Universität Zürich[UZH]

Seminar
NLP for Finance
Herbstsemester 2024

Speech Recognition in Central Bank Livestreams

Verfasser: Artem Shkabruk
Matrikel-Nr: 19-924-810

Dozent: Martin Volk
Institut für Computerlinguistik

Abgabedatum: 19.01.2025

1 Introduction

Central banks have a pivotal role in shaping the global economy through their monetary policies and economic strategies. Among these, the Federal Reserve (Fed) stands out as one of the most influential central banks, driving significant changes not only in the United States but across international financial markets. Central bank speeches, including those from the Federal Reserve, are vital channels for communicating policy updates and economic outlooks. These speeches often carry key indicators for interest rate changes, inflation expectations, and other macroeconomic signals [Bernanke, 2007]. Market participants, from institutional investors to individual traders, carefully analyze these speeches to anticipate market movements and adjust their strategies accordingly.

In foreign exchange (forex) markets, central bank speeches have an even greater impact. Forex markets operate on high liquidity and fast reaction times, making them especially sensitive to changes in sentiment and policy signals conveyed during these speeches. For instance, when a central bank hints at tightening monetary policy, it can strengthen the national currency, leading to immediate fluctuations in currency pairs. On the other hand, dovish statements indicating low interest rates might cause depreciation. Such movements create opportunities for profit in forex trading but demand rapid access to accurate information [Fratzscher, 2012].

Accurate transcription of central bank speeches is not only crucial for understanding policy implications but also holds great potential for enhancing profitability in financial markets. Forex traders and analysts, for instance, can utilize these transcripts for more effective sentiment analysis, keyword tracking, and predictive modeling based on central bank statements. Real-time processing systems integrated with such tools allow users to gain actionable insights faster than competitors, creating a tangible advantage [Hansen and McMahon, 2018].

The ultimate goal of this paper is to fine-tune a transcription model to better handle financial jargon specific to central bank speeches. Tools like Whisper can be leveraged for this purpose, offering advanced automatic speech recognition (ASR) capabilities. By incorporating forced alignment techniques, the transcribed text can be precisely matched to the corresponding audio, ensuring high reliability. This process enables the creation of a well-structured dataset tailored for training Whisper on financial data, including specific terms and phrases commonly used in forex and central bank communications. Such fine-tuning significantly improves transcription accuracy, addressing the unique challenges posed by domain-specific language.

1.1 Inflation

Inflation is a continuous increase in the general price levels of goods and services within an economy over time. This rise in prices reduces the purchasing power of money, meaning that over time, each unit of currency buys less. The impact of inflation on currency value is both complex and significant. It influences consumer spending, investment choices, and international trade competitiveness. Central banks pay close attention to inflation levels and use tools like interest rates to regulate inflation and aim for price stability. In forex markets, high inflation often results in a depreciation of the currency as investors shift their preference to currencies that retain more stable purchasing power. On the other hand, low and steady inflation can lead to a stronger currency. But the link between inflation and currency value is not always so direct, as other economic conditions and market sentiments can also influence this dynamic. For example, if high inflation happens alongside robust economic growth and rising interest rates, the currency could still appreciate due to higher investor confidence. To counteract extreme levels

of inflation, central banks adjust interest rates to impact the "cost of money." A higher interest rate makes borrowing more expensive, encouraging individuals and businesses to hold onto their money rather than spend or invest it. In contrast, lower interest rates incentivize borrowing, stimulating activities like starting businesses or expanding operations, which can boost the economy. This interest rate adjustment is a key policy instrument for managing inflation, particularly for the Federal Reserve, which has a significant influence on the forex market [Ahrens et al., 2024].

For this seminar paper, I will focus on the Federal Reserve's regular and open communication through its speeches, which reflect its influential role in global financial systems. As one of the most powerful central banks, the Fed's announcements and policy statements are closely followed across financial sectors, providing a rich source of information. By constructing this dataset, I aim to provide a resource that captures the essence of the Fed's influence through its rhetoric and offers valuable material for further fine-tuning speech recognition models like Whisper.

2 Related Work

The development and improvement of speech recognition and forced alignment systems have been extensively studied in recent years. This section provides an overview of foundational and state-of-the-art (SotA) approaches, focusing on forced aligners and advanced automatic speech recognition models relevant to this paper.

[Povey et al., 2011] introduces Kaldi, a free and open-source toolkit for speech recognition research. Kaldi provides a finite-state transducer-based recognition system (using OpenFst), comprehensive documentation, and scripts for building complete ASR systems. Kaldi supports arbitrary phonetic-context sizes, acoustic modeling with subspace Gaussian mixture models, standard Gaussian mixture models, and a variety of linear and affine transforms. Released under the Apache License v2.0, Kaldi is widely accessible and used across the speech research community.

Kaldi plays a crucial role as the foundation for the Montreal Forced Aligner (MFA) [McAuliffe et al., 2017], a widely used open-source tool for speech-text alignment. MFA extends Kaldi's functionality by maintaining trainability on new data and incorporating advanced features such as triphone acoustic models and speaker adaptation. Leveraging Kaldi allows MFA to function as a stand-alone package and utilize parallel processing, making it scalable for larger datasets. Evaluations have shown MFA to outperform simpler aligners, such as Prosodylab-Aligner [Gorman et al., 2011] and FAVE, by delivering more precise word and phone boundary alignments through its enhanced architecture and training features.

To highlight MFA's recent dominance in forced alignment tasks, [Rousso et al., 2023] compares it against WhisperX and Massively Multilingual Speech (MMS) [Pratap et al., 2023], demonstrating MFA's consistent superiority. Additionally, [Mahr et al., 2021] reinforces MFA's SotA performance by evaluating it alongside Kaldi and earlier tools, such as Prosodylab-Aligner and the Penn Phonetics Lab Forced Aligner [Yuan and Liberman, 2008], further validating MFA's advancements in the field.

As for the ASR component of this paper, I drew inspiration from [Radford et al., 2022], which introduces Whisper, a cutting-edge speech processing system. Whisper is trained on vast amounts of transcripts sourced from internet audio and achieves impressive results without relying on self-supervision or self-training techniques. Whisper demonstrates SotA zero-shot performance on datasets like CoVoST2 across overall, medium, and low-resource language settings. However, it moderately underperforms on

high-resource languages compared to prior directly supervised models. Whisper has certain limitations, particularly when dealing with audio longer than 30 seconds, as this is the maximum input length it can process. For longer audio, Whisper deploys a forced aligner to generate timestamps and chunk the audio into smaller segments. This chunking process, while effective, contributes to slower transcription speeds. For example, transcribing a 50-minute audio file with Whisper can take up to 15 minutes, as results in this paper demonstrate. Despite this, Whisper remains widely regarded as the current SotA in ASR.

However, Whisper's functionality has been expanded upon by models such as WhisperX [Bain et al., 2023], which addresses key limitations. One significant drawback of Whisper is its sequential transcription process, which prohibits batched inference and limits processing speed. WhisperX overcomes this by introducing word-level timestamps that combine voice activity detection (VAD) and forced phoneme alignment. This approach not only improves the accuracy of long-form audio transcription but also achieves SotA performance in handling extended audio recordings. Additionally, WhisperX proposes the VAD Cut & Merge strategy to pre-segment audio, which enhances transcription quality and enables up to a twelvefold speedup through batched inference. By resolving Whisper's buffering limitations and leveraging more efficient alignment techniques, WhisperX demonstrates significant improvements.

3 Introduction to Forced Alignment

Forced alignment (FA) is a technique designed to align an orthographic transcription of an audio file with its corresponding acoustic signal, creating a time-aligned version. By employing a pronunciation dictionary to map words to their phonetic representations, FA determines the temporal boundaries of speech units such as words or phonemes, streamlining the alignment process [McAuliffe et al., 2017].. The importance of forced alignment lies in its ability to precisely label and align audio files, which plays a pivotal role in advancing linguistic research and supporting resource development for diverse language communities. For example, in phonetic studies, researchers increasingly rely on forced alignment to expedite the analysis of spoken language data. This reliance is no surprise, as the technique greatly enhances the efficiency of acoustic-phonetic analysis by reducing the time and effort required for manual alignment [Yuan et al., 2013].

Manual alignment, while accurate, is labor-intensive and impractical for large-scale projects. It often involves annotators listening to audio and painstakingly marking boundaries, which can lead to inconsistencies and fatigue-related errors. In contrast, forced alignment offers a significant efficiency advantage, completing tasks up to 200 to 400 times faster than manual methods [Yuan et al., 2013]. The scalability of forced alignment is a key reason for its growing popularity. It allows researchers to efficiently process large corpora of audio data with minimal human effort. For instance, tools like the Montreal Forced Aligner use well-established algorithms based on Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM), delivering robust and dependable alignment for both phonetic and word-level tasks [McAuliffe et al., 2017]. Additionally, modern advancements such as WhisperX and wav2vec 2.0-based(is wav2vec neural?) systems incorporate neural architectures, enabling them to handle multilingual and diverse acoustic scenarios, which further enhances the adaptability of FA. A comprehensive overview of forced alignment tools available in 2018, including their underlying algorithms, supported languages, and programming platforms, can be found in [Pettarin, 2018].

While FA has many advantages, it is not without limitations. Its accuracy heavily depends on the quality of the pronunciation dictionary and the acoustic models used. In some cases, especially with non-standard or spontaneous speech, manual alignment may still be preferred for its flexibility and ability to

handle edge cases. However, for most applications, FA remains the go-to method due to its unparalleled speed and scalability.

[Rousso et al., 2023] compares modern ASR methods for forced alignment (FA) by evaluating three prominent models: MFA, WhisperX, and the Massively Multilingual Speech (MMS) Model.

The MMS model is based on wav2vec 2.0 [Baevski et al., 2020], a self-supervised Transformer-encoder framework for learning speech representations. Pre-trained on a vast multilingual dataset covering over 1,000 languages, it focuses on representing contextualized speech information across many languages. It is fine-tuned using Connectionist Temporal Classification (CTC) loss, which aligns the output sequence with input audio without requiring pre-aligned training data. Instead of enforcing strict temporal alignment, CTC estimates probabilities over all possible alignments between input and output. This flexibility, however, leads to poorer temporal precision, making MMS less suited for forced alignment tasks requiring exact word or phoneme boundaries [Graves et al., 2006].

WhisperX builds on Whisper [Radford et al., 2022], an encoder-decoder Transformer trained on 680,000 hours of audio. Whisper's encoder processes up to 30-second chunks of speech to create fixed-length representations, while its decoder generates tokens sequentially. WhisperX enhances Whisper by incorporating external voice activity detection and phoneme alignment to improve word-level timestamps. These enhancements address Whisper's original limitations in generating accurate token alignments, which stem from its architecture and loss function [Bain et al., 2023].

MFA achieved the best result in [Rousso et al., 2023]. Its user-friendliness, quick installation and best performance as outlined in the paper, made me chose it and that is why I will explain its architecture in detail.

3.1 Montreal Forced Alignment

The Montreal Forced Aligner is widely recognized for its usability and popularity in both linguistic and speech research. Its appeal lies in its ability to provide accurate, high-quality forced alignment while maintaining user-friendly interfaces and workflows. Designed as a wrapper for the Kaldi ASR toolkit [Povey et al., 2011], MFA simplifies the often-complex process of creating and using acoustic models, making it accessible even for non-expert users [McAuliffe et al., 2017]. MFA allows customization of acoustic models, lexicons, and datasets to suit specific languages or dialects.

One of the reasons for its popularity is its open-source nature, which encourages widespread adoption across academic and industrial contexts. Additionally, MFA supports multiple alignment levels, including phonemes and words, making it suitable for various tasks like phonetic analysis, prosody studies, and sociolinguistic research [Gonzalez et al., 2020]. Its integration with standard phonetic alphabets, like ARPAbet [Shoup, 1980], further enhances its usability by aligning well with existing linguistic tools and frameworks. MFA provides detailed documentation and straightforward commands, reducing the learning curve for new users.

3.1.1 Kaldi

MFA is built upon the Kaldi architecture, leveraging its robust and modular design for speech processing. Before feature extraction begins, Kaldi requires the audio data to be prepared, including preprocessing raw audio signals and associated metadata. Audio is typically processed into a standard format (e.g.,

mono, 16kHz) using external tools like `ffmpeg`, as Kaldi does not handle these preprocessing steps automatically. Preparing audio in advance ensures compatibility with Kaldi's pipelines and allows efficient processing. This step is particularly important when working with high-resolution audio, such as YouTube downloads, where downsampling and conversion to a standardized format can significantly reduce file sizes.

Long audio files are optionally segmented into smaller, manageable chunks, often based on silence detection or provided timestamps. This segmentation is crucial for tasks such as forced alignment, where precise alignment of phonemes or words with audio is required. By dividing lengthy recordings, the system ensures higher accuracy and prevents potential memory and processing bottlenecks. Following this, the pronunciation lexicon is loaded, mapping words in the transcript to their corresponding phoneme sequences.

The next step involves converting the raw audio signals into feature representations that are suitable for acoustic modeling. To achieve this, the audio signal is divided into overlapping frames, typically 25 milliseconds long, with a 10-millisecond shift between frames. Each frame serves as a snapshot of the audio and is processed independently in the subsequent steps. A Short-Time Fourier Transform (STFT) is applied to each frame, transforming the signal from the time domain to the frequency domain. This conversion generates a spectrum that represents the frequencies and their amplitudes in each frame, which is essential for distinguishing phonemes. Vowels, for instance, exhibit distinct spectral patterns compared to consonants, making frequency domain analysis critical for speech recognition.

Once the spectral information is obtained, it is mapped onto the Mel scale, which reflects human auditory perception by emphasizing lower frequencies and compressing higher ones [Stevens et al., 1937]. This transformation accounts for the nonlinear sensitivity of the human ear, ensuring that the features extracted are perceptually relevant. Triangular filters are then applied to the Mel-scaled spectrum, grouping the energy of neighboring frequencies into broader bands. The output is logarithmically scaled to mimic the human perception of loudness, where smaller changes in quiet sounds are perceived as more significant than equivalent changes in louder sounds. Finally, the log-compressed filter bank outputs are transformed using the Discrete Cosine Transform (DCT), which decorrelates the data and reduces redundancy. This results in a compact set of Mel-Frequency Cepstral Coefficients (MFCCs), typically 12-13 per frame, which capture the spectral envelope of the speech signal. These coefficients are robust to variations in pitch and channel conditions, making them effective for speech recognition tasks.

Kaldi's forced alignment relies on acoustic models, consisting of models such as Gaussian Mixture Models with Hidden Markov Models, to map frames to phonemes or words. Users can customize these models to fit specific tasks. Advanced scenarios can incorporate Deep Neural Networks (DNNs) to replace or augment GMM-HMM systems.

The system integrates a Weighted Finite-State Transducer (WFST) [Mohri et al., 2002] to unify the acoustic model, pronunciation lexicon, and language model into a decoding graph. This graph facilitates efficient decoding by representing all possible sequences weighted by their probabilities. Language models, typically in ARPA format [Shoup, 1980], add contextual constraints to enhance recognition accuracy by distinguishing between similar-sounding words based on context.

Once the decoding graph is constructed, the system processes the input audio, searches for the most likely sequence of states using algorithms like Viterbi, and generates transcriptions. During this process, Kaldi produces timestamps by mapping frame indices to time values using the frame shift. These

timestamps align phonemes or words with their corresponding segments in the audio, providing precise temporal information critical for transcription and forced alignment tasks.

Detailed installation for MFA can be found in 6.2

4 Automatic Speech Recognition

Speech recognition, also known as Automatic Speech Recognition (ASR), refers to the technology that enables machines to understand and transcribe human speech into text. Over the years, ASR has found applications in diverse fields, including virtual assistants, transcription services, and accessibility tools. The development of large-scale machine learning models has significantly improved the accuracy and robustness of ASR systems, making them suitable for complex tasks involving noisy environments and multiple languages [Jurafsky and Martin, 2009].

Recent advancements in neural network architectures have revolutionized the field of Automatic Speech Recognition, giving rise to large language models that harness extensive datasets to deliver state-of-the-art performance. Models such as Whisper, developed by OpenAI, utilize encoder-decoder architectures to effectively handle transcription and translation tasks across multiple domains and languages without relying on self-supervision and self-training techniques [Radford et al., 2022].

Unlike traditional ASR systems, which often require manual adaptation for specific tasks or domains, these models demonstrate remarkable zero-shot capabilities. This means they can tackle new tasks, such as transcribing previously unseen languages or noisy environments, without requiring additional training. However, while zero-shot performance is impressive, fine-tuning these models on domain-specific datasets often yields even better results. By tailoring the model to a particular context, such as industry-specific jargon or recurring patterns in specialized speech, the model can achieve significantly higher accuracy [Xiao et al., 2021].

For instance, fine-tuning has been shown to enhance recognition of domain-specific terminology and reduce errors in specialized settings, such as medical transcriptions or financial reporting [Hsu et al., 2021]. This is particularly critical in scenarios where technical terms, acronyms, or unique phrasing appear frequently, as the model learns to better align its predictions with the target vocabulary. Thus, while Whisper's pre-trained model already offers robust performance, fine-tuning serves as a valuable tool for optimizing its functionality for specific use cases.

Whisper has served as the foundation for the development of an enhanced model called WhisperX [Bain et al., 2023]. This advanced system builds upon Whisper's capabilities and outperforms not only its predecessor but also other prominent models like wav2vec2 [Baevski et al., 2020] in tasks such as long-form audio transcription and word segmentation. Consequently, WhisperX is regarded as the current state-of-the-art open-source model for transcription tasks, particularly for handling extended audio recordings.

WhisperX is specifically designed to provide efficient and accurate speech transcription for long-form audio, with the added benefit of word-level time alignment. The system begins by segmenting the input audio using VAD. This process identifies regions of active speech and divides the audio into approximately 30-second chunks, ensuring the segment boundaries fall in areas with minimal speech activity. These segments are then processed in parallel using Whisper for transcription. Finally, a phoneme recognition model performs forced alignment, refining the word-level timestamps and delivering high-

throughput transcription with precise temporal markers.

While WhisperX has demonstrated superior performance in long-form transcription and word segmentation, its role in forced alignment has raised some questions. For example, [Rousso et al., 2023] compared the performance of WhisperX, Massively Multilingual Speech Model, and the Montreal Forced Aligner. Their findings suggest that MFA remains the best tool for forced alignment tasks. However, it is important to note that their study did not assess transcription quality, focusing solely on alignment accuracy. Similarly, the WhisperX paper does not explicitly evaluate forced alignment against other systems, leading to the belief that WhisperX truly excels as a state-of-the-art open-source transcription system rather than as a forced alignment tool. This highlights the need for further comprehensive evaluations to solidify WhisperX's standing across both transcription and alignment tasks.

Since WhisperX is built upon the foundation of Whisper, it allows for seamless transfer of weights from a fine-tuned Whisper model to a WhisperX model. This compatibility makes it easier to adapt WhisperX for specific domains or use cases where a fine-tuned Whisper model has already been optimized. In addition to its flexibility, WhisperX excels not only in transcription accuracy but also in processing efficiency. It achieves a remarkable 12x faster transcription speed compared to Whisper, making it particularly suitable for scenarios involving long-form audio or large-scale transcription tasks [Bain et al., 2023]. This significant improvement in speed is attributed to its parallelized processing pipeline, which uses VAD for efficient segmentation and parallel transcription of audio chunks.

Since my goal is to transcribe text, I have decided to compare Whisper, WhisperX, and a fine-tuned version of Whisper to truely highlight its performance. To support this comparison, I will provide a detailed explanation of the Whisper architecture and further elaborate on what WhisperX does differently and better.

4.1 Whisper's Architecture

Whisper [Radford et al., 2022] is trained on 680,000 hours of audio data along with its corresponding transcriptions. The audio is first resampled to 16 kHz, and an 80-channel Log-Mel Spectrogram is computed using 25-millisecond windows with a stride of 10 milliseconds. Interestingly, this spectrogram setting matches the one used in Kaldi in section 3.1.1. Although no sources or arguments explicitly state why this setting was chosen, it can be assumed that this configuration provides the best representation of spectrograms, though this is not explicitly documented. Next, the features are normalized.

Whisper employs an encoder-decoder Transformer architecture [**?**] due to its proven scalability and reliability. The encoder processes the normalized features using two convolution layers with a filter width of 3 and the Gaussian Error Linear Units (GELU) activation function [Hendrycks and Gimpel, 2023]. GELU has been shown to improve performance in both natural language processing and speech tasks compared to ReLU and ELU activation functions.

Sinusoidal positional embeddings are added to the output of the stem, after which the encoder Transformer blocks are applied. Since Transformers process input sequences as a set of tokens without any inherent order, this step provides the encoder with the necessary positional information. The Transformer uses pre-activation residual blocks [Child et al., 2019], and the encoder uses a stack of Transformer blocks with self-attention and feedforward layers where a final layer normalization is applied to the encoder output.

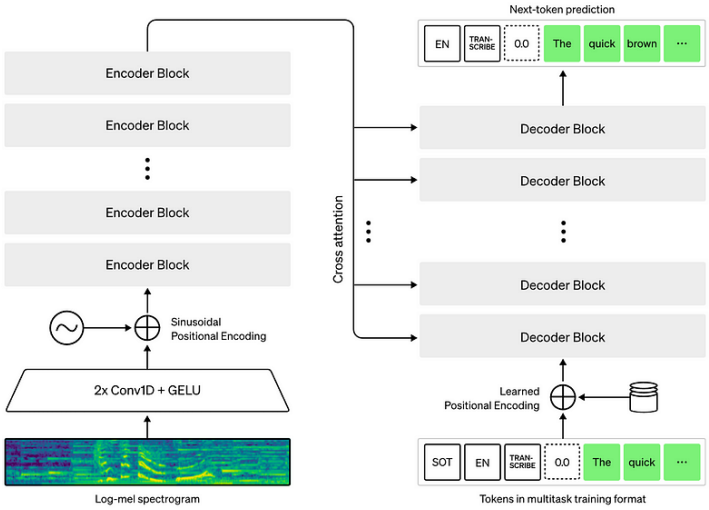


Figure 1: Whisper's architecture visualized [Radford et al., 2022]. The audio is first converted into a log-mel spectrogram, which represents the input features. These features are processed through two convolutional layers to extract patterns and reduce dimensionality. Positional encodings are then added and the processed data is passed to a transformer-based encoder-decoder architecture. The encoder extracts meaningful representations of the audio, while the decoder generates the corresponding text output.

After the encoder processes the input audio, the decoder connects to the encoder's output using a cross-attention mechanism. At each decoding step, the decoder attends to the encoder's outputs to extract relevant context. This mechanism allows the decoder to focus on specific parts of the encoder's output that are most useful for generating the next token in the sequence. In the cross-attention layer, the encoder outputs serve as the key and value inputs, while the decoder generates a query based on the tokens produced so far. The attention mechanism then computes a weighted sum of the encoder outputs, guided by the query, to retrieve contextually relevant information from the encoded audio features.

Once the context is retrieved, the decoder processes it alongside the sequence of previously generated tokens. The combined information passes through additional Transformer layers in the decoder, where it is further refined. Finally, the decoder predicts the next token by generating a probability distribution over all possible tokens, incorporating both the encoder's context and the decoder's own token history. This process repeats iteratively until the entire output sequence is generated.

Whisper achieves impressive results without relying on self-supervision or self-training techniques, showing that training on a large and diverse supervised dataset alone can greatly enhance the robustness of a speech recognition system. Moreover, the paper highlights that the results could likely be improved further through fine-tuning. The authors also suggest that Whisper's robustness may be partly attributed to its powerful decoder.

4.2 Fine-Tuning

Fine-tuning is a highly effective technique in machine learning that improves a model's performance by adapting it to new data. As a form of transfer learning, it involves taking a pre-trained model, originally trained on one task, and adjusting its weights for a related task. The process typically includes selecting a suitable pre-trained model, preparing the data, and iteratively refining the model to achieve better results [Quinn, 2020]. This approach is particularly beneficial when working with limited data, as it allows the use of knowledge already embedded in the pre-trained model. While fine-tuning becomes less essential with access to large datasets, where training a model from scratch can yield excellent results, it is still valuable for enhancing model performance further or tackling tasks that differ from those for which the model was initially trained.

Whisper is designed to be user-friendly and optimized for fine-tuning. Users can download a pre-trained model with a customizable number of parameters and apply it to their own datasets. For instance, I provided the model with audio data and corresponding transcripts to tailor it to my needs.

5 Dataset

While there is already a dataset consisting of 5,000 hours of recorded company earnings calls and their respective transcriptions [O'Neill et al., 2021], it is limited to academic research and internal use, which excludes commercial applications. The dataset is preprocessed into 5 to 15-second segments and has the ideal 16 kHz sample rate in mono, making it ideal for fine-tuning Whisper. However, to monetize this content legally, prior written consent or a separate licensing agreement with Kensho would be necessary. This restriction is not suitable for my target group of investors, individuals who invest with limited knowledge and lack access to premium tools like Bloomberg, which can cost several thousand dollars annually [Insights, 2025]. Therefore, I aim to create my own dataset tailored specifically for my needs. Building my dataset has distinct advantages, such as having full knowledge of the sources and the ability to narrow the domain further. Instead of using a unified dataset that spans various industries like healthcare, industrials, and information technology, I want to focus exclusively on central banks, particularly the Federal Reserve.

The data for this project was collected from the official YouTube channel of the Federal Open Market Committee (FOMC), which is a component of the U.S. Federal Reserve. For this seminar paper, I focused solely on data from the U.S. Federal Reserve. This decision kept the project manageable and provided access to official transcripts of speeches, ensuring higher data accuracy and consistency. Similar datasets could also be sourced from other organizations, like the European Central Bank, the Reserve Bank of Australia, or other national financial institutions, depending on the research objectives. However, transcription availability may vary across these institutions.

The data collection involved downloading press conference videos along with their official transcripts. The videos were converted into WAV audio files using the ffmpeg tool, which requires prior installation. Each press conference typically starts with a 20-minute prepared statement by the spokesperson, followed by a Q&A session with journalists. This format often leads to uneven speaking times, with the spokesperson dominating the audio. Nonetheless, normalization techniques, such as those in the Montreal Forced Aligner and Whisper, were applied to address potential dataset biases. As a result, this imbalance should not significantly affect fine-tuning and analysis, especially when the task is transcription and not speaker classification.

One notable issue during this process was inconsistent naming conventions for the video files. To resolve this, I standardized the naming system, improving organization and enabling efficient dataset management. Each audio file lasted about 50 minutes. For fine-tuning, I selected 20 files, ensuring around 20 hours of audio data.
To streamline data extraction, I created a CSV file containing 46 video links, their corresponding transcript links, and dates. Python scripts were then used to automate this process efficiently. The process of downloading 20 videos and preparing them for fine-tuning took approximately 7 minutes, leveraging a Python library for YouTube video downloads. These libraries, however, often operate on the edge of copyright regulations and may be removed. If one is taken down, another typically replaces it, perpetuating this cycle. The post-processing module ensured compliance with Whisper's requirements by resampling audio to 16 kHz (the maximum supported frequency) and converting it to mono. WAV files, chosen for their high-quality, lossless properties, were downsampled from 44 kHz to meet these specifications.

To include official transcriptions, I expanded the CSV file to include direct transcript links in PDF format. However, processing these PDFs presented challenges, as they often contained extra metadata or annotations, such as "[laughter]." These elements needed to be cleaned during preprocessing to ensure only spoken content was retained, a critical step for accurate alignment using forced aligner tools.

Something interesting to point out: While the individuals speaking at the conference are considered highly articulate, occasional stutters and self-corrections still occur. [Lea et al., 2023] investigates the experiences of individuals who stutter when using consumer-grade ASR systems. The study highlights that disfluencies, such as stuttering and self-corrections, often result in higher error rates and user dissatisfaction with ASR outputs. Interestingly, in this dataset, the transcriptions do not include stuttered words, but self-corrections are accurately transcribed along with the preceding incorrect utterance. While there are limited sources on the specific impact of stuttering in speech transcription tasks, Whisper's transcription approach [Radford et al., 2022] appears well-suited for this scenario. The model will be fine-tuned on audio that includes some instances of stuttering, while the labels will exclude these disfluencies, ensuring that stutters are not reflected in the transcriptions. This is advantageous because stutters are generally non-contributory to the task of transcription and do not hold semantic importance for the end-user. For investors relying on these transcripts, stuttering does not enhance or diminish the interpretability of the speech content. What matters is the accurate capture of key phrases and financial terminology that directly contribute to actionable insights.

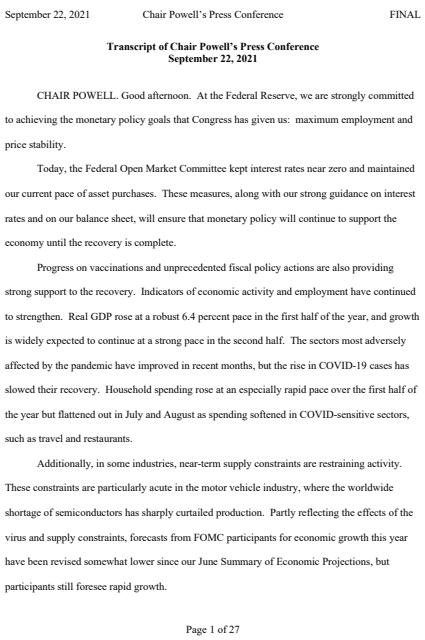
5.1 Preprocessing

As previously mentioned, the audio was extracted in 16 kHz mono and saved in WAV format, which is ideal for processing with tools like Whisper.

To extract text from the PDF files, optical character recognition (OCR) was applied using the `pypdf2` library [Fenniak et al., 2022]. However, the extracted text was not guaranteed to be entirely accurate, as no gold standard reference was available for comparison. This lack of a benchmark made it challenging to evaluate the accuracy of the OCR output reliably. Manual inspection revealed numerous inconsistencies, such as "fed" being misread as "f ed," where random letters were separated from words. While some of these issues could potentially be fixed with regular expressions, the irregularity of these errors made it difficult to address them systematically since there are also correct expressions like "a strategy".

Since OCR is not the main focus of this paper, I did not explore advanced OCR methods or evaluate

state-of-the-art techniques. Improving OCR accuracy would undoubtedly be beneficial, particularly for fine-tuning models, but these specific errors do not directly impact the model's performance. For instance, the model may predict the correct word, such as "fed," but discrepancies in the extracted text (e.g., "f ed") would skew evaluation metrics like word error rate (WER). As my goal is to compare model performance, it is critical to understand which model performs better, even if the evaluation itself is affected by inconsistencies in the dataset.



September 22, 2021 Chair Powell's Press Conference FINAL

Transcript of Chair Powell's Press Conference
September 22, 2021

CHAIR POWELL. Good afternoon. At the Federal Reserve, we are strongly committed to achieving the monetary policy goals that Congress has given us: maximum employment and price stability.

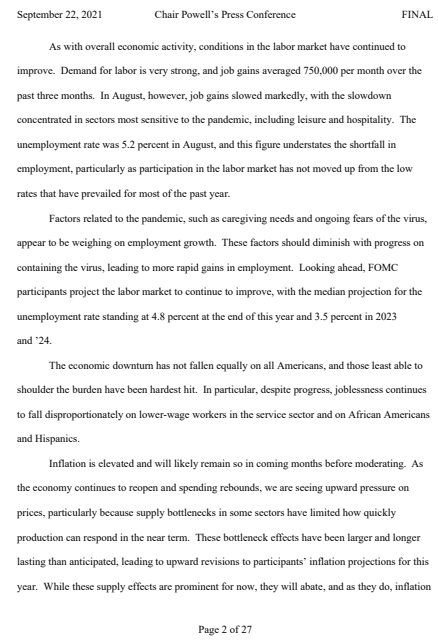
Today, the Federal Open Market Committee kept interest rates near zero and maintained our current pace of asset purchases. These measures, along with our strong guidance on interest rates and on our balance sheet, will ensure that monetary policy will continue to support the economy until the recovery is complete.

Progress on vaccinations and unprecedented fiscal policy actions are also providing strong support to the recovery. Indicators of economic activity and employment have continued to strengthen. Real GDP rose at a robust 6.4 percent pace in the first half of the year, and growth is widely expected to continue at a strong pace in the second half. The sectors most adversely affected by the pandemic have improved in recent months, but the rise in COVID-19 cases has slowed their recovery. Household spending rose at an especially rapid pace over the first half of the year but flattened out in July and August as spending softened in COVID-sensitive sectors, such as travel and restaurants.

Additionally, in some industries, near-term supply constraints are restraining activity. These constraints are particularly acute in the motor vehicle industry, where the worldwide shortage of semiconductors has sharply curtailed production. Partly reflecting the effects of the virus and supply constraints, forecasts from FOMC participants for economic growth this year have been revised somewhat lower since our June Summary of Economic Projections, but participants still foresee rapid growth.

Page 1 of 27

Figure 2: First page layout of a transcript from a press conference of the FOMC.



September 22, 2021 Chair Powell's Press Conference FINAL

As with overall economic activity, conditions in the labor market have continued to improve. Demand for labor is very strong, and job gains averaged 750,000 per month over the past three months. In August, however, job gains slowed markedly, with the slowdown concentrated in sectors most sensitive to the pandemic, including leisure and hospitality. The unemployment rate was 5.2 percent in August, and this figure understates the shortfall in employment, particularly as participation in the labor market has not moved up from the low rates that have prevailed for most of the past year.

Factors related to the pandemic, such as caregiving needs and ongoing fears of the virus, appear to be weighing on employment growth. These factors should diminish with progress on containing the virus, leading to more rapid gains in employment. Looking ahead, FOMC participants project the labor market to continue to improve, with the median projection for the unemployment rate standing at 4.8 percent at the end of this year and 3.5 percent in 2023 and '24.

The economic downturn has not fallen equally on all Americans, and those least able to shoulder the burden have been hardest hit. In particular, despite progress, joblessness continues to fall disproportionately on lower-wage workers in the service sector and on African Americans and Hispanics.

Inflation is elevated and will likely remain so in coming months before moderating. As the economy continues to reopen and spending rebounds, we are seeing upward pressure on prices, particularly because supply bottlenecks in some sectors have limited how quickly production can respond in the near term. These bottleneck effects have been larger and longer lasting than anticipated, leading to upward revisions to participants' inflation projections for this year. While these supply effects are prominent for now, they will abate, and as they do, inflation

Page 2 of 27

Figure 3: The page layout of a transcript, excluding the first page, from a press conference of the FOMC.

After extraction, the text was processed to identify speaker names, which were consistently marked with "NAME." For example, in 2, "CHAIR POWELL." was used to denote the speaker. This standardized format across all transcriptions simplified the task of scanning for speaker names and removing them. It was necessary to exclude these names because the forced aligner processes only spoken content, and including non-spoken text significantly slowed down the alignment process—by up to 4x, based on my observations. Furthermore, all other unspoken text was removed like the page number, date, version and title.

A notable challenge with the OCR library was its use of UTF-8 encoding without an option to convert the output to Latin-1, which Whisper requires. This mismatch caused issues during evaluation because the two encodings do not align seamlessly. For example, fractions like "¼" in the transcripts were handled differently by Whisper, which transcribed them as "one quarter" or "1/4." Although these variations represent the same value, they resulted in false mismatches in WER calculations. This problem becomes more pronounced in financial or statistical contexts, where fractions are common and carry

significant meaning. While preprocessing the text to standardize these representations before calculating WER could mitigate the issue, this adds an extra layer of complexity to the workflow, which may not be practical for larger datasets. Finally, non-essential elements like newlines and text enclosed in square brackets (e.g., "[laughter]") were removed to clean the dataset. The processed text was then saved as a .txt file, ready for the next stages of analysis and model fine-tuning.

6 Methods

6.1 Setup

Table 1: Whisper Model Sizes and Specifications [Radford et al., 2022]

Size	Parameters	English-only Model	Multilingual Model	Required VRAM	Relative Speed
tiny	39 M	`tiny.en`	`tiny`	1 GB	32x
base	74 M	`base.en`	`base`	1 GB	16x
small	244 M	`small.en`	`small`	2 GB	6x
medium	769 M	`medium.en`	`medium`	5 GB	2x
large	1550 M	N/A	`large`	10 GB	1x

While Table 1 refers to the Video Random Access Memory (VRAM), which is exclusively part of a Graphics Processing Unit (GPU), it is important to note that the listed VRAM requirements apply only to the inference stage of using the Whisper model. Inference refers to running the model to transcribe audio without modifying its parameters. The VRAM specified in the table is sufficient for loading the pretrained model and processing audio inputs to generate transcriptions. However, these requirements do not account for fine-tuning the model, which involves updating its parameters with domain-specific data. Fine-tuning is a significantly more resource-intensive process that typically requires larger VRAM, higher system memory, and additional storage for handling intermediate computations and model checkpoints.

The most critical parameters for fine-tuning a Whisper model include `per_device_batch_size` and `generation_max_length`. These parameters directly influence the memory requirements and training dynamics of the model. For instance, on my setup with 8 GB of VRAM, I can fine-tune the `small` Whisper model with a batch size of 8. However, increasing the batch size to 16 causes the GPU to run out of memory, making it impractical to use larger batch sizes. Gradient accumulation can simulate larger batches by dividing them into smaller micro-batches and accumulating gradients over multiple steps, but it comes at the cost of longer training times. Furthermore, with my current specifications, fine-tuning the `medium` or `large` Whisper models is not feasible due to their significantly higher memory requirements. These models demand more VRAM for storing the model weights, activations, and gradients during backpropagation. For example, the `medium` model would require approximately 12-16 GB of VRAM for fine-tuning with a batch size of 8, while the `large` model may require over 20 GB. So I proceeded with the `small` Whisper model.

6.2 Running A Forced Aligner

After completing the preprocessing steps in 5.1, the data was prepared for alignment using the Montreal Forced Aligner. Initially, I attempted to automate the process by writing a script to loop through all audio

files. However, I encountered a limitation where the MFA outputs critical information only through the terminal. To address this, I processed the audio files in batches through the terminal and initialized a PostgreSQL server, as the default SQLite database caused issues during alignment. Below is a detailed explanation of the steps and commands executed in the terminal, assuming the Conda virtual environment is active and MFA is already installed, as mentioned in the Appendix.

1. **Download the Acoustic Model and Dictionary:**

```
mfa model download acoustic english_mfa
mfa model download dictionary english_us_mfa
```

   Explanation: These commands download the English acoustic model (`english_mfa`) and the U.S. English pronunciation dictionary (`english_us_mfa`) needed for alignment.

2. **Initialize the PostgreSQL Server:**

```
mfa server init
```

   Explanation: This sets up a PostgreSQL database server, avoiding issues caused by SQLite during the alignment process.

Starting a Session

1. **Start the PostgreSQL Server:**

```
mfa server start
```

2. **Run the Aligner:**
   Use the following command to align the data:

```
mfa align --clean --verbose data/sample_data \
    english_us_mfa english_mfa combined_output \
    --use_postgres --auto_server \
    --beam 100 --retry_beam 400
```

   Explanation of Parameters:

   - `--clean`: Removes temporary files and directories after processing.
   - `--verbose`: Outputs detailed logs, including warnings and progress updates.
   - `data/sample_data`: Path to the directory containing the audio and transcription files.
   - `english_us_mfa`: Specifies the pronunciation dictionary for alignment.
   - `english_mfa`: Specifies the acoustic model for alignment.
   - `combined_output`: Directory where aligned output files will be saved.
   - `--use_postgres`: Configures MFA to use PostgreSQL instead of SQLite.

- `--auto_server`: Automatically starts and manages the PostgreSQL server.
- `--beam 100`: Sets the beam width for decoding during alignment.
- `--retry_beam 400`: Expands the search range for retries, increasing reliability.

Depending on how good the transcriptions are for the audio, processing a 16 kHz mono audio file with a length of 50 minutes can take at least 20 minutes, or up to 80 minutes if the transcription is not perfect. If this happens, it is recomended to review the transcription again and check where the inconsistencies might be.

After the alignment process is completed, the Montreal Forced Aligner (MFA) returns TextGrid files. These files contain the vocabulary of a given audio file along with corresponding timestamps. Since FOMC speeches are approximately 50 minutes long and Whisper fine-tuning requires segments with a maximum length of 30 seconds, the audio had to be chunked into smaller parts based on the timestamps provided by MFA. Although Whisper can automatically segment the data using its own forced aligner, it performs worse than using MFA [Rousso et al., 2023] and uses more computational resources during fine-tuning. MFA also identifies silences and includes their timestamps, which I used as natural cutting points for segmenting the audio. Whenever an arbitrary duration of silence occurred, the audio was split at that point. This approach, while simple, can occasionally produce segments with very few words or even single-word audio, which mostly consist of function words like ”and”, ”to” and ”is”. Despite this, such segmentation is still suitable for Whisper fine-tuning, as the model is capable of handling both short and long audio segments. The average segment length ended up being around 5 seconds, consisting of a mix of shorter and longer clips. In case of audio files exceeding 30 minutes, Whisper chunked it itself. This approach comes with trade-offs, such as increased training time. Shorter segments require padding to match the longest segment in a batch, which can result in computational inefficiencies.

One major challenge with forced alignment is the lack of objective metrics to evaluate its quality. The only viable approach is manual inspection. To address this, I implemented a process to randomly select audio segments along with their transcriptions for review. By manually inspecting approximately 20 randomly selected audio-transcription pairs, I concluded that the alignment and segmentation performed by MFA were accurate, albeit the self-implemented chunking could be expanded on. Nevertheless, I assume the performance is mostly due to FOMC's high quality audio recordings and the speakers' pronounciations. Despite its simplicity, the silence-based segmentation approach proved effective for this dataset and met the requirements for Whisper fine-tuning.

Overall, from the 1,000 minutes of audio, I segmented them into 16,000 shorter audio files with an average length of almost 4 seconds. Fine-tuning Whisper with longer audio could be considered in a follow-up.

6.3 Fine-Tuning Whisper

The primary objective of this paper is to fine-tune a speech recognition model to improve performance for specific tasks. OpenAI's Whisper was chosen for this purpose due to its state-of-the-art performance in automatic speech recognition [Radford et al., 2022]. Whisper offers extensive documentation and guides for fine-tuning, and HuggingFace provides practical examples that simplify the process [Gandhi, 2022]. Given hardware limitations, the small Whisper model was selected, as larger models require significantly more VRAM and longer training times, which were beyond the resources available for this project. Initial tests were conducted on the full dataset of 16,000 short audio files to estimate the time and feasibility of training. These tests revealed significant challenges like frequent crashes occuring when

attempting to train on larger models, using high batch sizes, or processing the full dataset. E.g. running the full dataset with large model would need at least 124 hours, an estimation provided by the training function.

To address these issues, I reduced the batch size and focused on a subset of the dataset. This adjustment allowed the model to train without exhausting system resources. The estimated time for training and evaluation was approximately 36 hours for this configuration, which underscored the need for careful resource management in fine-tuning processes. 36 hours were also not feasible, so I used Low-rank Adaptation (LoRA), which which freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks [Hu et al., 2021]. This led to an estimate of 12 hours training time. A lot of runs were conducted until something promising came out.

6.4 First Run

The objective of the first run was to gain an estimate of training time and assess preliminary performance. For this, I used a small subset of data consisting of two hours of audio, with training configured for five epochs, a batch size of eight, and a step size of 500. The training process took approximately five hours and resulted in a loss of around 1.4. While the loss value was not particularly promising, this run served as an introduction to fine-tuning a model and helped identify potential challenges and limitations. I evaluated this model and its performance in word error rate against the base model, but there were no significant differences. I attributed this to the quality of the training data, which could consist of many function words, and the low learning rate.

6.5 Second Run

The second run involved a more extensive configuration. A dataset of 6,000 WAV files was selected for training, with each file containing at least seven words. This criterion was chosen to reduce training time while maintaining a similar amount of data, as shorter audio files (fewer than seven words) require additional padding, which increases processing time without significantly contributing to model improvement. The training parameters remained largely consistent with the first run, including a batch size of eight and five epochs. However, the learning rate was doubled to compensate for the limited number of epochs, ensuring a greater impact on training. Despite these adjustments, attempts to use the entire dataset led to VRAM exhaustion. Training proceeded with the modified dataset, and the learning curve can be seen in Figure 4. The loss began to plateau after 500 steps, even though the training was initially planned to run for 1,500 steps. After observing no further improvements in the loss and evaluating the word error rate, I terminated training early at 900 steps. This required 14 hours of training time, exceeding the initial estimate of 12 hours. However, the final loss value of approximately 0.13 was satisfactory for this experiment, which is way better compared to the first run.

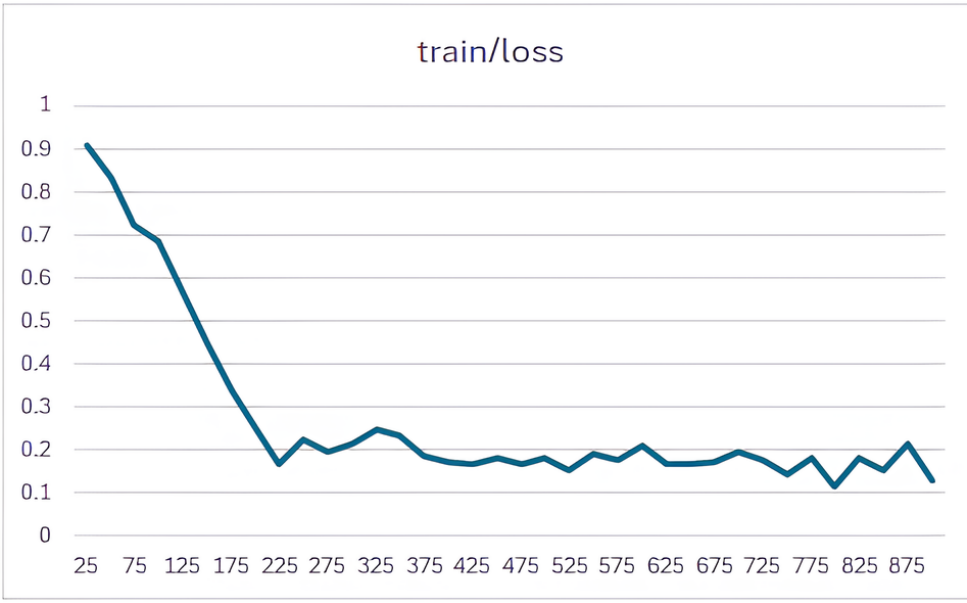


Figure 4: Loss during fine-tuning in the second run visualized.

6.6 Evaluation

After having trained a model, I was finally able to evaluate tasks with. Since WhisperX is built upon Whisper, one would be able to transfer weights from the fine-tuned model, but I encountered many problems in transferring the weights leading to me not being able to transfer them, even after spending a large amount of time debugging it. This can be looked at in a follow-up.
ChatGPT-4: For transcription tasks of central bank speeches, what is important for an investor to consider in a speech recognition tool?

- *Accuracy in Transcription*
- *Real-Time or Near Real-Time Capabilities*
- *Contextual Understanding*
- *Timestamped Transcriptions*
- *Multilingual Support (if applicable)*
- *Noise Robustness*
- *Data Privacy and Compliance*
- *Speed and Scalability*
- *Cost-Efficiency*
- *Integration Capabilities*

These are valid points raised by ChatGPT, and I aim to address each point in detail.

Contextual understanding is a significant factor, particularly when coupled with sentiment analysis, as

explored by [Ahrens et al., 2024]. This combination could enhance the transcription tool's ability to not only transcribe content but also derive meaningful insights from the tone and sentiment conveyed in central bank speeches. Multilingual is not important in my application case but could be scaled to different languages and English accents. Noise robustness, while important, is less critical in this specific context due to the typically high-quality audio and speech delivery in central bank conferences. Integration capabilities are highly feasible given that a fine-tuned model can be deployed via HuggingFace, allowing easy access to the transcription system. Additionally, the transcriptions can be formatted as text files or integrated into a larger workflow for further processing within an application.

Ultimately, the most critical factors to focus on are accuracy, speed and timestamps. A common evaluation metric for accuracy used in transcription tasks is word error rate (WER):

$$\text{WER} = \frac{\text{Substitutions} + \text{Deletions} + \text{Insertions}}{\text{Total Words in Reference}}$$

For the subsequent tests, I evaluated three different models: the base small Whisper, the fine-tuned small Whisper, and the base small WhisperX. The aim was to analyze their performance in terms of word error rate and processing time on both shorter audio segments and a longer audio file.

Table 2 presents the performance of the models on audio files with durations ranging between 5 and 30 seconds. The fine-tuned Whisper achieved the best accuracy, with a WER of 9.78%, significantly outperforming the base Whisper and WhisperX. This makes it ideal for scenarios where shorter audio files need to be transcribed accurately, such as real-time transcription. However, it is worth noting that the fine-tuned Whisper required slightly more processing time compared to the base Whisper, averaging 6.12 seconds per file. On the other hand, WhisperX demonstrated its strength in speed, with an average processing time of just 0.26 seconds, although its WER of 18.03% indicates a trade-off in accuracy.

For longer audio files, Table 3 highlights the performance differences across the same models when tested on a 50-minute recording. WhisperX emerged as the most efficient, with a WER of 13.55% and a significantly reduced processing time of 22.35 seconds. This speed advantage makes WhisperX particularly suitable for transcribing large volumes of data or performing near real-time analysis. In contrast, the fine-tuned Whisper showed better accuracy (WER of 22.75%) than the base Whisper (WER of 25.83%) but required more time for processing. I attribute the better performance of WhisperX mainly due to its voice activation detection [Bain et al., 2023] and proper chunking of longer audio, something where Whisper currently struggles. This reflects that having a model which can segment audio very well is better than a model, which is trained on that data, but lacks good chunking.

I want to address the significant discrepancies observed in WhisperX's WER between shorter and longer audio segments. Intuitively, one might expect the performance on longer audio to either match or degrade compared to shorter audio, given the challenges of perfectly chunking longer files. However, I suspect this inconsistency is not due to WhisperX itself or the MFA, which I trust to perform reliably. Instead, I believe the primary issue lies in the limitations of my OCR implementation, which may have introduced errors that skewed the evaluation results.

Overall, the results demonstrate that fine-tuning Whisper improves accuracy, particularly on shorter audio files, making it an excellent choice for precision-focused tasks. Meanwhile, WhisperX excels in speed, especially for longer audio, but with some compromises in accuracy. The choice between these

models depends on the specific requirements of the task, such as whether accuracy or speed is the priority.

Model	WER (Avg. %)	Avg. Time (s)
Base Whisper	16.74	5.89
Fine-Tuned Whisper	**9.78**	6.12
WhisperX	18.03	**0.26**

Table 2: Performance comparison of zero-shot Whisper, fine-tuned Whisper, and WhisperX on audio files of length $5 \leq t < 30$ seconds. The reported WER is higher than expected due to OCR and preprocessing issues.

Model	WER (%)	Time (s)
Base Whisper	25.83%	822.99
Fine-Tuned Whisper	22.75%	930.87
WhisperX	**13.55**%	**22.35**

Table 3: Performance of zero-shot Whisper, fine-tuned Whisper and WhisperX on a longer audio file with a duration of 50 minutes. This should be more accurate depiction of WER, since this is a raw audio file without forced aligner being used.

6.7 Examples

Original Transcription: *i don't i don't worry in the near term*

Whisper Transcription: *i don't worry in the near term*
Whisper WER: 25.0%

Fine-tuned Transcription: *i don't i don't worry in the near term*
Fine-tuned WER: 16.67%

WhisperX Transcription: *i don't i don't worry in the near term*
WhisperX WER: 22.22%

This example highlights a transcription where both WhisperX and the fine-tuned model captured a repetition by the speaker. Such repetitions, depending on the context, can be crucial for making informed financial decisions.

Original Transcription: *we will aim to achieve inflation moderately above 2 percent for some time so that inflation averages 2 percent over time*

Whisper Transcription: *we will aim to achieve inflation moderately above 2% for some time so that inflation averages 2% over time*
Whisper WER: 19.05%

Fine-tuned Transcription: *we will aim to achieve inflation moderately above 2 percent for some time so that inflation averages 2 percent over time*

Fine-tuned WER: 0.0%

WhisperX Transcription: *we will aim to achieve inflation moderately above 2% for some time so that inflation averages 2% over time*
WhisperX WER: 19.05%

This example shows how the WER can be skewed, eventhough the transcription is factually correct.

Original Transcription: *that might cause you to change about the policy as it now is projected i would start with that's where i know we were we just had a period of unemployment as you know that was*

Whisper Transcription: *that might cause you to change about the policy as it now is projected well i think we just had a period of unemployment as you know that was*
Whisper WER: 25.0%

Fine-tuned Transcription: *that might cause you to change about the policy as it now is projected start i would say well i think you know we were we just had a period of unemployment as you know that was*
Fine-tuned WER: 16.67%

WhisperX Transcription: *that might cause you to change about the policy as it now is projected start i would say well i think you know we were we just had a period of unemployment as you know that was*
WhisperX WER: 16.67%

Whisper did not pick up a part of the audio. This could be due to audio being incomprehensible and the fine-tuned model being more accustomed to the speaker, while WhisperX generally has better voice detection.

7 Conclusion

Fine-tuning even a small model with just 400 steps already yields significantly better performance in word error rate compared to the base model. However, it is crucial to consider training a model for longer periods to potentially achieve further improvements, and optionally implementing an early stopping mechanism can help avoid overfitting. Even though I thought that the FOMC speeches do not contain excessive amounts of financial jargon, the fine-tuning process still led to a noticeable enhancement in the model's performance. Longer audio segments remain challenging for both the base Whisper model and its fine-tuned counterpart, as their chunking approach struggles to handle extensive recordings effectively. WhisperX, with its superior voice activation detection, performs much better in such scenarios. I genuinely intended to implement a fine-tuned WhisperX model, but the project turned out to be time-intensive, with a significant amount of debugging and extensive research on various methods scattered across the internet. The same challenges applied to working with the Montreal Forced Aligner.

For future work, several improvements could be considered. These include ensuring the correctness of the forced alignments produced by MFA, improving OCR accuracy for text extraction, better audio segmentation techniques to avoid creating numerous single-word audio files and expanding to live tran-

scription. I believe that after improving upon these points, the transcription model can be a valuable asset to an investor with limited access to high-end tools.

References

[Ahrens et al., 2024] Ahrens, M., Erdemlioglu, D., McMahon, M., Neely, C. J., and Yang, X. (2024). Mind your language: Market responses to central bank speeches. *Journal of Econometrics*, page 105921.

[Baevski et al., 2020] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 12449–12460.

[Bain et al., 2023] Bain, M., Huh, J., Han, T., and Zisserman, A. (2023). Whisperx: Time-accurate speech transcription of long-form audio.

[Bernanke, 2007] Bernanke, B. S. (2007). *The Federal Reserve and the Financial Crisis*. Princeton University Press.

[Child et al., 2019] Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers.

[Fenniak et al., 2022] Fenniak, M., Stamy, M., pubpub zz, Thoma, M., Peveler, M., exiledkingcc, and PyPDF2 Contributors (2022). The PyPDF2 library.

[Fratzscher, 2012] Fratzscher, M. (2012). The role of central bank communication in market behavior. *Journal of Economic Perspectives*, 26(4):25–46.

[Gandhi, 2022] Gandhi, S. (2022). Fine-tune whisper for multilingual asr with transformers. Blog post. Published on Hugging Face Blog. Available at `https://huggingface.co/blog/fine-tune-whisper`.

[Gonzalez et al., 2020] Gonzalez, S., Grama, J., and Travis, C. E. (2020). Comparing the performance of forced aligners used in sociophonetic research. *Linguistics Vanguard*, 6(1):20190058.

[Gorman et al., 2011] Gorman, K., Howell, J., and Wagner, M. (2011). Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376. ACM.

[Hansen and McMahon, 2018] Hansen, S. and McMahon, M. (2018). Shocking language: Understanding central bank communication. *American Economic Journal: Macroeconomics*, 10(2):1–36.

[Hendrycks and Gimpel, 2023] Hendrycks, D. and Gimpel, K. (2023). Gaussian error linear units (gelus).

[Hsu et al., 2021] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units.

[Hu et al., 2021] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

[Insights, 2025] Insights, F. (2025). What is a bloomberg terminal (bt)? functions, costs, and alternatives. Accessed January 2025.

[Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing*. Prentice Hall, 2nd edition.

[Lea et al., 2023] Lea, C., Huang, Z., Tooley, L., Narain, J., Yee, D., Georgiou, P., Tran, T. D., Bigham, J. P., and Findlater, L. (2023). From user perceptions to technical improvement: Enabling people who stutter to better use speech recognition.

[Mahr et al., 2021] Mahr, T., Berisha, V., Kawabata, K., Liss, J., and Hustad, K. (2021). Performance of forced-alignment algorithms on children's speech. *Journal of Speech, Language, and Hearing Research*, 64(6S):2213–2222.

[McAuliffe et al., 2017] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech 2017*, pages 498–502.

[Mohri et al., 2002] Mohri, M., Pereira, F., and Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 20(1):69–88.

[O'Neill et al., 2021] O'Neill, P. K., Lavrukhin, V., Majumdar, S., Noroozi, V., Zhang, Y., Kuchaiev, O., Balam, J., Dovzhenko, Y., Freyberg, K., Shulman, M. D., Ginsburg, B., Watanabe, S., and Kucsko, G. (2021). SPGISpeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition. *arXiv e-prints*, page arXiv:2104.02014.

[Pettarin, 2018] Pettarin, A. (2018). Forced alignment tools. `https://github.com/pettarin/forced-alignment-tools`. Accessed: 2025-01-16.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[Pratap et al., 2023] Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., Elkahky, A., Ni, Z., Vyas, A., Fazel-Zarandi, M., Baevski, A., Adi, Y., Zhang, X., Hsu, W.-N., Conneau, A., and Auli, M. (2023). Scaling speech technology to 1,000+ languages.

[Quinn, 2020] Quinn, J. (2020). *Dive into Deep Learning: Tools for Engagement*. Thousand Oaks, California.

[Radford et al., 2022] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

[Rousso et al., 2023] Rousso, R., Cohen, E., Keshet, J., and Chodroff, E. (2023). Tradition or innovation: A comparison of modern asr methods for forced alignment. In *Proceedings of the Conference on Automatic Speech Recognition and Understanding (ASRU)*, Israel and Switzerland. Technion - Israel Institute of Technology and University of Zurich.

[Shoup, 1980] Shoup, J. E. (1980). Phonological aspects of speech recognition. In Lea, W. A., editor, *Trends in Speech Recognition*, pages 125–138. Prentice Hall.

[Stevens et al., 1937] Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *Journal of the Acoustical Society of America*, 8:185–190.

[Xiao et al., 2021] Xiao, A., Zheng, W., Keren, G., Le, D., Zhang, F., Fuegen, C., Kalinli, O., Saraf, Y., and Mohamed, A. (2021). Scaling asr improves zero and few shot learning.

[Yuan and Liberman, 2008] Yuan, J. and Liberman, M. (2008). Speaker identification on the scotus corpus. *The Journal of the Acoustical Society of America*, $123(5_{Supplement}) : 3878 - -3878$.

[Yuan et al., 2013] Yuan, J., Ryant, N., Liberman, M., Stolcke, A., Mitra, V., and Wang, W. (2013). Automatic phonetic segmentation using boundary models. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2306–2310.

A. Appendix

Installation and Setup

Installing the necessary dependencies can be challenging due to version compatibility issues, as each dependency often relies on others being the correct version. Below is a step-by-step guide to setting up the environment and installing the required tools. Consult the internet in case of problems. It is recomended to use Conda for creating an environment, due to is easiness with MFA. Relevant resources:

- WhisperX GitHub Repository
- Whisper GitHub Repository

1. **Create a new Python environment:**

   ```
   conda create --name <env_name> python=3.10
   conda activate <env_name>
   ```

   Replace `<env_name>` with your preferred environment name.

2. **Install PyTorch and related dependencies:**

   ```
   conda install pytorch==2.0.0 torchaudio==2.0.0 \
   pytorch-cuda=11.8 -c pytorch -c nvidia
   ```

3. **Install WhisperX:** Clone and install the WhisperX repository:

   ```
   pip install git+https://github.com/m-bain/whisperx.git
   ```

4. **Install FFmpeg:** For Windows, use Chocolatey to install FFmpeg. First, install Chocolatey (https://chocolatey.org/), then run in the terminal of the environment:

```
choco install ffmpeg
```

5. **Install CTranslate2:** Upgrade or reinstall CTranslate2:

   ```
   pip install --upgrade --force-reinstall ctranslate2==3.24.0
   ```

6. **Install the Montreal Forced Aligner:** MFA can be installed via Conda:

   ```
   conda install -c conda-forge montreal-forced-aligner
   ```

   Alternatively, it can be manually downloaded and installed from the official website.

7. **Install other dependencies as needed:** After the main setup, install any additional dependencies that your specific workflow may require.