LiveFC

A System for Live Fact-Checking of Audio Streams

Venktesh, V.; Setty, Vinay

DOI
10.1145/3701551.3704128

Publication date
2025

Document Version
Final published version

Published in
WSDM 2025

LiveFC: A System for Live Fact-Checking of Audio Streams

Venktesh V
Delft University of Technology
Delft, Netherlands
v.viswanathan-1@tudelft.nl

Vinay Setty
University of Stavanger, Factiverse AI
Stavanger, Norway
vsetty@acm.org

Abstract

The advances in the digital era have led to rapid dissemination of information. This has also aggravated the spread of misinformation and disinformation. This has potentially serious consequences, such as civil unrest. While fact-checking aims to combat this, manual fact-checking is cumbersome and not scalable. While automated fact-checking approaches exist, they do not operate in real-time and do not always account for spread of misinformation through different modalities. This is particularly important as proactive fact-checking on live streams in real-time can help people be informed of false narratives and prevent catastrophic consequences that may cause civil unrest. This is particularly relevant with the rapid dissemination of information through video on social media platforms or other streams like political rallies and debates. Hence, in this work we develop a platform named LiveFC, that can aid in fact-checking live audio streams in real-time. LiveFC has a user-friendly interface that displays the claims detected along with their veracity and evidence for live streams with associated speakers for claims from respective segments. The app can be accessed at http://livefc.factiverse.ai and a screen recording of the demo can be found at https://bit.ly/3WVAoIw.

CCS Concepts

• **Computing methodologies** → **Natural language processing**.

Keywords

Live Fact-checking, Claim Decomposition

ACM Reference Format:
Venktesh V and Vinay Setty. 2025. LiveFC: A System for Live Fact-Checking of Audio Streams. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), March 10–14, 2025, Hannover, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3701551.3704128

1 Introduction

The rapid proliferation of misinformation and disinformation in the digital era has lasting impacts on society, politics, and the shaping of public opinion. as manual fact.checking is cumbersome, automated fact-checking approaches have been proposed [5, 11] which has made tremendous advances. Majority of the existing automated fact-checking approaches are primarily focused on textual modality [7, 9]. However, real-world misinformation and disinformation can be spread through multiple possible modalities, such as audio, video, and images [1, 16] and has higher engagement and spreads faster than text only content [8]. Hence, it is crucial to fact-check multi-modal content.

Misinformation spread through multi-modal content, such as political debates, interviews, and election campaigns, is time-critical due to its potential to sway public opinion and its perceived reliability [10]. Manual fact-checking is cumbersome and time-consuming, and existing automated tools focus on post-hoc verification, which is ineffective against rapidly spreading misinformation. To address this, we developed LiveFC, a tool that transcribes, diarizes speakers, and fact-checks spoken content in live audio streams in real-time (within seconds), targeting misinformation at its source. While focused on live events like election debates and campaign rallies, LiveFC also works with long-form offline content such as parliament discussions, interviews, and podcasts. Fact-checkers and news reporters find LiveFC particularly useful for detecting and verifying claims in real-time. This was validated in a pilot study with Danish fact-checkers Tjekdet[1] during the European Parliament election in June 2024, where the tool helped catch important claims that would otherwise have been missed.[2] Additionally, we conducted a case study of the first US presidential debate of 2024, comparing manual fact-checks from the Politifact with those done by LiveFC.

Recent advances in automatic speech recognition (ASR) models, like Whisper by OpenAI [12], have significantly improved audio transcription quality. Existing solutions like Pyannote for speaker diarization [4] also perform well, but mainly for offline content. Real-time transcription and speaker diarization for live content pose unique challenges. To enable live fact-checking, we need to transcribe and identify speakers in smaller segments of the audio stream and align speakers with the transcribed text. This demo system showcases techniques to extend Whisper and Pyannote for live-streaming applications.

LiveFC architecture is depicted in Figure 1 which has 6 key components: 1) A transcriber module that can operate on streaming data to transform live audio streams to text, 2) A diarization module that identifies the speaker for the audio segments. 3) A claim detection and normalization module that identifies check worthy claims from the transcribed segments in real-time 4) A claim decomposition and topic assignment module that aids in decomposing claims to questions for reasoning which renders the fact-checking process explainable and assigns a broad set of topics for analysis of the fact-checks 5) An evidence retrieval module that retrieves up-to-date evidence from the web search and past fact-checks and a 6) claim verification component that employs state-of-the-art fine-tuned Natural Language Inference (NLIs) models.

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3704128

[1] https://tjekdet.dk
[2] https://factiverse.ai/live

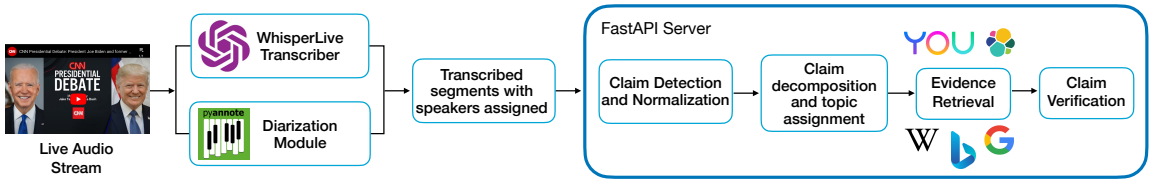

Figure 1: LiveFC pipeline for fact-checking live audio streams like political debates.

Our key contribution is that the entire pipeline operates in a real-time manner using efficient and effective quantized models. We posit that this would aid fact-checkers in curbing the spread of misinformation at the source without delay.

2 System Design

An overview of our live fact-checking pipeline LiveFC is shown in Figure 1. The live audio stream is provided as input to a speaker diarization module and transcription module in parallel. This is followed by a mapping phase where the transcribed segments are mapped to respective speakers based on timestamps and other meta-data. The resulting transcribed segments are then sent to a claim identification module followed by claim decomposition, evidence retrieval and claim verification. The pipeline is hosted using a Python FastAPI backend. The frontend is implemented using the Streamlit framework.[3].

2.1 Transcription of Live Audio Stream

We adapt the Whisper Live[4] implementation for our fact-checking pipeline. We use the whisper-large-v3 model, a sequence-to-sequence model pre-trained on a large amount of weakly supervised (audio, transcript) pairs, which directly produces raw transcripts. We process the audio stream in segments to support HLS (HTTP Live Streaming), which is then buffered and transmitted to the transcription client via the FFmpeg encoder.[5] Unlike traditional systems, Whisper Live employs Voice Activity Detection (VAD) to send data to Whisper only when speech is detected, making the process more efficient and producing high-quality transcripts.

2.2 Online Diarization Module

For attribution and offline analysis, linking claims to the corresponding speaker is essential. Our diarization module performs real-time speaker identification, known as online speaker diarization with limited context. LiveFC employs an overlap-aware online diarization approach [3], involving speaker segmentation and clustering. We adapt the diart module[6] for our use, utilizing websockets to stream audio content.

The audio stream is sent via websocket to the diarization server, where it undergoes speaker segmentation using a neural network. Every 500ms, the server processes a 5-second rolling audio buffer and outputs speaker active probabilities $A = s_1 ... s_n$, where n is the number of frames. Speakers with an active probability above a tunable threshold τ_{active} are identified, while inactive speakers are discarded. This approach effectively handles overlapping speakers, making it ideal for live fact-checking of debates. We set $\tau_{active} = 0.65$ to reduce false positives.

The segmentation model's permutation invariance means a speaker may not be consistently assigned the same speaker ID over time. To address this, we use incremental clustering to track speakers throughout the audio stream. Initially, speaker embeddings are created after segmentation for the first buffer, forming a centroid matrix C. As the rolling buffer updates, local speaker embeddings $(se_1..se_l)$ are compared to the centroids to assign them using an optimal mapping $(m*)$:

$$m* = \arg\min_{m \in M} \sum_{i=1}^{l} d(m(i), se_i)$$

, where M is the set of mapping functions between local speakers and centroids, with the constraint that two local speakers cannot be assigned to the same centroid. If the distance between a local speaker embedding and all centroids exceeds a threshold Δ_{new}, a new centroid is created. We set $\Delta_{new} = 0.75$ to balance sensitivity, avoiding the misclassification of slight tone changes as new speakers, while ensuring new speakers are accurately identified.

Speaker IDs are mapped to transcript segments using timestamps from diarization and transcription components, run in parallel for efficiency. We use *pyannote/embedding* computing embeddings and the *pyannote/segmentation-3.0* model for segmentation.

Split	NC	C	True	False	Total
Train	609	548	332	196	1,076
Dev	38	25	15	10	63
Test	62	38	26	12	100

Table 1: Dataset distribution for check-worthy claim detection. NC - Not Check-worthy, C - Checkworthy

2.3 Check-Worthy Claim Detection Module

The function of this component is to identify claims from transcribed segments that warrant verification.

Sentence segmentation and Claim Normalization: We first segment the transcription text into sentences using the Spacy library due to its speed and accuracy. Since speech segments may contain implicit references, we transform the sentences to make them self-contained by resolving co-references and removing any unwanted text from the spoken content. This is performed through a generative LLM (Mistral-7b). We term this step as *claim normalization*, as it yields self-contained candidate claims.

The self-contained candidate claims are then passed through a check-worthy claim detection model. We fine-tune a XLM-RoBERTa-Large model, using datasets from ClaimBuster and CLEF CheckThat Lab! [2] along with a dataset collected from Factiverse production system (see Table 1) to classify sentences into 'Check-worthy' and 'Not check-worthy'. We also assign a set of topics to the claims.

[3]https://streamlit.io
[4]https://github.com/collabora/WhisperLive
[5]https://www.ffmpeg.org
[6]https://github.com/juanmc2005/diart

2.4 Claim Decomposition and Evidence Retrieval

Prompt: Claim Decomposition

Instruction: Transform the given claim into questions to verify the veracity of the given claim and find the potential correct facts from search engines. The questions must be cover all aspects of the claim. Generate exactly num_questions questions for the claim and prefix the questions with Q̈uestion number:ẅithout making any references to the claim. Here are some **examples**:
Examples: Claim: "Kelvin Hopins was suspended from the Labor Party due to his membership in the Conservative Party."
Question 1: Was Kelvin Hopins suspended from Labor Party?
Question 2: Why was Kelvin Hopins suspended ...**Claim**: {claim}

Figure 2: Prompt for claim decomposition

The main goal is this component is to retrieve high quality evidence for verifying the check-worthy claims from the previous step. Fact-checking is not a linear process and involves multi-step reasoning, where fact-checkers synthesize diverse queries and search the web and other knowledge sources to gather multiple perspectives and evidence to verify a claim. To emulate the process of fact-checkers, we employ a claim decomposition module where we prompt a LLM (Mistral-7b) as shown in Figure 2.

Following the decomposition step, we retrieve evidence from diverse sources such as Google, Bing, Wikipedia, You.com, Semantic Scholar (contains 212M scholarly articles). Since some claims might be duplicates or similar to existing fact-checked claims, we also search our ElasticSearch index, which houses Factiverse's fact-checking collection named **FactiSearch**, which comprises 280K fact-checks updated in real-time to retrieve related evidence. We filter out evidence from fact-checking sites and deduplicate evidence using meta-data like url, titles and approximate matching of content. We then employ a multilingual cross-encoder model [13] (*nreimers/mmarco-mMiniLMv2-L12-H384-v1*) to rank the evidences.

2.5 Claim Verification

Using the ranked evidences, we perform claim verification by formulating the task as a Natural language Inference (NLI) problem. The NLI task involves categorizing whether a claim is supported, refuted by a given piece of evidence or evidence is unrelated to the claim. We cast this problem to a binary classification task of predicting supported or refuted as we filter out unrelated evidence in the ranking step. We fine-tune an *XLM-Roberta-Large* model from Huggingface on combined data from FEVER [14], MNLI [15], X-fact [6] and our collection of real-world fact-checks in **FactiSearch**. Since each claim has multiple relevant evidence snippets, the NLI model is applied to claim and evidence in a pairwise manner followed by a majority voting phase to obtain the final verdict. We also summarize the evidence snippets providing justification for the verdict to the user to foster trust in the system.

3 Performance Evaluation

3.1 Offline Evaluation of Claim Detection and Verification Components

For offline evaluation of individual components of LiveFC pipeline, we employ the dataset collected from production environment of Factiverse. The statistics of the dataset, are shown in Table 1. We observe that our fine-tuned XLM-Roberta model outperforms LLM based approaches for tasks of claim detection and verification. We primarily observe that in claim verification, LLMs underperform when compared to smaller fine-tuned models due to their inability to reason and extract required information from evidence and due to hallucination. Hence, we employ our fine-tuned model as part of the pipeline in the LiveFC tool.

EC (α_K)	**EU** (α_K)	**TR** (α_K)
3.46±1.49 (0.76)	3.60±1.39 (0.65)	4.37±1.12 (0.51)

Table 4: Manual evaluation metrics (Likert 1–5) with Krippendorff's alpha (α_K): evidence completeness, usefulness, and topic relevance.

Model	**Claim Detection**		**Veracity Prediction**	
	Ma.-F1	**Mi.-F1**	**Ma.-F1**	**Mi.-F1**
Mistral-7b	0.590	0.600	0.526	0.527
GPT-3.5-Turbo	0.607	0.625	0.605	0.605
GPT-4	0.695	0.701	0.630	0.632
Ours	**0.899**	**0.900**	**0.708**	**0.737**

Table 5: Claim detection and verification results for English data.

3.2 End to End Evaluation on Live Stream

We also evaluate LiveFC on the first presidential debate of 2024. A screenshot of the tool is shown in Figure 3.

Debate Statistics: We report the statistics obtained from live fact-checking of the debate through our tool LiveFC. The number of supported and disputed claims made by each speaker is shown in Table 2 and topicwise distribution of claims are shown in Table 3. We observe that topic related to *War and Defense* was the most discussed during the debate. The plot shows the distribution of claims across 7 key topics, and the rest of claims that do not fall into any of these topics are categorized as "Other" and are not shown in the graph. We also observe that there are a significant number of disputed claims made by the speakers, which highlights the significance of live fact-checking. We also display the evidence and summarize the justification for the veracity label, rendering the process more transparent to the end user.

Comparison based evaluation of claims identified and veracity prediction to Politifact: We compare the claims identified and corresponding predicted labels from manual fact-checker Politifact to those identified and verified by our tool LiveFC for the 2024 presidential debate. We observed that we were able to identify all the 30 claims identified by Politifact. We were further able to identify more claims not covered by Politifact which highlights the advantages of automated fact-checking. However, we also acknowledge that some of the claims we identify are false positives and may not be significant enough, which is removed by us in post-processing phase. When comparing the veracity labels with Politifact for the 30 claims, we observe a macro P, R and F1 scores of **82.59, 85.78** and **83.92** respectively and weighted F1 of **87.26**.

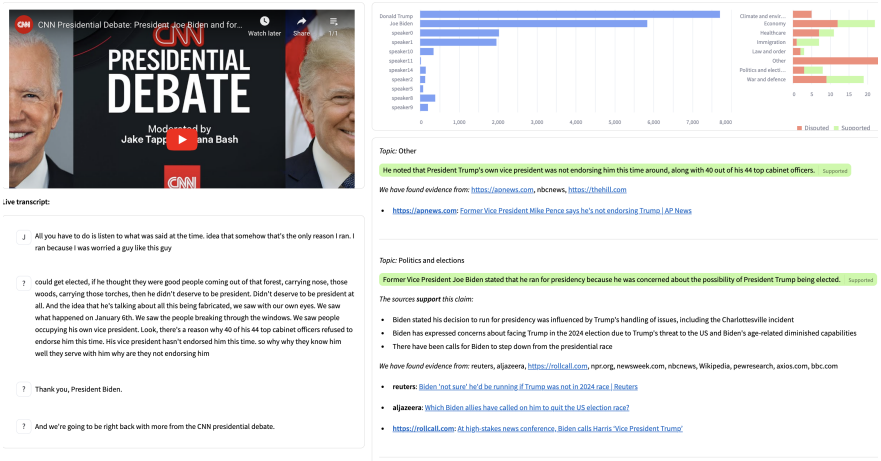

Figure 3: A screenshot of LiveFC UI.

Speaker	Supported	Disputed	Total
Trump	147	205	352
Biden	169	170	339

Table 2: Stats from fact-checks of 2024 debate

Category	Biden	Trump
Defense	42	70
Economy	35	28
Politics	25	31
Climate	20	23
Immigration	15	20
Law	9	14
Healthcare	16	7

Table 3: Number of claims by Biden and Trump

Qualitative evaluation of evidence utility and topic assignments: We perform a qualitative evaluation of fact-checks performed on 2024 US presidential debate live-stream by sampling 20 claims with retrieved evidence, topic assigned and veracity predictions using our tool LiveFC. We requested three annotators with background in automated fact-checking to rate the samples on three factors such as evidence usefulness, evidence completeness and topic relevance on Likert scale (1-5). The average ratings across annotators with inter-annotator agreement are shown in **Table 4**.

4 Conclusion

This paper presents the LiveFC system, an end to end approach for real-time fact-checking which employs efficient, effective and smaller models. We applied it to the live stream of 2024 political debate and observed that it was able to detect and verify facts in real-time. We conducted offline evaluation of different core components of the system using fact-checking benchmarks. We also conducted manual and qualitative evaluation of fact-checks generated from debate and observed that the system was able to detect all claims detected by manual fact-checkers and also retrieve useful evidence for accurate verification of claims. In the future, we plan to further extend LiveFC to handle multi-modal evidence sources.

5 Acknowledgements

This work is partly funded by the Research Council of Norway project EXPLAIN (grant no: 337133). We acknowledge valuable contributions from Factiverse AI team: Erik Martin for backend, Tobias Tykvart for Frontend, Sowmya AS for UI, Henrik Vatndal for diarization module and others for manual analysis (Maria Amelie, Gaute Kokkvol, Sean Jacob, Christina Monets and Mari Holand)

References

[1] Mubashara Akhtar, Michael Schlichtkrull, Zhijiang Guo, Oana Cocarascu, Elena Simperl, and Andreas Vlachos. 2023. Multimodal Automated Fact-Checking: A Survey. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Singapore.

[2] Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish, Abdulaziz Al-Homaid, Wajdi Zaghouani, Tommaso Caselli, Gijs Danoe, Friso Stolk, Britt Bruntink, and Preslav Nakov. 2021. Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society. In *Findings: EMNLP 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.).

[3] Hervé Bredin and Antoine Laurent. 2021. End-to-end speaker segmentation for overlap-aware resegmentation. arXiv:2104.04045 [eess.AS]

[4] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. 2020. Pyannote. audio: neural building blocks for speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7124–7128.

[5] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics* 10 (2022).

[6] Ashim Gupta and Vivek Srikumar. [n. d.]. X-Fact: A New Benchmark Dataset for Multilingual Fact Checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.).

[7] Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting Check-worthy Factual Claims in Presidential Debates *(CIKM '15)*. Association for Computing Machinery.

[8] Yiyi Li and Ying Xie. 2020. Is a Picture Worth a Thousand Words? An Empirical Study of Image Content and Social Media Engagement. *Journal of Marketing Research* 57 (2020). arXiv:https://doi.org/10.1177/0022243719881113

[9] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated Fact-Checking for Assisting Human Fact-Checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization. Survey Track.

[10] Eryn Newman, Maryanne Garry, Daniel Bernstein, Justin Kantner, and D Lindsay. 2012. Nonprobative photographs (or words) inflate truthiness. *Psychonomic bulletin and review* 19 (2012).

[11] Andreas L Opdahl, Bjørnar Tessem, Duc-Tien Dang-Nguyen, Enrico Motta, Vinay Setty, Eivind Throndsen, Are Tverberg, and Christoph Trattner. 2023. Trustworthy journalism through AI. *Data and Knowledge Engineering* 146 (2023).

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

[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL]

[14] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the NAACL-HLT, Volume 1 (Long Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.).

[15] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana.

[16] Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. 2023. End-to-End Multimodal Fact-Checking and Explanation Generation: A Challenging Dataset and Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery.