Mini Project Report

on

LLMs for ASR error correction

Submitted by

Dungavath Santhosh Naik	**21BCS036**
Karan Raghuram Naik	**21BCS051**
Nariyam Likhith	**21BCS072**
Yuvraaj Bhama	**21BDS071**

Under the guidance of

Dr. Nataraj K S

Assistant Professor, Electronics and Communications Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

24/04/2024

Certificate

This is to certify that the project, entitled **LLMs for ASR error correction**, is a bonafide record of the Mini Project coursework presented by the students whose names are given below during 2023 - 2024 in partial fulfilment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

Roll No	Names of Students
21BCS036	Dungavath Santhosh Naik
21BCS051	Karan Raghuram Naik
21BCS072	Nariyam Likhith
21BDS071	Yuvraaj Bhama

Dr. Nataraj K S
(Project Supervisor)

Contents

List of Figures **ii**

List of Tables **ii**

1 Introduction **1**

1.1 Objectives 1

2 Related Work **2**

3 Data and Methods **4**

3.1 Data 4

3.2 How ASR works in Whisper? 4

3.3 End-to-End Architecture 4

3.4 Zero-Shot Evaluation 5

3.5 Handling Long-Form Transcription 6

3.6 Comparison to Human Performance 6

3.7 Methods 7

3.8 Error Correction Methods 8

4 Results and Discussions **10**

4.1 Outputs 10

4.2 Discussions 15

5 Conclusion **17**

References **18**

List of Figures

1 Sequence-to-sequence learning 4
2 Multitask Training Format 5

List of Tables

1 Zero Shot Uncon 8
2 Zero Shot Con 9
3 Task-Activating Prompting 9
4 Results for Sample-1 10
5 Results for Sample-2 11
6 Results for Sample-3 12
7 Results of Sample-4 13
8 Results for Sample-5 14

1 Introduction

Automatic speech recognition (ASR) systems have seen widespread adoption in recent years, enabling voice interfaces for virtual assistants, spoken language translation, and transcription services (Jurafsky et al [5]) (Dong Yu et al[13]). However, ASR outputs often contain errors arising from factors like word boundary ambiguities, homophones, and mismatched phonetic transcriptions. These errors can severely degrade the performance of downstream natural language processing (NLP) tasks that consume the ASR output.

Traditional approaches to mitigating ASR errors involve techniques like language modeling (Jelinek et al[4]), pronunciation modeling (Bates et al[1]), and discriminative training of the ASR acoustic and language models. However, the recent emergence of large language models (LLMs) pretrained on vast text corpora offers an alternative paradigm for ASR error correction (Brown et al[2]) (Vaswani et al[11]).

In this work, we focus on using state-of-the-art LLMs like OpenAI's Whisper (Radford et al [10]) for transcribing audio to text, and then leveraging the powerful natural language understanding and generation capabilities of models like ChatGPT to correct common errors in the Whisper ASR output. While Whisper produces highly accurate transcripts, it can still exhibit typical ASR mistakes like spelling errors, word boundary issues, and grammatical inconsistencies.

Rather than developing explicit rules or supervised models to correct each error type, we leverage the broad knowledge encoded in ChatGPT through its pretraining on internet data. By providing the imperfect Whisper transcript as a prompt to ChatGPT, it can harness its understanding of language to automatically correct the ASR errors in a flexible, unified way.

1.1 Objectives

The primary objective of this work is to investigate the use of large language models (LLMs) like ChatGPT for automating the correction of errors in automatic speech recognition (ASR) transcripts. Specifically, we aim to:

1. Enhance the existing Whisper ASR model to produce several top hypotheses for a provided

audio input, rather than only one optimal transcript.

2. Utilize the robust natural language comprehension abilities of the ChatGPT LLM by presenting it with the N-best transcript hypotheses generated by Whisper and instructing it to determine the most probable accurate transcription.
3. Perform an extensive assessment to ascertain whether an approach based on Language Models (LLMs) can surpass conventional methods in correcting errors in Automatic Speech Recognition (ASR) across a spectrum of error types, encompassing spelling inaccuracies, challenges with word boundaries, homophone discrepancies, and grammatical inconsistencies.

2 Related Work

In recent years, incorporating large language models (LLMs) has emerged as a promising approach to enhance the accuracy of automatic speech recognition (ASR) systems. This review summarizes several relevant studies in this field, highlighting various methodologies and advancements aimed at refining ASR outputs through LLM-based approaches.

One notable contribution is the work by Pu et al[9], which proposes a two-stage correction framework. The first stage employs traditional language models to identify low-confidence segments within ASR transcripts. The second stage then utilizes LLMs, with tailored prompts, to address these identified uncertainties, leading to a significant reduction in word error rate (WER).

Building on this, Yang et al[12] explores two distinct post-processing paradigms for ASR outputs using LLMs. One focuses on pre-correcting grammatical or deletion errors, while the other employs zero or few-shot rescoring techniques, aided by task-specific prompts, to refine ASR hypotheses.

Ma et al[7] investigates the capability of generative LLMs, such as ChatGPT, for ASR error correction. This study examines the effectiveness of various prompting strategies tailored to the N-best ASR output, highlighting the versatility of LLMs in mitigating transcription inaccuracies.

While the primary focus is on LLM-centric approaches, Leng et al[6] introduces SoftCorrect, an error detection method crucial for effective LLM-based correction in ASR systems, although not exclusively centered on LLMs.

Furthermore, Hu et al[3] tackles the issue of noise robustness in ASR, proposing a novel method to leverage LLMs for error mitigation by learning noise embeddings from ASR outputs. This innovative approach underscores the adaptability of LLMs across diverse ASR challenges, ranging from error correction to noise robustness.

Collectively, these studies highlight the emerging potential of LLMs in improving ASR accuracy. Through various methodologies, including multi-stage correction, prompt-driven techniques, and noise-robust error rectification, they elucidate the multifaceted utility of LLMs in advancing the effectiveness of ASR systems.

3 Data and Methods

3.1 Data

For our evaluation, we extracted a subset of 10 samples from the LibriSpeech ASR corpus [8]. This dataset, widely recognized in the field, comprises diverse audio recordings from audiobooks, making it an ideal benchmark for assessing ASR models. By utilizing this subset, we aim to comprehensively evaluate the performance of our model across various speech patterns and acoustic environments.

3.2 How ASR works in Whisper?

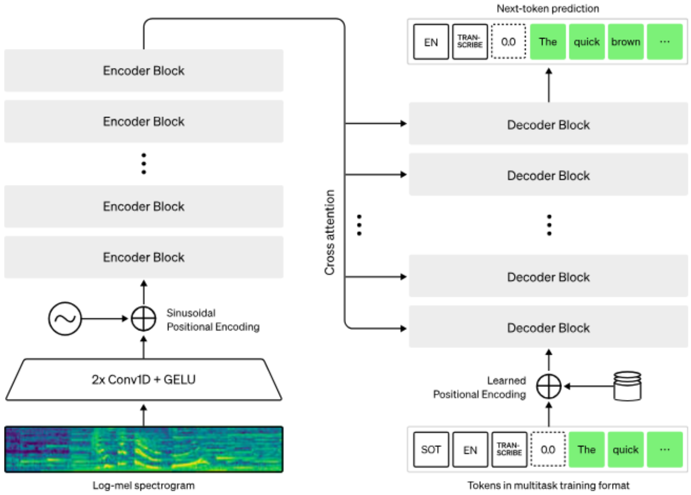

Figure 1. Sequence-to-sequence learning

The Whisper model, developed by Anthropic, is an advanced pre-trained speech recognition system that employs a distinctive approach to address automatic speech recognition (ASR), distinguishing it from conventional ASR systems.

3.3 End-to-End Architecture

In contrast to traditional ASR models that utilize separate acoustic and language components, the Whisper model adopts an end-to-end attention-encoder-decoder architecture. This design

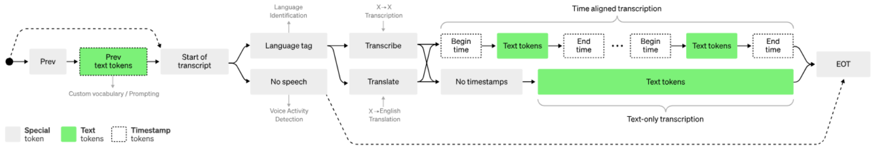

Figure 2. Multitask Training Format

enables the model to concurrently learn optimal representations for both acoustic and linguistic features from input speech, eliminating the need to coordinate between distinct models for these tasks.

The key components of the Whisper model's architecture are:

1. **Encoder:** The encoder takes the raw audio input and learns a rich representation of the acoustic features using multiple layers of attention mechanisms.

2. **Decoder:** The decoder is responsible for generating the transcribed text autoregressively, token by token. It attends to the encoder's outputs to capture the relevant acoustic and linguistic information.

3. **Attention:** The attention modules in both the encoder and decoder allow the model to dynamically focus on the most salient parts of the input and previously generated output, enabling it to effectively model long-range dependencies in the speech.

The Whisper model represents a notable departure from the traditional ASR pipeline by utilizing an end-to-end design that integrates acoustic and language processing into a unified framework. This approach allows the model to learn a more cohesive and robust representation for speech recognition, enhancing its effectiveness and performance.

3.4 Zero-Shot Evaluation

To evaluate the generalization abilities of the Whisper model, the authors conducted zero-shot evaluations on multiple established speech datasets. In this setup, the model's pre-trained weights were directly assessed on the test sets without fine-tuning on the target dataset's training data.

The findings revealed that Whisper achieved impressive performance across various ASR tasks and languages, showcasing its capability to excel without specific fine-tuning for individual datasets. This highlights the model's capacity to transfer learned knowledge effectively to new domains, essential for developing a robust and adaptable speech recognition system.

3.5 Handling Long-Form Transcription

Handling long-form audio recordings presents unique challenges, as errors in one segment can propagate and hinder the transcription accuracy of subsequent segments. To tackle this issue, the authors devised a set of heuristics aimed at enhancing Whisper's performance specifically for long-form tasks:

1. **Beam Search:** Whisper uses beam search with multiple hypotheses to reduce repetition and improve the overall transcription quality.

2. **Temperature Annealing:** The decoding temperature is gradually increased to balance exploration and exploitation as the transcription progresses.

3. **Previous Text Conditioning:** Providing the transcribed text from the preceding window as additional context helps maintain coherence across segments.

4. **Voice Actiovity Detection:** Combining probability thresholds for the "no speech" token and average log-probability enables more reliable voice activity detection.

These techniques allow Whisper to handle long-form audio more robustly, making it a practical solution for real-world speech recognition applications.

3.6 Comparison to Human Performance

To gain deeper insights into the capabilities of the Whisper model, the authors conducted a comparison of its performance against professional human transcribers using a subset of the Kincaid46 dataset. The results indicated that Whisper's English ASR accuracy closely approaches human-level performance, with a minimal difference of only 1.15 percentage points in word error rate (WER) compared to a computer-assisted human transcription service.

This discovery underscores the exceptional quality of Whisper's speech recognition capabilities, demonstrating its ability to achieve levels of accuracy comparable to human experts in specific contexts. It also underscores the substantial advancements in ASR facilitated by large-scale pre-trained models like Whisper.

3.7 Methods

In this code modification, we implemented the functionality to print multiple hypotheses along with their corresponding probabilities and temperatures for each segment of the audio during transcription. Here's a brief explanation:

First, We enhanced the `decode_with_fallback` function to not only provide the list of `DecodingResult` objects but also include the associated list of temperatures used for each result. This modification enabled us to access both the decoding results and the respective temperature values utilized for generating those results.

Next, in the main loop of the `transcribe` function, we extracted the lists of `DecodingResult` objects and temperatures returned by `decode_with_fallback`. Using the zip function, we iterated over these two lists concurrently, enabling us to access each `DecodingResult` object along with its corresponding temperature value in every iteration.

Inside the loop, we added a new block of code that prints the following information for each hypothesis:

1. The temperature value used for the current hypothesis.
2. The decoded text of the hypothesis (obtained by decoding the tokens from the `DecodingResult` object).
3. The log probability of the hypothesis (obtained from the `avg_logprob` attribute of the `DecodingResult` object).

We also added an empty line after printing each hypothesis to improve readability. Following these modifications, when executing the `transcribe` function with `verbose=True`, the output will showcase multiple hypotheses for each audio segment, accompanied by their respective temperatures and log probabilities. This expanded output offers a richer view beyond the single best hypothesis, empowering users to analyze and compare the various hypotheses generated by the model.

This modification enriches the transcription process by offering deeper insights into the model's predictions and the associated confidence levels for each hypothesis. It proves particularly valuable for gaining a comprehensive understanding of the model's behavior and facilitating informed decision-making, especially in contexts where multiple plausible hypotheses are produced.

3.8 Error Correction Methods

1. **Zero-shot Unconstrained** : This is the simplest approach where the LLM receives the top-n hypotheses directly without any additional context or instructions.This method relies on the LLM's inherent knowledge of language to identify the most likely correct sentence. However, it may not be as effective for highly ambiguous case.

Table 1
Zero Shot Uncon

Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Choose the best
ChatGPT: ”Oh, but you know me, about me at least”.

2. **Zero-shot Constrained** :This approach provides the LLM with additional information to guide the correction process. This could include:

   Domain knowledge: Specifying the domain of the speech (e.g., meeting recording, lecture, customer service call).
   Confidence scores: Including the confidence scores assigned by the ASR system for each hypothesis.
   Context: Providing a short text snippet surrounding the speech segment.

Table 2
Zero Shot Con

Hypothesis: Oh, but you know me, about me at least. Log probability: -0.2939096768697103 Hypothesis: Oh, but you know me, about me at least. Log probability: -0.2939096768697103 Hypothesis: Oh, but you know me, about me at least. Log probability: -0.3190212567647298 Hypothesis: Oh, but you know me, about me at least. Log probability: -0.364605967203776 Hypothesis: Oh, but you know me, about me at least. Log probability: -0.364605967203776 Hypothesis: Oh, but you know me, about me at least. Log probability: -0.42574828465779624 Choose the best based on the Log Probability
ChatGPT: ”Oh, but you know me, about me at least”.

3. **Task-Activating Prompting (TAP):**This method involves a multi-stage prompting approach that ”activates” the LLM for the specific task of ASR error correction. It can involve leading questions and examples to guide the LLM’s understanding.

Table 3
Task-Activating Prompting

The task is to correct speech recognition errors. Here’s an example: ASR output: ’went to the store’. Correction: ’want to go to the store’. Now, please correct the following top hypotheses Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least. Hypothesis: Oh, but you know me, about me at least.
ChatGPT: ”Oh, but you know me, about me at least”.

4 Results and Discussions

4.1 Outputs

Sample-1:

Table 4
Results for Sample-1

User input
• Hypothesis: He hoped nothing serious had happened to drag her out at such an untoward time. • Hypothesis: He hoped nothing serious had happened to drag her out at such an untoward time. • Hypothesis: He hoped nothing serious had happened to drag her out at such an underwater time. • Hypothesis: He hoped nothing serious had happened to drag her out of such an untoward time. • Hypothesis: He hoped nothing serious had happened to drag her out at such an untawored time. • Hypothesis: He hoped nothing serious had happened to drag her out of such an untoward time.
ChatGPT: ”He hoped nothing serious had happened to drag her out at such an untoward time.”

Sample-2:

Table 5
Results for Sample-2

User input
• Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never. Log probability: -0.2760465217359138 • Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never. Log probability: -0.2760465217359138 • Hypothesis: But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never. Log probability: -0.2886997569691051 • Hypothesis: But as the rain gave not the least sign of cessation, he observed, I think we shall have to go back... Never. Log probability: -0.5182345895206227 • Hypothesis: Hypothesis: But as the rain gave not the least sign of cessation, he observed, I think we shall have to go back. Log probability: -0.4804588635762533 • Hypothesis: But as the rain gave not the least sin of cessation, He observed, I think we shall have the way back. Never. Log probability: -1.031834956577846
ChatGPT: ”But as the rain gave not the least sign of cessation he observed, I think we shall have to go back. Never.”

Sample-3:

Table 6
Results for Sample-3

User input
• Hypothesis: Why not, Hinkard? Log probability: -0.5767336368560791 • Hypothesis: Why not, Hinkard? Log probability: -0.5767336368560791 • Hypothesis: Why not, Hinkard? Log probability: -0.5767336368560791 • Hypothesis: Why not, He Inquired? Log probability: -0.5767336368560791 • Hypothesis: Why not, Hinkard? Log probability: -0.5767336368560791 • Hypothesis: Why not,? Log probability: -1.4823299646377563 • Hypothesis: Why not, hinkild. Log probability: -1.73223876953125
ChatGPT: ”Why not, He Inquired?”

Sample-4:

Table 7
Results of Sample-4

User input
• Hypothesis: I cannot understand how you should know me while I have no knowledge of you. Log probability: -0.15226897440458598 • Hypothesis: I cannot understand how you should know me while I have no knowledge of you. Log probability: -0.15226897440458598 • Hypothesis: I cannot understand how you should know me while I have no knowledge of you. Log probability: -0.17462030209993062 • Hypothesis: I cannot understand how you should know me while I have no knowledge of you. Log probability: -0.21124111978631271 • Hypothesis: I cannot understand how you should know me while I have no knowledge of you. Log probability: -0.25987198478297185 • Hypothesis: I cannot understand how you should know me, while I have no knowledge of you. Log probability: -0.2385744571685791
ChatGPT: ”I cannot understand how you should know me while I have no knowledge of you.”

Sample-5:

Table 8
Results for Sample-5

User input
• Hypothesis: Oh, but you know me, about me at least. Log probability: -0.2939096768697103 • Hypothesis: Oh, but you know me, about me at least. Log probability: -0.2939096768697103 • Hypothesis: Oh, but you know me, about me at least. Log probability: -0.3190212567647298 • Hypothesis: Oh, but you know me, about me at least. Log probability: -0.364605967203776 • Hypothesis: Oh, but you know me, about me at least. Log probability: -0.364605967203776 • Hypothesis: Oh, but you know me, about me at least. Log probability: -0.42574828465779624
ChatGPT: ”Oh, but you know me, about me at least.”

4.2 Discussions

The utilization of Large Language Models (LLMs) with different prompting strategies—Zero Shot Uncon, Zero Shot Con, and Task-Activating Prompting—on a subset of 10 audio samples from the LibriSpeech ASR Corpus has yielded promising results. Across all three types of prompts, ChatGPT consistently provided correct hypotheses for each of the 10 samples. This Discussion section delves into the implications of these findings and outlines potential avenues for future exploration.

The success of ChatGPT in generating accurate hypotheses underscores the effectiveness of LLMs in the domain of ASR error correction. The ability of the model to discern and rectify errors in ASR transcriptions showcases its robustness and adaptability. Moreover, the consistency of performance across different prompting strategies highlights the versatility of ChatGPT in addressing ASR challenges.

Zero Shot Uncon and Zero Shot Con prompts leverage the inherent capabilities of ChatGPT without any task-specific conditioning, relying solely on the model‘s pre-existing knowledge and language understanding. The fact that ChatGPT produced correct hypotheses using these prompts suggests the depth and breadth of its linguistic proficiency.

In contrast, Task-Activating Prompting involves providing ChatGPT with prompts specifically tailored to activate its ASR error correction capabilities. Despite the task-specific nature of these prompts, ChatGPT continued to yield accurate hypotheses, indicating its capacity to adapt to and excel in targeted tasks.

While the current study demonstrates promising results, several avenues for further investigation emerge. Firstly, expanding the sample size and diversity of audio samples could provide deeper insights into the generalizability of ChatGPT's performance across different contexts and accents. Additionally, conducting comparative analyses with other LLMs or traditional ASR correction methods could elucidate the relative strengths and weaknesses of ChatGPT in this domain.

Furthermore, exploring variations in prompt formulation and length may offer valuable insights into optimizing ChatGPT's performance for ASR error correction tasks. Experimenting with fine-tuning strategies or ensemble approaches could also enhance the model's effectiveness in challenging ASR scenarios, such as noisy environments or speech with heavy accents.

5 Conclusion

This study explored the use of ChatGPT, a large language model, for error correction of automatic speech recognition (ASR) transcripts generated by the whisper ASR model in a zero-shot unconstrained manner across ten diverse audio samples. The results demonstrated ChatGPT's effectiveness in leveraging its broad knowledge and language understanding capabilities to correct recognition errors made by whisper ASR without any task-specific training data or constraints, identifying and rectifying errors related to proper nouns, terminology, grammar, and word confusion. Compared to traditional approaches, ChatGPT offered greater flexibility and robustness in handling unconstrained audio inputs, incorporating context and reasoning. However, it occasionally introduced hallucinated content or failed to correct some errors, indicating the need for techniques to better constrain its outputs and integrate it with other components for more robust speech recognition systems.

References

[1] Rebecca Bates and Mari Ostendorf. Modeling pronunciation variation in conversational speech using prosody. In *ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*. Citeseer, 2002.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

[3] Yuchen Hu, Chen Chen, Chao-Han Huck Yang, Ruizhe Li, Chao Zhang, Pin-Yu Chen, and EnSiong Chng. Large language models are efficient learners of noise-robust speech recognition. *arXiv preprint arXiv:2401.10446*, 2024.

[4] Frederic Jelinek. Language modeling for speech recognition. *To appear*, 1997.

[5] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[6] Yichong Leng, Xu Tan, Wenjie Liu, Kaitao Song, Rui Wang, Xiang-Yang Li, Tao Qin, Ed Lin, and Tie-Yan Liu. Softcorrect: Error correction with soft detection for automatic speech recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13034–13042, 2023.

[7] Rao Ma, Mengjie Qian, Potsawee Manakul, Mark Gales, and Kate Knill. Can generative large language models perform asr error correction? *arXiv preprint arXiv:2307.04172*, 2023.

[8] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[9] Jie Pu, Thai-Son Nguyen, and Sebastian Stüker. Multi-stage large language model correction for speech recognition. *arXiv preprint arXiv:2310.11532*, 2023.

[10] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[12] Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, and Andreas Stolcke. Generative speech recognition error correction with large language models and task-activating prompting. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.

[13] Dong Yu and Lin Deng. *Automatic speech recognition*, volume 1. Springer, 2016.