Digits micro-model for accurate and secure transactions

Chirag Chhablani[1,*], **Nikhita Sharma**[2,†], **Jordan Hosier**[2], **and Vijay K. Gurbani**[2,3]

[1]Department of Computer Science, University of Illinois, Chicago
[2]Vail Systems, Inc.
[3]Department of Computer Science, Illinois Institute of Technology
cchhab2@uic.edu, nikhitasharma2606@gmail.com, jhosier@vailsys.com, vgurbani@{vailsys.com, iit.edu}

Abstract

Automatic Speech Recognition (ASR) systems are used in the financial domain to enhance the caller experience by enabling natural language understanding and facilitating efficient and intuitive interactions. The increasing use of ASR systems requires that such systems exhibit very low error rates. The predominant ASR models to collect numeric data are large, general-purpose models provided commercially — Google Speech-to-text (STT), or Amazon Transcribe — or available through open source (OpenAI's Whisper). Such ASR models are trained on hundreds of thousands of hours of audio data and require considerable resources to run. Despite recent progress in such large speech recognition models, in this work we highlight the potential of smaller, specialized *"micro"* models. Such light models can be trained perform well on number recognition specific tasks, competing with general models like Whisper or Google STT while using less 80 minutes of training time and occupying at least an order of less memory resources. Also, unlike larger speech recognition models, micro-models are trained on carefully selected and curated datasets, which makes them highly accurate, agile, and easy to retrain, while using low compute resources. We present our work on creating micro models for multi-digit number recognition that handles diverse speaking styles reflecting real-world pronunciation patterns. Our work contributes to domain-specific ASR models, improving digit recognition accuracy, and maintaining privacy of data. An added advantage of our micro-models is their low resource consumption allows them to be hosted on-premise, thereby keeping private data local instead of sending it to an external cloud for processing. Our results indicate that our micro-model makes less errors than the best-of-breed commercial or open-source ASRs in recognizing digits (1.8% error rate of our best micro-model versus 5.8% error rate of Whisper), and has a low memory footprint (0.66 GB VRAM for our model versus 11 GB VRAM for Whisper).

Introduction

Today, many financial transactions take place over the phone. In this setting, deploying accurate speech recognition systems is crucial for precise digit recognition. The accuracy with which machine learning models recognize multi-digit utterances plays a pivotal role in shaping the efficiency of various financial applications. For example, a scenario in which a voice-enabled transaction involves a series of digits that are mistranscribed due to the limitations of a domain-general model, may result in financial discrepancies or authentication failures. Figure 1, shows an example of text where the utterance ”twelve six” is transcribed incorrectly as ”well fix” because of large domain of the large ASR model whereas it is transcribed correctly because of small domain of micro ASR model. The consequences could range from minor errors to significant financial discrepancies and bad user experience. Moreover in this context of voice-enabled financial transactions, the importance of deploying models that go beyond mere efficiency to ensure precise digit recognition becomes paramount.

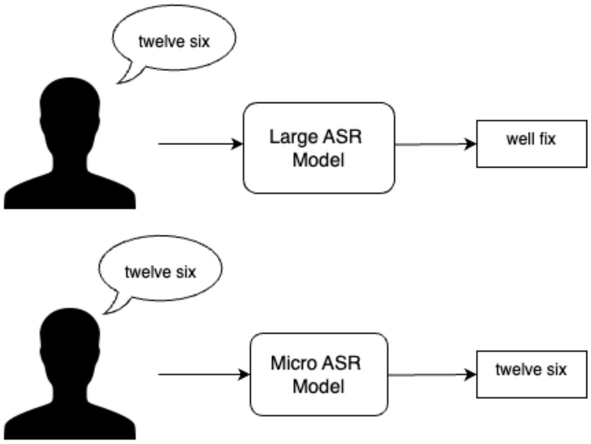

Figure 1: Figure shows the effect of large and micro language model on ambiguous inputs. Due to large vocabulary the large language model can lead to erroneous outputs whereas the micro models can generate an output which is within the range of provided vocabulary

The significance of accurate digit recognition extends beyond efficiency to include privacy considerations - particularly in financial domains that deal with credit numbers,

[*]Work was done as part of PhD summer internship at Vail Systems, Inc.

[†]Work was done while author was employed by Vail Systems, Inc.

CVV or account numbers. Accurate STT APIs like Google-speech APIs require sending the data to an external cloud that can have significant privacy concerns. This is where the imperative for socially responsible, privacy-aware micro-models comes into play. Consider a micro-model specifically tailored for financial contexts and adept at recognizing spoken digits with a heightened emphasis on privacy. Such a model not only ensures the accurate processing of financial transactions but also prioritizes the protection of personal financial information by avoiding an external API call. Further by addressing pronunciation variations and potential misinterpretations, this micro-model exemplifies the fusion of accuracy and efficiency. In this work, we focus on training a multi-digit micro-model that can handle diverse production of digits with high accuracy and computational efficiency.

There is a noticeable research gap in the open-source datasets available for training models on spoken digit recognition. Most prior datasets either focus solely on single-digit utterances or predominantly on multi-digit utterances where digits are spoken individually. Our work bridges this gap by including a wide range of sequences representing various digit lengths and variations in pronunciation. For example, "653" may be spoken as "six hundred fifty-three," "six fifty-three," or "sixty-five three." Specifically, the curated dataset is comprised of 14,000 utterances and aims to address the lack of diversity in the production of spoken digit data. This dataset is publicly available[1] and can be used by the research community to train speech recognition models to recognize credit card and account numbers - cases in which the data is not generally available due to its highly sensitive nature.

Morever, the utilization of external APIs may raise privacy concerns due to the potential training of the ASR model on user data[2] . While external providers often assert non-ownership of private data, the process of training their models relies on large datasets, including user inputs. This indirect exposure of sensitive information to external entities poses a risk to user privacy; published literature documents successful identification of individuals from data used to train models [Xu, Cohn, and Ohrimenko2023, Yin et al.2023, Fredrikson, Jha, and Ristenpart2015, Ravindra and Grama2021]. In contrast, the adoption of micro-models in an in-house capacity offers a more controlled and socially-responsible environment. With micro-models, organizations maintain ownership and oversight of the training process, reducing the likelihood of inadvertent data exposure. This approach not only enhances privacy but also provides greater transparency and control over how user data is handled, mitigating the potential risks associated with external API usage.

Our contributions can be summarized as follows:

1. We highlight the potential of micro-models over domain-general models for specialized use cases.
2. We create — and make publicly available at the URL shown in the footnote — a dataset for multi-digit recognition that can handle diverse articulation of digits for upto five-digit numbers.
3. We train two micro-models that outperform domain-general commercial ASR models like Google and Whisper [Radford et al.2023] in terms of recognition accuracy, and require a fraction of resources to train and deploy them.

The rest of the paper is structured as follows: We briefly summarize how ASR systems work in the next section. Related work and experiment details follow to outline the curation of our dataset and the techniques used for training the models. We discuss the results next followed by a conclusion.

ASR Background

The objective of an Automatic Speech Recognition (ASR) system is to convert an audio signal into its corresponding text, known as a *transcript*. The transcript is subsequently used by back-end systems to drive a specific application. As an example, consider an account holder calls into a financial institution, and the system challenges her by asking an account number. The ASR captures the acoustic signal corresponding to user's utterance as she speaks out the digits and converts, or decodes, it to a textual transcript consisting of the digits spoken during authentication. The transcript is then presented to a back end system that retrieves the account based on the digits contained in the transcript.

ASR systems typically include an acoustic model, a language model, and a decoder. This system processes speech input to produce the most accurate transcription of spoken words. In short, the feature extraction module isolates pertinent features from the speech signal, minimizing unnecessary noise. The acoustic model captures speech acoustics and transcribes the extracted audio features into a sequence of context-dependent phonemes, often using deep neural networks in contemporary models. The language model determines the likelihood of specific words or word sequences based on surrounding context. Finally, the decoder utilizes the acoustic model, grammar, and language model in tandem to generate probable word sequences for a given audio frame, with the highest probability word sequence being the final text output. A comprehensive exploration of deep neural network-based ASR systems is presented in the work of Roger et al. [Roger, Farinas, and Pinquier2022].

To delve further into the process, the input waveform undergoes segmentation into small frames, typically 25 milliseconds in duration, and from these frames, specific features are extracted. Commonly utilized features include Mel-Frequency Cepstral Coefficients (MFCCs), Cepstral Mean and Variance Normalization (CMVNs) representing audio content, or i-Vectors capturing speaker or utterance style. The selected features must effectively capture human speech characteristics while minimizing unwanted noise. This compression of the audio signal results in a sequence of fixed-length vectors through feature extraction. Subsequently, the acoustic model predicts the phoneme spoken in each audio frame. The acoustic model is tasked with mod-

[1] The dataset can be downloaded from https://github.com/chiraguic/SpokenMultiDigitVarietyDataset

[2] A representative example of a privacy statement is https://cloud.google.com/speech-to-text/docs/data-logging

eling speech acoustics and transcribing the extracted audio features into a sequence of context-dependent phonemes. Training acoustic models involves Deep Neural Networks (DNNs) processing extensive datasets, typically comprising thousands of hours of human-transcribed audio data.

Finally, in (ASR) systems, a lexicon or dictionary links each word to its phonetic representation. This mapping helps convert predicted phonemes into words and, ultimately, full sentences. The language model plays a crucial role in determining the likelihood of specific words or word sequences based on surrounding context. This context is typically generated by neural networks or n-gram models trained on extensive textual datasets. The decoder then utilizes the acoustic model, grammar, and language model together to generate word sequences for a given audio frame. The final text output is the word sequence with the highest probability.

Related Work

Several open-source datasets exist in the realm of speech processing for digit recognition, each having its own set of strengths and limitations. The *MNIST Speech Dataset* curates a dataset by embedding spoken digit recognition into the well-known MNIST format [mni]. However, a notable constraint of this dataset is that it is focused solely on single-digit numbers. It does not address the challenges posed by multi-digit sequences containing a great deal of variation in production style. For instance, numbers like “480” can be produced as “four hundred eighty” or “four hundred and eighty,” variations that are not accounted for in the dataset.

In contrast, the *Snips SLU Dataset*, *Fluent Speech Commands dataset*, and *SLURP dataset* focus on speech command recognition [Saade et al.2019, Lugosch et al.2019, Bastianelli et al.2020]. While these datasets contribute to real-world applications, they don’t encompass the nuances of recognizing numbers spoken in varied forms or numbers embedded within sentences. These limitations are shared by the *Google Speech Commands Dataset*, which excels in recognizing short spoken commands but falls short in adequately handling multi-digit variations and numbers spoken within larger contexts.

Furthermore, even when datasets cover multi-digit utterances, they lack variety. For example, the *Timers-and-such Dataset* offers an intriguing avenue for exploring temporal event classification and spoken timer detection [Lugosch et al.2021]. However, it exhibits limitations, such as fewer utterances for numbers from ten to nineteen (10-19), which are crucial for recognizing the complete spectrum of five-digit numbers. Additionally *Aurora Dataset* focuses on standalone multi-digit utterances [aur]. However, a key limitation of the Aurora dataset is that it records the digits spoken one at a time, potentially not addressing the variations in pronunciation seen in real-world scenarios.

Domain-specific datasets, like those detailed above, as well as domain-specific ASR models have garnered significant attention in recent research. For instance, [Jha2021], targets domain-specific models for cases where domain-specific data is available in different languages. [Bekal et al.2021] focuses on the accurate recognition of domain-specific words and named entities (i.e. addresses, names, etc.) More recently, [de Vos and Verberne2023] present experiments using a domain-specific language model for political speeches. [Dong et al.2023] propose a complex but effective speech recognition method based on a domain-specific language speech network (DSL-Net) and a confidence decision network (CD-Net). They test their results on various open-source medical domain datasets.

Unlike the prior work outlined in this section, we focus on creating a domain-specific micro-model for commonly occurring digit recognition tasks that can be used in any situation, for instance, the financial domain. While existing datasets and models trained on them contribute significantly to digit recognition endeavors, there remains a gap in handling standalone multi-digit utterances spoken in diverse ways. Addressing this gap is particularly useful to enhance caller experiences in financial transactions where accurate and flexible multi-digit recognition is vital. Our proposed dataset and the associated micro-model trained on it aim to tackle this challenge, providing improved accuracy and adaptability to variable pronunciations.

Experimental Details

Dataset

We curated the proposed dataset by harnessing data available from the Timers and Such [Lugosch et al.2021] and LibriSpeech datasets [Lugosch et al.2019]. The Timers and Such dataset is an open-source dataset of spoken English commands for common voice control use cases involving numbers. It has four intents, corresponding to four common offline voice assistant uses: SetTimer, SetAlarm, SimpleMath, and UnitConversion. The dataset is fairly small, with 2,151 non-synthetic utterances, but it is considered useful for experimentation. The LibriSpeech dataset is a corpus of approximately 1000 hours of read English speech with a sampling rate of 16 kHz. It is derived from read audiobooks from the LibriVox project and has been carefully segmented and aligned . Our objective was to meticulously extract spoken numbers from sentences present in these datasets, ultimately constructing a comprehensive repository for flexible multi-digit recognition. Extending prior work on single and double-digit numbers, we focus on numbers up to five digits in length. We compiled an exhaustive vocabulary comprising all tokens necessary for articulating five-digit numbers, encompassing numerical digits and number-related phrases. These tokens include single digits from zero to nine, double digits tokens from ten to nineteen, and tokens like “twenty”, “thirty,” etc. as shown in Table 1. In addition to digits, tokens representing key numerical expressions like ”hundred” and ”thousand” constitute the foundational building blocks for constructing complex numeric sequences. Further, we include the tokens “and”, “Oh”, and “O” as they are often used interchangeably with the digit zero in conversational speech.

Subsequently, we employed state-of-the-art Whisper models [Radford et al.2023] to glean precise timestamps of these tokens from the audio samples. The Whisper model delivered a detailed JSON file containing words and their corresponding timestamps (for e.g. ”words”: [”text”:

"eighty","start": 0.5, "end":0.9,"confidence": 0.90])[3]. We first identified the timestamps corresponding to the words of interest (for e.g. "four"-0:38-0:40s, "five"-0:40-0:42, etc). We then use the timestamp information for the each word to combine continuous sequences of digits (for e.g. "four five"-0:38-0:42) to assemble all multi-digit numbers articulated within the sentence. Finally, after extracting all relevant words from the audio, we append one second of silence at the beginning and end of the audio files. This step ensures that the individual audio files are of sufficient length for training.

List of Tokens in the Vocabulary		
One	Eleven	Twenty
Two	Twelve	Thirty
Three	Thirteen	Forty
Four	Fourteen	Fifty
Five	Fifteen	Sixty
Six	Sixteen	Seventy
Seven	Seventeen	Eighty
Eight	Eighteen	Ninety
Nine	Nineteen	Hundred
Oh	O	Ten
Thousand	And	Zero

Table 1: Vocabulary

The Kaldi toolkit [Povey et al.2011] was used for training the ASR model. Kaldi is a free, open-source toolkit for speech recognition research. It is considered best-of-breed open source ASR training toolkit and has been used as the foundation for many commercial ASR systems.

Train and Test Datasets To train the model we also included the Aurora dataset [aur] to ensure that the model can recognize sequential as well as non-sequential digits. Refer to the appendix about brief introduction on Aurora 5 dataset. We adopted a train-test ratio of 90:10 over the entire dataset. The dataset boasts a cumulative duration of about 4 hours. To standardize the audio data and ensure compatibility with phone audio recordings, all files were converted to an 8 kHz sample rate and encoded in 16-bit PCM format. The contribution from each dataset is shown in Table 2. The number of audio files and corresponding hours of data contributed by each of the datasets is specified.

Dataset	Train files (hrs)	Test files (hrs)
Aurora-5	7421 (2.30)	1355 (0.28)
Timers-and-such	2502 (0.61)	211 (0.10)
Librispeech	2721 (0.72)	230 (0.12)

Table 2: Dataset Information

Micro-model Architecture

We trained two neural network based digit models- *Micro-model-dense* and *Micro-model-light* with varied architectures. The dense model is more accurate but demonstrates slightly higher latency during decoding.

The input of the *Micro-model-dense* network comprises two components: an i-vector input with a dimensionality of 100 and a raw input with a dimensionality of 40. The network commences its transformation by applying an affine operation using an LDA matrix to the concatenated spliced frames of the i-vector, thereby reducing its dimensionality while retaining essential information. The subsequent stages involve a series of carefully orchestrated layers that progressively extract and manipulate features. A fully connected layer, characterized by ReLU activation, batch normalization, and dropout, introduces non-linearity and regularization of the data. Following this, a sequence of 13 time-delay neural network (TDNN) layers are introduced, leveraging factorized structures for computational efficiency and capturing temporal dependencies. These TDNN layers, each comprising 1,536 dimensions with a bottleneck dimension of 160 and varying time strides, enable the network to capture intricate temporal patterns in the input data. Continuing the transformation, a linear layer processes the features, and a pre-final layer undertakes non-linear transformations with dimension reduction, thereby molding the data to match the requirements of specific modeling tasks. Here, the architecture bifurcates into two distinct branches. One branch caters to chain modeling, involving another pre-final layer and an output layer. The pre-final layer reduces the dimensions from 1,536 to 256, while the output layer generates predictions without the application of log-softmax transformation. This configuration is tailored for tasks necessitating sequence labeling. Simultaneously, the second branch addresses cross-entropy training, encompassing analogous pre-final and output layers. These layers facilitate learning through cross-entropy loss, and the output layer produces classifications.

The **Micro-model-dense** neural network architecture, while achieving a lower Word Error Rate (WER[4]), necessitates a slightly higher decoding time and demands more space. In an effort to strike a better balance between accuracy, space efficiency, and decoding time, we opted for a simplification of the neural network, dubbing it **Micro-model-light**. A significant adjustment involved a meticulous reduction in the number of TDNNF layers from 13 to 4, effectively mitigating continuous and resource-intensive weight multiplication operations. Additionally, recognizing the potential impact of excessively high dimensions on the space and time intensity of the model, we deliberately chose to judiciously decrease the dimensionality of the network's internal representations from 1,536 to 1,024. This architectural streamlining resulted in reduction in decoding time, which is more noticeable for CPU based decoding. For GPU

[3] It was observed that JSON outputs often contain a slight offset in the start and end times for words, sometimes capturing a very small portion of the previous or next word. Nevertheless, it appears that this has minimal impact on the performance of the model.

[4] The WER is a widely accepted standard measure of ASR performance; it is expressed as a value between [0, 1.0] or as a percentage. Lower values are better. Please see the Appendix for more information on the WER.

based decoding, the difference is marginal. The simpler design leads to slight increase in the WER.

	Micro-model-light	**Micro-model-dense**
WER	2.6%	1.8%
Model size (GPU, VRAM)	0.64 GB	0.66 GB
Model size (CPU, RAM)	0.16 GB	0.20 GB
Latency (RTF) on CPU	0.00343	0.01432
Latency (RTF) on GPU	0.00061	0.00060

Table 3: Performance comparison of **Micro-model-light** and **Micro-model-dense** on a test set of 1,149 utterances of 3,646.3 seconds total duration.

Results and discussion

Table 3 presents the performance comparison between two micro-models. The models are tested on a dataset comprising of 1,149 utterances with a cumulative duration of 3,646 seconds. We assess the models for CPU and GPU based inference tasks. The CPU used for testing is an AMD 16-Core Processor 2.9 GHz CPU with 263 GB RAM. The GPU used is an NVIDIA A100 with 80GB VRAM. Micro-model-light demonstrates a Word Error Rate (WER) of 2.65% and a smaller model size, resulting in a faster decoding time and lower latency measured in terms of the Real-Time Factor (RTF[5]). By contrast, Micro-model-dense achieves a lower WER of 1.84% but has a larger model size, resulting slower decoding, specially on the CPU.

We also compare our models to three popular, pre-trained models: Whisper-Small, Whisper-Large, and Google-STT (Speech-to-text); Whisper is the best-of-breed of open source ASR engines while Google-STT is the best-of-breed for commercial ASR engines. The results are shown in Table 4. For the Google-STT, we specifically configured the context to transcribe only numeric content. This tailored configuration allowed us to assess the performance of the Google Speech API in accurately transcribing numerical sequences, providing insights into its suitability for applications involving voice-enabled transactions, numerical data processing, and similar scenarios. Whisper-Small is a 244 million parameter model that requires at least 2 GB VRAM, while Whisper-Large is a 1.5 billion parameter model requiring about 11 GB VRAM. As can be observed in the table, the WER of our micro models are superior to those of the commercial engine (Google-STT) and the open source alternative (Whisper). In terms of RTF, we note that the RTF of our micro-models shown in the table are at least three orders of magnitude less than the RTF of Whisper, which we measured at 0.84 for Whisper-Small and 0.89 for Whisper-Large. We were not able to measure the RTF for Google-STT because it is a cloud service, using a RESTful API interface, and in such a setting, the network latency will dominate the RTF.

Model	**WER (%)**	**RTF (measured on GPU)**
Whisper-Small	5.8	0.84
Whisper-Large	4.6	0.89
Google-STT	2.9	N/A
Micro-model-light	**2.6**	**0.00061**
Micro-model-dense	**1.8**	**0.00060**

Table 4: Word Error Rates

Both of our micro models outperform the commercial engines. This indicates that small vocabulary micro models have comparable, or better accuracy compared to large vocabulary domain-general models. It is noteworthy that commercial engines are trained on hundreds of thousands of hours of audio and require sizable resources to perform decoding. Whisper is a good example, it is trained on 680,000 hours of audio [Radford et al.2023] over a period of many days on a cluster of NVIDIA A100 GPUs. The large version of Whisper occupies nearly 11 GB VRAM on a GPU compared to the 0.66 GB of VRAM for our micro-models. By contrast, our micro models require low memory resources to run, many orders of magnitude less training data and take less training time. (Our model training time was about 1 hour on 1xA100 NVIDIA GPU.) An added advantage of a low-memory footprint is multiple models can be loaded simultaneously in memory and an application can choose the appropriate model to use for decoding. For example, a credit card CVV consists of three numbers requiring less variation in pronouncing the three numbers. Most users when prompted to say the CVV will utter each number independently and in sequence. Therefore, a 3-digit CVV model that is highly trained to recognize individual numbers spoken in sequence can be used to collect the CVV while a different model can be used to collect the ZIP code where there is more variability in uttering the five numbers corresponding to the ZIP code.

To better understand the difference in WER between each model, we conducted a word-error analysis summarized in Table 5. We found that Google, at times, fails to capture single-digit utterances and decodes the short utterances as blank outputs. However, we found that the Google model performs well on multidigit utterances. Whisper-Small and Whisper-Large also exhibited frequent errors on single- and two-digit utterances, with these errors becoming less frequent for multi-digit numbers. Making more errors on short duration utterances is a known deficiency of Whisper models, since it is trained on longer audio segments (30s) making decoding shorter utterances challenging as the audio may not have enough context [Radford et al.2023]. Clearly, our micro-models do not suffer from this problem and are highly accurate for short audio utterances.

In closing, we note that our micro-models are not general-purpose models; they are highly specific to a particular domain, namely digit recognition. Thus, using them in any

[5] The Real-Time Factor (RTF) metric used to measure the speed of a system that processes an input signal (such as audio) in real time. Values under 1.0 are preferred. Please see the Appendix for more information on RTF.

other domain will cause high WER.

Conclusion

In this work, we present a domain-specific Kaldi ASR micro-model for accurate five-digit number recognition, which can outperform large, domain-general ASR models. We also present a diverse dataset for spoken digit recognition which adds to the existing resources in this domain, especially for high-privacy use cases like financial transactions where credit card or account numbers cannot be used directly for training. We trained micro-models that give low WER and highlighted the tradeoffs between the density of network, space occupied by the model and time for decoding the utterances. Both of our micro-models outperformed the best-of-breed commercial and open-source ASR systems while using less compute resources. Due to their inherent agility and low compute requirements, the micro-models can be run on-premises, thereby alleviating the need for sending sensitive data to a third party for decoding and transcription. These efforts underline the utility of micro-models in specialized and lightweight ASR applications, offering efficient and accurate solutions for specific tasks.

Future work will involve extracting digits from other data sources and improving the model's performance on out-of-distribution audio sequences. Moreover, our approach can also be applied to noisy datasets. Recognizing the significance of channel characteristics in real-world scenarios, our methodology lays the groundwork for future investigations into incorporating training audio that closely mimics the production environment. This avenue of exploration holds the potential to further enhance the robustness of our model, making it well-suited for applications in diverse and challenging acoustic conditions.

References

[aur] Aurora dataset. http://aurora.hsnr.de/download.html.

[Bastianelli et al.2020] Bastianelli, E.; Vanzo, A.; Swietojanski, P.; and Rieser, V. 2020. Slurp: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Bekal et al.2021] Bekal, D.; Shenoy, A.; Sunkara, M.; Bodapati, S.; and Kirchhoff, K. 2021. Remember the context! asr slot error correction through memorization. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 236–243. IEEE.

[de Vos and Verberne2023] de Vos, H., and Verberne, S. 2023. Political corpus creation through automatic speech recognition on eu debates. *arXiv preprint arXiv:2304.08137*.

[Dong et al.2023] Dong, Z.; Ding, Q.; Zhai, W.; and Zhou, M. 2023. A speech recognition method based on domain-specific datasets and confidence decision networks. *Sensors* 23(13):6036.

[Fredrikson, Jha, and Ristenpart2015] Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, 1322–1333. New York, NY, USA: Association for Computing Machinery.

[Jha2021] Jha, S. 2021. Learning domain specific language models for automatic speech recognition through machine translation. *arXiv preprint arXiv:2110.10261*.

[Lugosch et al.2019] Lugosch, L.; Ravanelli, M.; Ignoto, P.; Tomar, V. S.; and Bengio, Y. 2019. Speech model pre-training for end-to-end spoken language understanding. In *Interspeech*.

[Lugosch et al.2021] Lugosch, L.; Papreja, P.; Ravanelli, M.; Heba, A.; and Parcollet, T. 2021. Timers and such: A practical benchmark for spoken language understanding with numbers. *arXiv preprint arXiv:2104.01604*.

[mni] Mnist speech dataset. https://www.tensorflow.org/datasets/catalog/mnist_speech.

[Povey et al.2011] Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society.

[Radford et al.2023] Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, 28492–28518. PMLR.

[Ravindra and Grama2021] Ravindra, V., and Grama, A. 2021. De-anonymization attacks on neuroimaging datasets. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, 2394–2398. New York, NY, USA: Association for Computing Machinery.

[Roger, Farinas, and Pinquier2022] Roger, V.; Farinas, J.; and Pinquier, J. 2022. Deep neural networks for automatic speech processing: a survey from large corpora to limited data. *EURASIP Journal on Audio, Speech, and Music Processing* 2022(1):19.

[Saade et al.2019] Saade, A.; Coucke, A.; Caulier, A.; Dureau, J.; Ball, A.; Bluche, T.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; et al. 2019. Spoken language understanding on the edge. *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

[Xu, Cohn, and Ohrimenko2023] Xu, Q.; Cohn, T.; and Ohrimenko, O. 2023. *Fingerprint Attack: Client De-Anonymization in Federated Learning*.

[Yin et al.2023] Yin, H.; Liu, Y.; Li, Y.; Guo, Z.; and Wang, Y. 2023. Defeating deep learning based de-anonymization attacks with adversarial example. *Journal of Network and Computer Applications* 220:103733.

[Zechner and Waibel2000] Zechner, K., and Waibel, A. 2000. Minimizing word error rate in textual summaries of spoken language. In *1st Meeting of the North American*

Error Type	**Google STT**	**Whisper-Small/Large**	**Micro-model dense/light**
Single-digit (0-9) transcription and detection errors	Frequent blank outputs and occasional errors (e.g. ”Two” as ”too”)	Detection with more frequent errors (e.g. ”three” as ”spree”, ”nine” as ”fine”)	**Occasional blank outputs**
Two-digit (11-99) transcription errors	Occasional errors (e.g. ”Twelve” as ”well”)	More frequent errors (e.g. ”sixteen” as ”sixty”)	**Highly accurate**
Multi-digit numbers>99 errors	Highly accurate	Occasional errors (e.g. ”hundred” as ”dread”, ”thousand” as ”housing”)	**Highly accurate except rarely missing short words like ”and”, ”two”etc.**

Table 5: Error analysis of Automatic Speech Recognition systems: Google STT exhibits single-digit errors, Whisper-Small/Large shows increased errors for two-digit numbers, and micro-models demonstrate high accuracy with occasional multi-digit errors, particularly in recognizing short words

Chapter of the Association for Computational Linguistics.

Appendix

Word Error Rate (WER)

The WER [Zechner and Waibel2000] is a widely accepted standard measure of ASR performance; it is expressed as a value between [0, 1.0] or as a percentage. ASR systems seek to minimize the WER. It is represented as the ratio of the number of edits required to transform a hypothesis string into a reference string to the total number of words in the reference string, or

$$\text{WER} = \frac{S + D + I}{N} \tag{1}$$

where S = number of substitutions required to change the hypothesis string to the reference string, D = number of deletions required, I = number of insertions, and N = total number of words in the reference string. Lower values of WER are preferred since they indicate an ASR model that makes less errors.

Real-Time Factor (RTF)

The Real-Time Factor (RTF) metric used to measure the speed of a system that processes an input signal (such as audio) in real time. Since ASR systems process speech and produce a transcript, the RTF can be defined as

$$\text{RTF} = \frac{T}{D} \tag{2}$$

where T = Time to transcribe the audio file and D = Duration of the audio file. Values of RTF < 1.0 are preferred since values ≥ 1.0 indicate that the decoding (transcribing) an audio file takes a larger amount of time than the duration of the audio itself.

Aurora-5

Aurora-5, introduced in 2006, extends the Aurora dataset series, focusing on hands-free speech input in real-world settings like rooms and cars. The dataset investigates additional distortion effects, including additive background noise and the impact of transmitting speech over a cellular network. Built upon the TIDigits database, Aurora-5 offers a valuable resource for assessing speech recognition system performance in complex, everyday scenarios, contributing to the refinement of speech processing technologies in telecommunications. Its exploration of hands-free input and cellular transmission adds a practical dimension to evaluating automatic speech recognition systems.