Note: Summary is provided on page no. 8 Start with summary before reading the full paper

WhisperX: Time-Accurate Speech Transcription of Long-Form Audio

Max Bain, Jaesung Huh, Tengda Han, Andrew Zisserman

Visual Geometry Group, University of Oxford

{maxbain,jaesung,htd,az}@robots.ox.ac.uk

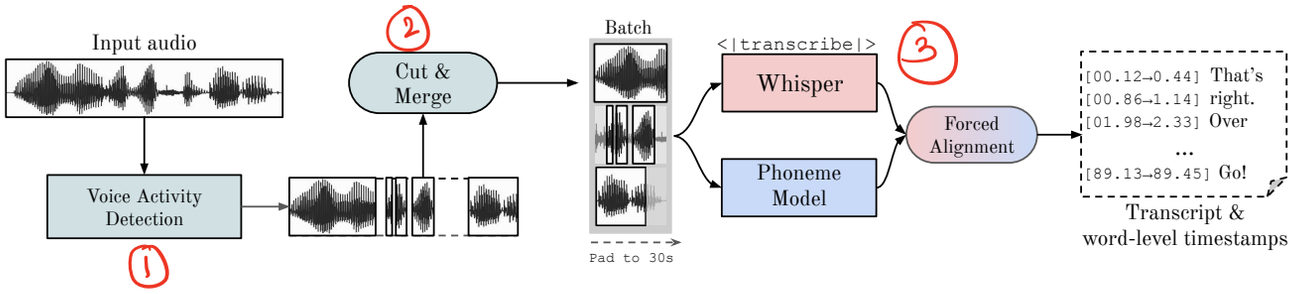

Figure 1: ***WhisperX****: We present a system for efficient speech transcription of long-form audio with* word-level time alignment*. The input audio is first segmented with Voice Activity Detection and then cut & merged into approximately 30-second input chunks with boundaries that lie on minimally active speech regions. The resulting chunks are then: (i) transcribed in parallel with whisper and (ii) forced aligned with a phone recognition model to produce accurate word-level timestamps at high throughput.*

Abstract

Large-scale, weakly-supervised speech recognition models, such as Whisper, have demonstrated impressive results on speech recognition across domains and languages. However, their application to long audio transcription via buffered or sliding window approaches is prone to drifting, hallucination & repetition; and prohibits batched transcription due to their sequential nature. Further, timestamps corresponding each utterance are prone to inaccuracies and word-level timestamps are not available out-of-the-box. To overcome these challenges, we present *WhisperX*, a time-accurate speech recognition system with word-level timestamps utilising voice activity detection and forced phoneme alignment. In doing so, we demonstrate state-of-the-art performance on long-form transcription and word segmentation benchmarks. Additionally, we show that pre-segmenting audio with our proposed VAD Cut & Merge strategy improves transcription quality and enables a *twelve-fold* transcription speedup via batched inference. The code is available open-source[1].

1. Introduction

With the availability of large-scale web datasets, weakly-supervised and unsupervised training methods have demonstrated impressive performance on a multitude of speech processing tasks; including speech recognition [1, 2, 3, 4], speaker recognition [5, 6], speech separation [7] and keyword spotting [8, 9]. Whisper [10] utilises this rich source of data to another scale. Leveraging 680,000 hours of noisy speech training data, including 96 other languages and 125,000 hours of English translation data, they showcase that weakly supervised pretraining of a simple encoder-decoder transformer [11] can robustly achieve *zero-shot* multilingual speech transcription on existing benchmarks.

[1] https://github.com/m-bain/whisperX

Most of the academic benchmarks are comprised of short utterances, whereas real-world applications typically require transcribing long-form audio that can easily be hours or minutes long, such as meetings, podcasts and videos. Automatic Speech Recognition (ASR) models are typically trained on short audio segments (30 seconds for the case of Whisper) and and the transformer architectures employed prohibit transcription of arbitrarily long input audio due to memory constraints.

Recent works [12] employ heuristic sliding-window style approaches which are prone to errors due to (i) overlapping audio, that can lead to inconsistencies in the transcription when the model processes the same speech twice; (ii) incomplete audio: if some words lie on at the beginning or end of the input segment then they can be missed or incorrectly transcribed. Whisper proposes a buffered transcription approach that relies on accurate timestamp prediction to determine the amount of shift the input window by. Such a method is prone to severe drifting since timestamp inaccuracies in one window can accumulate to subsequent windows. They employ a series of handcrafted heuristics in order to reduce these errors with limited success.

Whisper's coupled decoding of both the transcriptions and timestamps with a single encoder-decoder is prone to the usual challenges faced by auto-regressive language generation, namely: hallucination and repetition. This has catastrophic consequences for buffered transcription of long-form and other timestamp-sensitive tasks such as speaker diarization [13, 14], lip-reading [15] and audiovisual learning [16]. As the Whisper paper details, partial data (audio-transcription pairs without timestamp information) makes up a substantial portion of the training corpus via use of a <|nottimestamps|> token. Scaling on partial and noisy transcription data naturally trades speech transcription performance for less accurate timestamp prediction. Therefore there is a need to correctly align the transcript and speech using additional modules.

A plethora of works exist on aligning speech transcription with audio waveforms at the word-level or phoneme-

The problems related to long form audio transcription with the current approaches

Why academic benchmarks are not good enough?

Approach applied in Whisper used

Why coupled decoding in Whisper is not good enough? And why is there a need for additional modules for accurate transcriptions

How forced alignment is typically implemented?

How merging is performed for different segments that are shorter than the upper bound?

level, so called "forced alignment". A traditional approach is to train the acoustic phone models in Hidden Markov Model (HMM) [17, 18, 19, 20] framework and using the by-product of possible state alignments. These word-level or phone-level timestamps are often adjusted by using external boundary correction models [21, 22]. With the rapid development in deep learning based methods, a few of recent works employ the deep learning strategies for forced alignment such as using bi-directional attention matrix [23] or CTC-segmentation with end-to-end trained model [24]. Further improvement could be made by leveraging a state-of-the-art ASR model with a lightweight phoneme recognition model, both of which are trained with substantial large-scale datasets.

To address these challenges, we propose *WhisperX*, a system for efficient speech transcription of long-form audio with accurate word-level timestamps. It consists of three additional stages to Whisper transcription: (i) pre-segmenting the input audio with an external Voice Activity Detection (VAD) model; (ii) cut and merging the resulting VAD segments into approximately 30 seconds input chunks with boundaries lying on minimally active speech regions enabling batched whisper transcription; and finally (iii) forced alignment with an external phoneme model to provide accurate word-level timestamps.

2. WhisperX

In this section we describe *WhisperX* and its components for long-form speech transcription with word-level alignment.

2.1. Voice Activity Detection

Before transcribing the input audio and performing alignment, we first pre-segment the audio with VAD. The benefits of this pre-processing stage are two-fold. First, VAD allows segmentation of the input audio into chunks with boundaries that do not lie on active speech regions, thereby minimising errors due to boundary effects and enabling batched transcription of chunks. Second, the temporal boundaries of each segment can be used to constrain the alignment to local segments and remove any reliance on Whisper's timestamps – which we show to be too unreliable when used for alignment.

Formally, given a long-form audio waveform A, we apply VAD, resulting in a list of N non-overlapping segments each corresponding to start and end times of active speech regions $\mathcal{S} = [S_1, S_2, ..., S_N]$, where $S_i = (t_0^i, t_1^i)$.

2.2. VAD Cut & Merge

VAD segments $\mathcal{S}$ can be of varying lengths, shorter or longer than the input duration that the ASR model (Whisper) was trained on, $|A_{\text{train}}|$ = 30 seconds. Whilst the transformer architecture can handle sequences of arbitrary input sizes, the attention operation scales with its the square of the input length and therefore long segments with no upper bound in duration can result in impractically high memory consumption.

To address this, we propose a min-cut operation whereby segments longer than the maximum input duration ($|A_{\text{train}}|$) are divided at the point with the *lowest voice activation score* from the VAD model. This ensures that the newly divided segments do not exist on word boundaries, minimizing boundary errors during the transcription process.

With an upper bound now set on the duration of input segments, the other extreme must be considered: very short segments, posing their own set of unique challenges. Transcribing short speech segment forgoes the wider context which can be beneficial for modelling speech in challenging settings. In addition, transcribing many shorter segments increases the time taken for transcription of the entire audio due to the additional number of forward passes.

Therefore, we propose the following merge operation:

$$S_{i+1} \begin{cases} (t_0^i, t_1^{i+1}), & \text{if } t_1^{i+1} - t_0^i < \tau \\ (t_0^{i+1}, t_1^{i+1}), & \text{otherwise} \end{cases}$$

where τ is the maximal duration of merged segments, which we show to be optimal at $\tau = |A_{\text{train}}|$. This essentially merges neighbouring short speech segments until their total duration is no greater than the input size of the transcription model, providing the greatest possible context when transcribing and keep the data distribution similar to that seen during training.

2.3. Whisper Transcription

The resulting speech segments, now with duration approximately equal to the input size of the model, $|S_i| \approx |A_{\text{train}}| \; \forall i \in N$, and boundaries that do not lie on active speech, can be effectively batch transcribed with Whisper, outputting text for each segment $S_i \rightarrow \mathcal{T}_i$. Transcription is performed without conditioning on previous text since this would break the independence assumption of each sample in the batch. Empirically, we find this to be beneficial for robust transcription since conditioning on previous text tends to be more prone to hallucination.

2.4. Forced Phoneme Alignment

For each segment S_i, and its resulting text transcription $\mathcal{T}_i$ consisting of a sequence of words $\mathcal{T}_i = [w_0, w_1, ..., w_m]$, our goal is to estimate the start and end time of each word. For this, we leverage a phoneme recognition model, trained to classify the smallest unit of speech distinguishing one word from another, *e.g. the element p in "tap"*. Let $C = [c_1, c_2, ..., c_K]$ be the set of phoneme classes in the dictionary. Given an input audio segment, a phoneme classifier, takes an audio segment S as input and outputs a logits matrix $L \in \mathbb{R}^{K \times T}$, where T varies depending on the temporal resolution of the phoneme model.

Formally, for each segment, $S_i \in \mathcal{S}$, and its corresponding text $\mathcal{T}_i$:

1. Perform phoneme classification over the input segment S_i, where classification is restricted to $C' = c_1, .., c_{K'}$ classes, the set of all phonemes in the current segment's transcription $\mathcal{T}_i$.
2. Apply Dynamic Time Warping (DTW) on the resulting logits matrix $L_i \in \mathbb{R}^{K' \times T}$, to obtain the optimal temporal path of phonemes in $\mathcal{T}_i$.
3. Obtain start and end times for each word w_i in $\mathcal{T}_i$ by taking the start and end time of the first and last phoneme respectively.

For transcript characters not present in the phoneme dictionary C, we assign the timestamp from the next nearest phoneme in the transcript. The for loop described aboved can be batch processed in parallel, enabling fast transcription and word-alignment of long-form audio.

2.5. Multilingual Transcription and Alignment

WhisperX can also be applied to multilingual transcription, with the caveat that (i) the VAD model should be robust to different languages and (ii) the alignment phoneme model ought to be trained on the language(s) of interest. Multilingual phoneme

VAD module, functionality, and its advantages

Why is there a need for cut and merge operation?

Why transcription conditioned on previous text is a bad idea?

How does the phoneme recognition model works?

recognition models [25] are also be suitable option, possibly generalising to languages not seen during training – this would just require an additional mapping from language-independent phonemes to the phonemes of the target language(s).

2.6. Word-level Timestamps without Phoneme Recognition

A natural question is whether word-level timestamps can be extracted from the Whisper model directly, without the addition of an external phoneme model. Such a method alleviates the additional inference overhead (although in practice we find this overhead is minimal, approx. $<$10%), nor requires a mapping between the Whisper and phoneme dictionaries.

With some tweaking of the forward pass, it is possible to infer word-level timestamps from the cross-attention scores of the decoded tokens. Attempts of this nature have been made by the original Whisper authors in their official open-source repository [26], as well as implementations by others [27]. However, as we show later in the results (Section 3.4), this Whisper-only method underperforms considerably compared to the proposed external phoneme approach and is prone to the aforementioned timestamp inaccuracies.

3. Evaluation

Our evaluation addresses the following questions: (1) the effectiveness of *WhisperX* for long-form transcription and word-level segmentation compared to state-of-the-art ASR models (namely Whisper and wav2vec2.0); (2) the benefit of VAD Cut & Merge pre-processing in terms of transcription quality and speed; and (3) the effect of the choice of phoneme model and whisper model on word segmentation performance.

3.1. Datasets

The AMI Meeting Corpus. We used the test set of the AMI-IHM from the AMI Meeting Corpus [28] consisting of 16 audio recordings of meetings, each approximately 30 minutes in duration. Manually verified word-level alignments are provided for the test set which we use to evaluate word segmentation performance.

Switchboard-1 Telephone Speech Corpus (SWB). SWB [29] consists of approximately 2,400 hours of speech of two-sided telephone conversations. Ground truth transcriptions are provided with manually correct word alignments. We randomly subsampled a set of 100 conversations to evaluate word segmentation compared to the manually verified word-level alignments.

TED-LIUM 3. To evaluate transcription quality and speed (no timestamps) of long-form audio, we follow [10] and report WER on the TEDLIUM test set [30] consisting of 11 TED talks each approximately 20 minutes in duration. We additionally report transcription speed.

3.2. Metrics

Speech recognition benchmarks typically only measure *word error rate (WER)* and do not evaluate the accuracy of the predicted timestamps. Thus, we also evaluate word segmentation metrics, for datasets that have word-level timestamps, that jointly evaluate transcription and timestamp quality. We report *Precision* and *Recall* where a true positive is where a predicted word segment overlaps with a ground truth word segment within a collar, where both words are an exact string match. For all evaluations we use a collar value of 200 milliseconds to account for differences in annotation and models.

Table 1: *Default configuration for WhisperX.*

Type	Hyperparameter	Default Value
VAD	Model	`pyannote`
	Onset threshold	0.767
	Offset threshold	0.377
	Min. duration on	0.136
	Min. duration off	0.067
Whisper	Model version	`large-v2`
	Decoding strategy	greedy
	Condition on previous text	False
Phoneme Recognition	Architecture	wav2vec2.0
	Model version	`BASE_960H`
	Decoding strategy	greedy

3.3. Implementation Details

WhisperX. Unless specified otherwise, we use the default configuration in Table 1 for all experiments.

Whisper [10]. For Whisper-only transcription and word-alignment we inherit the default configuration from Table 1, and infer the word-level timestamps from the cross-attention peaks in the decoded tokens (as in [26, 27]). Timestamp heuristics must be employed (including clamping negative duration timestamps) in order to prevent failed alignments.

Wav2vec2.0 [2]. For wav2vec2.0 transcription and word-alignment we use the default settings in Table 1 unless specified otherwise. We obtain the various model versions from the official torchaudio repository[2]. Base_960h, Large_960h and HuBERT [3] models were finetuned on Librispeech [31] data, whereas the VoxPopuli model was trained on the Voxpopuli [32] corpus.

For benchmarking inference speed, all models are measured on an NVIDIA A40 gpu.

3.4. Results

3.4.1. Word Segmentation Performance

We compare *WhisperX* to previous state-of-the-art works in speech transcription, namely Whisper and wav2vec2.0. In Table 2, we see that Whisper outperforms both wav2vec2.0 and Whisper by substantial margins on word segmentation benchmarks, as well as significant improvements to WER and transcription speed over Whisper. WhisperX with batched transcription is even faster than the lightweight wav2vec2 model. We see that mining word-level timestamps from Whisper alone underperforms considerably at word segmentation precision and recall across both SWB and AMI corpuses, even performing worse than wav2vec2.0, a smaller model trained on far less data. This suggests that the large-scale noisy training data of Whisper alone is insufficient to correctly learn word-level timestamps with the current architecture and training regime used.

[2] https://pytorch.org/audio/stable/pipelines.html#module-torchaudio.pipelines

Table 2: ***State-of-the-art comparison of long-form audio transcription and word segmentation*** *on the TEDLIUM, AMI, and Switchboard corpuses. WER denotes Word Error Rate. †Word-level timestamps are not obtainable out-of-the-box from Whisper and are inferred from the cross-attention scores of the decoded tokens via Dynamic Time Warping.*

Model	Version	TED-LIUM		AMI		SWB	
		WER↓	Speed↑	Precision↑	Recall↑	Precision↑	Recall↑
wav2vec2.0 [2]	BASE_960H	19.8	10.3×	81.8	45.5	92.9	54.3
Whisper [10]†	large-v2	10.5	1.0×	70.5	42.9	84.1	54.3
WhisperX	large-v2	**9.7**	**11.8**×	**84.1**	**60.3**	**93.2**	**65.4**

Table 3: ***Effect of WhisperX's VAD Cut & Merge and batched transcription on long-form audio transcription*** *on the TEDLIUM benchmark and AMI corpus. Full audio input corresponds to WhisperX without any VAD pre-processing, VAD-CM$_\tau$ refers to VAD pre-processing with Cut & Merge, where τ is the merge duration threshold in seconds.*

Input	Batch Size	TED-LIUM		AMI	
		WER↓	Spd.↑	Prec.↑	Rec.↑
Full audio	1	10.52	1.0×	82.6	53.4
	32	78.78	7.1×	43.2	25.7
VAD-CM$_{15}$	1	9.72	2.1×	84.1	56.0
	32		7.9×		
VAD-CM$_{30}$	1	**9.70**	2.7×	**84.1**	**60.3**
	32		**11.8**×		

Table 4: ***Effect of whisper model and phoneme model on WhisperX on word segmentation.*** *Both the choice of whisper and phoneme model has a significant effect on word segmentation performance.*

Whisper Model	Phoneme Model	AMI		SWB	
		Prec.	Rec.	Prec.	Rec.
base.en	Base_960h	83.7	58.9	93.1	64.5
	Large_960h	84.9	56.6	93.1	62.9
	HuBERT	83.6	58.5	**94.3**	**65.3**
	VoxPopuli	**87.4**	**60.3**	86.3	60.1
small.en	Base_960H	84.1	59.4	92.9	62.7
	Large_960H	84.6	55.7	**94.0**	**64.9**
	HuBERT	84.0	58.9	93.4	63.0
	VoxPopuli	**87.7**	**61.2**	84.7	56.3
large-v2	Base_960H	84.1	60.3	93.2	65.4
	Large_960H	84.9	57.1	**93.5**	**65.7**
	HuBERT	84.0	59.8	93.3	63.0
	VoxPopuli	**87.7**	**61.7**	84.9	58.7

3.4.2. *Effect of VAD Chunking*

Table 3 illustrates the benefits of pre-segmenting the audio with VAD and the Cut & Merge operations. Interestingly, there is an improvement on on the transcription-only WER metric, suggesting that VAD preprocessing for general transcription quality. The benefit of VAD is more pronounced on word segmentation precision and recall, indicating that whisper timestamps are not sufficient to bound the alignment window. Batched transcription without VAD chunking experiences a severe degradation in transcription quality (WER) and word segmentation.

Batched inference with VAD provides an almost twelve-fold speed increase with no performance loss, since each segment in the batch can be independently transcribed. This overcomes the limitations of buffered transcription as in [10]. We see that setting the merge threshold τ to the input duration that whisper was trained on $|A_{\text{train}}| = 30$, results in optimal transcription speed and the lowest WER, compared to lower merge thresholds such as $\tau = 15.0$, suggesting that the greatest amount of context provides the most accurate transcription.

3.4.3. *Effect of Chosen Whisper and Alignment Models*

We compare the effect of different Whisper and phoneme recognition models on word segmentation performance across the AMI and SWB corpuses in Table 4. Unsurprisingly, we see consistent improvements in both precision and recall when using a larger Whisper model. In contrast, the bigger phoneme model is not necessarily the best is instead more nuanced. The model trained on the VoxPopuli corpus significantly outperforms other models on AMI, suggesting that there is a higher degree of domain similarity between the two corpuses. The lack of consistent high performance from the large model suggests that increasing the amount of phoneme-supervised finetuning data would realise further gains. Overall the base model trained on LibriSpeech performs consistently well and can be concluded to be a suitable default choice of phoneme alignment for word-level alignment of Whisper.

4. Conclusion

To conclude, we introduce WhisperX, an efficient and time-accurate speech recognition system enabling parallelised and time-aligned transcription with Whisper utlizing phoneme alignment. We show that the proposed VAD Cut & Merge preprocessing reduces hallucination (WER) and enables within-audio batched transcription, resulting in a twelve-fold speed increase without sacrificing transcription quality. Further, we show that these transcribed VAD segments can be force aligned with phoneme models, providing accurate word-level segmentations with minimal inference overhead and resulting in time-accurate transcriptions benefitting a range of applications (subtitling, diarization etc.). A promising direction for future work is the training of a single-stage ASR system that can efficiently transcribe long-form audio with accurate timestamps.

Acknowledgement This research is funded by the EPSRC VisualAI EP/T028572/1 (M. Bain, T. Han, A. Zisserman), a Global Korea Scholarship (J. Huh) and the Royal Society Research Professorship RP\R1\191132 (A. Zisserman).

5. References

[1] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *ICASSP*. IEEE, 2019, pp. 7180–7184.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *NeurIPS*, vol. 33, pp. 12 449–12 460, 2020.

[3] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[4] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[5] J. Huh, H. S. Heo, J. Kang, S. Watanabe, and J. S. Chung, "Augmentation adversarial training for self-supervised speaker recognition," *arXiv preprint arXiv:2007.12085*, 2020.

[6] J. Kang, J. Huh, H. S. Heo, and J. S. Chung, "Augmentation adversarial training for self-supervised speaker representation learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1253–1262, 2022.

[7] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," *NeurIPS*, vol. 33, pp. 3846–3857, 2020.

[8] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proc. AAAI*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[9] K. Prajwal, L. Momeni, T. Afouras, and A. Zisserman, "Visual keyword spotting with attention," *arXiv preprint arXiv:2110.15957*, 2021.

[10] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[12] C.-C. Chiu, W. Han, Y. Zhang, R. Pang, S. Kishchenko, P. Nguyen, A. Narayanan, H. Liao, S. Zhang, A. Kannan *et al.*, "A comparison of end-to-end models for long-form speech recognition," in *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2019, pp. 889–896.

[13] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *ICASSP*. IEEE, 2018, pp. 5239–5243.

[14] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP*. IEEE, 2019, pp. 6301–6305.

[15] K. Prajwal, T. Afouras, and A. Zisserman, "Sub-word level lip reading with visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5162–5172.

[16] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2630–2640.

[17] F. Brugnara, D. Falavigna, and M. Omologo, "Automatic segmentation and labeling of speech based on hidden markov models," *Speech Communication*, vol. 12, no. 4, pp. 357–370, 1993.

[18] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics*, vol. 39, no. 3, pp. 192–193, 2011.

[19] J. Yuan, N. Ryant, M. Liberman, A. Stolcke, V. Mitra, and W. Wang, "Automatic phonetic segmentation using boundary models." in *Interspeech*, 2013, pp. 2306–2310.

[20] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[21] Y.-J. Kim and A. Conkie, "Automatic segmentation combining an hmm-based approach and spectral boundary correction," in *Seventh International conference on spoken language processing*, 2002.

[22] A. Stolcke, N. Ryant, V. Mitra, J. Yuan, W. Wang, and M. Liberman, "Highly accurate phonetic segmentation using boundary correction models and system fusion," in *ICASSP*. IEEE, 2014, pp. 5552–5556.

[23] J. Li, Y. Meng, Z. Wu, H. Meng, Q. Tian, Y. Wang, and Y. Wang, "Neufa: Neural network based end-to-end forced alignment with bidirectional attention mechanism," in *ICASSP*. IEEE, 2022, pp. 8007–8011.

[24] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *Speech and Computer (SPECOM 2020)*. Springer, 2020, pp. 267–278.

[25] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[26] J. W. Kim, "Whisper word-level timestamps," https://github.com/openai/whisper/blob/word-level-timestamps/notebooks/Multilingual_ASR.ipynb, 2022.

[27] J. Louradour, "whisper-timestamped," https://github.com/linto-ai/whisper-timestamped/tree/f861b2b19d158f3cbf4ce524f22c78cb471d6131, 2023.

[28] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Edinburgh, UK, July 11-13, 2005, Revised Selected Papers 2*. Springer, 2006, pp. 28–39.

[29] J. Godfrey and E. Holliman, "Switchboard-1 release 2 ldc97s62," *Linguistic Data Consortium*, p. 34, 1993.

[30] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*. Springer, 2018, pp. 198–208.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[32] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

Summary

Last year OpenAI released Whisper, a large-scale, weakly-supervised speech recognition model that demonstrated impressive results. Leveraging 680,000 hours of noisy speech training data, including 96 other languages and 125,000 hours of English translation data, they showcased that weakly supervised pretraining of a simple encoder-decoder transformer can robustly achieve zero-shot multilingual speech transcription on existing benchmarks.

One big difference between the datasets used for benchmarks and real-world data is that real-world applications typically require transcribing long-form audio that can easily be hours or minutes long, such as meetings, podcasts, and videos. Thus deploying a model like Whisper comes with two main constraints:

- Typical ASR models are trained on short audio segments, 30 seconds for the case of Whisper.
- The transformer architecture leveraged by these models prohibits transcription of arbitrarily long input audio due to memory constraints.

Whisper's coupled decoding of both the transcriptions and timestamps with a single encoder-decoder are prone to the usual challenges faced by auto-regressive language generation, namely:

- Hallucination
- Repetition

It has catastrophic consequences for buffered transcription of long-form and other timestamp-sensitive tasks such as speaker diarization, lip-reading, and audio-visual learning. Whisper uses such a buffered transcription approach. It relies on accurate timestamp prediction to determine the shift amount required for the input window.

Hence using models like Whisper for long-form transcriptions, we need to rethink the alignment of transcripts and speech. Lots of efforts have been made in the past to align speech transcription with audio waveforms at the word level or phoneme level, known as forced alignment. Some of these methods are:

- A traditional approach is to train the acoustic phone models in Hidden Markov Model (HMM) framework and use the by-product of possible state alignments. These word-level or phone-level timestamps are adjusted by using external boundary correction models.
- Force alignment using a bi-directional attention matrix or CTC segmentation with an end-to-end trained deep learning-based model.

In this paper, the authors propose WhisperX, a system for efficient speech transcription of long-form audio with accurate word-level timestamps. WhisperX consists of three additional stages to Whisper transcription:

- Pre-segmentation of audio input using a Voice Activity Detection model
- Cut and Merge the resulting VAD segments into approximately 30 seconds input chunks with boundaries on minimally active speech regions to enable batched whisper transcription.
- Forced alignment with an external phoneme model to provide accurate word-level timestamps

There are a total of four steps that completes the workflow of WhisperX.

- Voice Activity Detection
  - Pre-segment audio with VAD
  - Given a long-form audio waveform A, apply VAD. This results in a list of N non-overlapping segments, each corresponding to the start and end times of active speech regions S = [S1, S2, ..., SN]

  It has two advantages:
  - VAD allows the segmentation of the input audio into chunks with boundaries that do not lie on active speech regions, thereby minimizing errors due to boundary effects. It also enables the batched transcription of the resulting chunks.
  - The temporal boundaries of each segment can be used to constrain the alignment to local segments and remove any reliance on Whisper's timestamp.

- Cut and Merge
  - Whisper was trained on segments of |Atrain|=30 seconds in length. Also, attention scales quadratically with the input length. Therefore we need a way to limit memory consumption. The authors propose a min-cut operation whereby segments longer than the maximum input duration (|Atrain|= 30 seconds) are divided at the point with the lowest voice activation score from the VAD model. It ensures that the newly divided segments do not exist on word boundaries, minimizing boundary errors during the transcription process. This proposal fixes the upper bound for the audio segments.
  - Transcribing many shorter segments increases the time taken to transcribe the entire audio due to the additional number of forward passes. To address this, the authors propose to repeatedly merge two neighboring segments unless the upper bound for the merged segment(30 seconds in this case) is reached

- Whisper Transcription:
  - Transcription is performed without conditioning on previous text since this would break the independence assumption of each sample in the batch that can lead to hallucinations otherwise

- Forced Phoneme Alignment: For each segment Si, we have the transcription Ti. The goal at this stage is to find the start and end times for each word. For this, the authors leverage a phoneme classification model. For each segment Si, and corresponding transcription Ti:
    - Perform phoneme classification over the input segment Si
    - Apply Dynamic Time Warping (DTW) on the resulting logits matrix Li $\in$ R^(K'×T) to obtain the optimal temporal path of phonemes in Ti.
    - Obtain start and end times for each word wi in Ti by taking the start and end times of the first and last phonemes respectively. For transcript characters not present in the phoneme dictionary C, the authors assign the timestamp from the next nearest phoneme in the transcript.