unknown
Dialectal Variation	Yes
Overlapping Speech	No
Filled Pauses	No
Speaker Noise	No
Acoustic Environment	Unknown
Recording Device	Unknown
	TRANSCRIPTION INPUTS
Reference	Segmented on speaker utterance level (one .txt with all utterances)
Audio	Segmented on speaker utterance level (.flac/.wav file per utterance)
Applied Segmentation (ref and audio)	Segments on speaker utterance level
	TESTSET
Test Set Definition	Default test set
Test Set Duration	3.9 hours
Number Utterances	2929
Number Speakers	171
Average Utterance Duration	4.78 seconds
Average Speaking Rate	171 wpm

B Utterance Data Structure

identifier		
speaker_id		
reference		
	text	
	original_reference	
	only_non_lexical_sounds	
	num_non_lexical_sounds_in_reference_text	
audio		
	file	
	original_audio_file_path	
	duration	
	samplerate	
	bitdepth	
	channels	
	extra	
		encoding
		num_samples
recording_setup		
	recording_device	
	acoustic_environment	
dialect		
accent		
gender		
overlappings		
speaker_noise_utterance		
speaking_rate		
extra		
original_audio		
	file	
	duration	
	samplerate	
	bitdepth	
	channels	
	extra	
		encoding
		num_samples
hypothesis		
	text	
scoring		
	alignment	
	mutations	
	wer	

C Dialects List of English Language

en-AU	English, Australia
en-BZ	English, Belize
en-CA	English, Canada
en-GB	English, United Kingdom
en-ID	English, Indonesia
en-HK	English, Hong Kong
en-IE	English, Ireland
en-IN	English, India
en-JM	English, Jamaica
en-MY	English, Malaysia
en-NZ	English, New Zealand
en-PH	English, Philippines
en-SG	English, Singapore
en-ZA	English, South Africa
en-TT	English, Trinidad and Tobago
en-US	English, United States
en-ZW	English, Zimbabwe

D Corpora Variants

Corpus	**Segmented Variant**	**Unsegmented Variant**	**Other Variants**
AMI	x	-	Depending on recording device: headset, headset_mix
Common Voice	x	-	-
LibriSpeech	x	-	Depending on difficulty level: clean, other
RT	x	-	Depending on recording device: headset, headset_sum
ST	x	-	-
Switchboard	x	-	-
Tedlium	x	x	-
Voxforge	x	-	-
Timit	x	-	-