SpeechEE@XLLM25: End-to-End Structured Event Extraction from Speech

Soham Chaudhuri[1], **Diganta Biswas**[2], **Dipanjan Saha**[3], **Dipankar Das**[4], **Sivaji Bandyopadhyay**[5]

[1]Dept. of EE, Jadavpur University, Kolkata, India

[2,3,4,5]Dept. of CSE, Jadavpur University, Kolkata, India

{sohamchaudhuri.12.a.38, biswasdiganta2001, sahadipanjan6, dipankar.dipnil2005, sivaji.cse.ju}@gmail.com

Abstract

Event extraction from text is a complex task that involves the identification of event triggers and their supporting arguments. When applied to speech, this task becomes even more challenging due to the continuous nature of audio signals and the need for robust Automatic Speech Recognition (ASR). This paper proposes an approach that integrates ASR with event extraction by utilizing the Whisper model for speech recognition and a Text2Event2 Transformer for extracting events from English audio samples. The Whisper model is used to generate transcripts from audio, which are then fed into the Text2Event2 Transformer to identify event triggers and their arguments. This approach combines two difficult tasks into one, streamlining the process of extracting structured event information directly from audio. Our approach leverages a robust ASR system (Whisper) followed by a parameter-efficient transformer (Text2Event2 fine-tuned via LoRA) to extract structured events from raw speech. Unlike prior work trained on gold textual input, our pipeline is trained end-to-end on noisy ASR outputs. Despite significant resource constraints and data noise, our system ranked first in the ACL 2025 XLLM Shared Task II.

1 Introduction

Event extraction from speech audio samples poses a challenge as the shortcomings of ASR like noise, substitution errors, hallucinations and other errors get propagated to the event extraction transformer leading to erroneous training dataset leading to the transformer learning from an erroneous data. Unlike previous work that utilizes clean, curated textual data for event extraction, we address the more realistic and challenging scenario of extracting structured events directly from raw audio input (Fei et al., 2024). This introduces transcription noise, alignment challenges, and limited supervision, requiring novel techniques to ensure generalizability and robustness.

We have streamlined the process of event extraction from English audio samples with **WiSE** (Whiper-to-Structured-Events) which utilizes Whisper-medium[1] developed by OpenAI[2] and fine-tuned Text2Event2 transformer model as in (Wang et al., 2024). The audio before passing through the whisper-medium model and tokenizer is processed to convert to a frequency of 16kHz and monophonic channel audio samples. It is done to bring the audio samples to the same reference frame for better transcript generation which will lead to better event extraction and can also use multimodal LLMs like (Wu et al., 2024).

Loudness of the audio samples is also standardized to bring them to the same reference frame. Then the audio samples are passed through the Whisper-medium model and transcripts are generated. This transcripts of train and development dataset in combination with the labelled events of the audio files was used to fine-tune the Text2Event2 transformer model for better accustomed with our scenario.

2 Dataset Description

Dataset was provided to us for a shared task organised by **XLLM** in collaboration with ACL in 2025. This dataset is specifically from the shared task II: Speech Event Extraction (SpeechEE). The dataset contains **33 event types** and **22 argument roles**, with **19217 training data**, **901 validation data** and **676 testing data**. The data was given to us inform of english audio samples. In addition to this, we were also given a detailed event schema in the form of json which included all the event types and the argument types to support a particular event.

[1] https://huggingface.co/openai/whisper-medium

[2] https://openai.com/

Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025), pages 283–287
August 1, 2025 ©2025 Association for Computational Linguistics

```
{
 "Start-Org": ["Agent", "Org", "Place"],
 "Marry": ["Person", "Place"],
"Start-Position": ["Person", "Entity", "Place"],
 "Acquit": ["Adjudicator", "Defendant"],
 "Meet": ["Entity", "Place"],
 "Merge-Org": ["Org"],
}
```

Figure 1: Schema of an event

Although the data set contains over $20,000$ audio samples, only $3,669$ samples had tagged events and the corresponding arguments in the combined training and development sets. This limited annotated data presents a significant challenge in training robust models. To illustrate the structure of the data, a sample transcript of "train-3.wav" is provided in "train.json" along with its tagged event. This annotation includes detailed information on the event triggers and their respective arguments, which are crucial to fine-tuning the Text2Event2 Transformer model to accurately extract events from the transcripts generated by the Whisper ASR system. The scarcity of annotated data highlights the need for efficient use of available resources and innovative strategies to improve model performance.

Transcript:
Even as the Secretary of Homeland Security was putting his people on high alert last month, a 30-foot Cuban patrol boat with four heavily armed men landed on American shores. Underly undetected by the Coast Guard Secretary Ridge now leads.

```
{
    "trigger": "landed", "type": "Transport",
     "arguments": [
         {"name": "boat", "role": "Vehicle"},
         {"name": "men", "role": "Artifact"},
      {"name": "shores", "role": "Destination"}
            ]
}
```

Figure 2: An example of events and their arguments

The training set and development set was combined and created into one dataset since the labeled dataset was so limited. A small set is kept aside for testing. It is important to note that although the dataset mirrors the ACE05EN schema, no gold transcripts were provided. All training data was supplied as raw English audio, requiring the construction of training data via ASR-generated transcripts. This modality shift introduces significant transcription noise, necessitating event extraction models that are robust to imperfect input.

3 Methodology

Automatic speech recognition also known as ASR is used to convert human speech to readable text. It has grown quite recently and is being used in various fields where human speech need fast transcriptions like live caption generation and live translation from one language to another language. This all requires speech recognition and speech-to-text conversion models. Whisper by OpenAI (Radford et al., 2022) is a state-of-the-art ASR model trained on $6,80,000$ hours of multilingual and multitasked supervised data. Training on this vast dataset has made the model robust to background noise, accents, and various languages.

We have used the whisper-medium model to generate the transcripts of the training set and the development set and created into a pandas data frame.

File Name	**Transcription**
train-10589.wav	Oh, uh-huh.
train-18281.wav	And now just so...
train-6191.wav	At the time...
train-140.wav	And the Democrats...
train-12985.wav	Tom Racings
train-11948.wav	I don't know.
train-2803.wav	It would talk about tips....
train-463.wav	I did not feel less than
train-2041.wav	They got to understand.
train-2815.wav	Famed World War II...

Table 1: Transcripts generated by **whisper-medium** model

3.1 BERTag

Transcripts and their respective event triggers and arguments are aligned and passed on to a BERT model previously fine-tuned for named entity recognition (NER). The BERT-base-NER[3] model was previously fine-tuned for BIO-tagged NERs. It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC). So to make it more aligned with our event schema we used an external label list for our event trigger and used **label2id** and **id2label** functions to map the event to new labels and vice versa. Then a tokenized data set was created for each training and validation set.

[3] https://huggingface.co/dslim/bert-base-NER

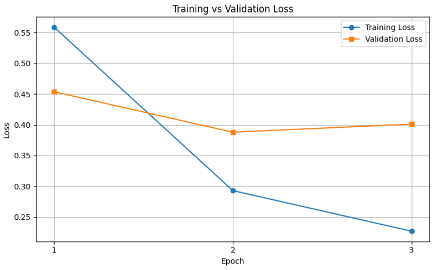

Figure 3: Performance of the model across epochs: Training Loss, Validation Loss, and Accuracy

We fine-tune the BERT[4] model using the HuggingFace `Trainer` API. The model is trained for *3* epochs with a batch size of *8* in both the training and evaluation datasets. We employ the **epoch** strategy for both evaluation and checkpoint saving. The best model is selected based on `eval_loss`, using `load_best_model_at_end=True` and `greater_is_better=False`. Logging is performed in every *10* step, and a maximum of *2* checkpoints are retained to limit storage. The `DataCollatorForTokenClassification` is used with the BERT tokenizer to handle dynamic padding. For evaluation, we report the accuracy at the token level, excluding padding tokens (label `-100`).

We use a simple token-level *accuracy* metric for evaluation. Model predictions are first reduced using *argmax* over the class dimension. For fairness, tokens labeled with *-100* (used to mask padding or special tokens) are excluded from both predictions and ground-truth labels. Accuracy is computed as the proportion of correctly predicted tokens over all valid (non-masked) tokens.

3.2 T2E2

Text2Event an end-to-end sequence to structure generation paradigm as proposed by (Lu et al., 2021). This model uses `google/t5-large`[5] model. Currently, most of the NER tasks use the decomposition method of diving the given sequence into multiple subtasks and then correlating the triggers with their specific arguments based on event schema. Text2Event was trained on **ACE05EN** dataset where the input is a linearized format for the encoder to encode and a trie-based decoder so that the outputs follow strictly the event schema.

We utilize the `BurgerTruck/text2event2` checkpoint based on a pretrained Transformer model for sequence-to-sequence learning. The tokenizer is initialized using `AutoTokenizer`, and the model is loaded via `AutoModelForSeq2SeqLM` with `load_in_8bit=False` and `device_map="cpu"` for CPU-based execution. For GPU acceleration, the model can be deployed with `load_in_4bit=True` and `device_map="auto"` to enable QLoRA training on low-memory GPUs. We employ the PEFT (Parameter-Efficient Fine-Tuning) framework and apply LoRA (Low-Rank Adaptation) (Hu et al., 2022). The model is first prepared with `prepare_model_for_kbit_training`, followed by a `LoraConfig` with rank *r=4*, scaling factor `lora_alpha=16`, dropout `lora_dropout=0.1`, and targeting the "q" and "v" attention modules. The final model is wrapped using `get_peft_model` for fine-tuning under the "SEQ_2_SEQ_LM" task type.

To accommodate the constraints of limited GPU access, we adopted a parameter-efficient fine-tuning (PEFT) approach using Low-Rank Adaptation (LoRA). This allowed us to fine-tune the Text2Event2 model entirely on CPU while maintaining performance. We applied LoRA to the attention layers of a T5-based sequence-to-sequence transformer, achieving competitive accuracy under extreme resource limitations.

The dataset had only 3669 labelled event samples so we used the whole set for fine-tuning and tested the model by generating outputs for the test set and scored it on the evaluating platform. Even though the dataset given to us closely resembled **ACE05EN** and Text2Event was trained on it, fine-tuning was necessary as the transcripts of ASR by whisper might be able to generate **ACE05EN** equivalent input sentences.

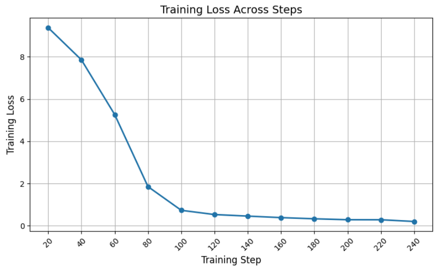

Figure 4: Training loss across different training steps.

4 Results and Discussion

To assess the effectiveness of event extraction models, organisers adopted a multi-task evaluation

[4] https://huggingface.co/google-bert/bert-base-uncased
[5] https://huggingface.co/google-t5/t5-large

framework comprising three subtasks. Each task evaluates different aspects of event structure and prediction quality. The evaluation metric for each task is the **F1-score**, computed from precision and recall. The final score is a weighted combination of the three task-specific F1-scores using the formula:

$$\boxed{\begin{aligned}\text{Overall Score} &= 0.3 \times \text{Task } 1_{F1} + 0.3 \times \text{Task } 2_{F1} \\ &+ 0.4 \times \text{Task } 3_{F1}\end{aligned}}$$

This weighting reflects the relative importance of each task in capturing comprehensive event understanding.

4.1 BERTag

BERT-base model which is downstreamed for NER tasks was not able to perform upto the mark as it was trained for BIO-tagged NERs. Our event schema being so extent, fine-tuning for such a small dataset and small number of epochs was not sufficient.

Task	Precision (%)	Recall (%)	F1-score (%)
Task 1	16.15	22.41	18.77
Task 2	3.14	5.08	3.88
Task 3	3.05	4.93	3.77
Overall Score	–	–	**8.31**

Table 2: Evaluation results across tasks and final weighted score.

Table 2 presents the precision, recall, and F1-score for each task, with the final overall score computed as a weighted sum of the individual F1-scores, resulting in an overall performance of **8.31%**.

4.2 T2E2

Text2Event2 is trained on ACE05EN which has similar event schema to the schema provided to us. Fine-tuning it to the transcripts of the whisper-medium makes it a little bit more robust to halucinations and errors of ASR.

Task	Precision (%)	Recall (%)	F1-score (%)
Task 1	64.5390	64.3868	64.4628
Task 2	37.0787	38.3164	37.6874
Task 3	34.4101	35.5588	34.9750
Overall Score	–	–	**44.6356**

Table 3: Evaluation results across tasks and final weighted score.

Table 3 shows the performance of the proposed model across all tasks, achieving an overall F1-score of **44.6356%**, calculated using the weighted combination of individual task scores. While the original Text2Event model reports an F1 score of approximately 72% on clean ACE05EN text, our model was evaluated on noisy ASR transcripts generated from the audio-only dataset. This challenging setup, combined with CPU-based training and a limited number of labeled samples, resulted in a top performance of 44.63% F1 in the shared task—demonstrating the effectiveness and robustness of our system.

This impressive overall score of **44.6356%** enabled us to secure **Rank 1** in the *Speech-to-Event Extraction Shared Task*, demonstrating the effectiveness of our proposed approach across all evaluation metrics.

5 Limitations

A major limitation in our pipeline stems from the use of ASR-generated transcripts without access to gold textual input. Whisper, while state-of-the-art, may hallucinate or omit important information, which gets propagated into the event extraction phase. Furthermore, due to the exhaustion of GPU quotas on Kaggle, the majority of training was conducted on CPU using LoRA, which limited the number of training epochs and speed of experimentation.

For us, resource constraint has also been a major problem. We could only fine-tune the event extraction models for 3 epochs and limited GPU usage leading to large amount of training and testing time.

6 Future Work

Event-tagged data can be expanded through human-annotated efforts, albeit at a significant cost in terms of time and labor (Ahn, 2006). Alternatively, data augmentation techniques can be employed to enhance dataset size and diversity. One effective method involves replacing event-triggering words and their corresponding arguments with appropriate synonyms using tools such as **spaCy** (Honnibal et al., 2020) or **WordNet** (Miller, 1994) (Lin et al., 2020).

Moreover, while existing datasets like **CoNLL-2003** (Tjong Kim Sang and De Meulder, 2003) are comprehensive in terms of named entity recognition, their event schemas remain relatively limited. Once the challenge of insufficient annotated data is addressed, alternative architectures beyond transformer-based models—such as **Bi-directional**

LSTMs (Huang et al., 2015) and **Graph Neural Networks** (Scarselli et al., 2009) with attention mechanisms—can be explored. These models are capable of capturing deeper semantic relationships, thereby improving the performance of event extraction systems, as demonstrated in (Liu et al., 2018) (Balali et al., 2021) (Fei et al., 2023).

7 Conclusion

This work demonstrates that effective event extraction from speech is possible even under compute-constrained, noisy-input scenarios. Through the use of PEFT via LoRA and a robust ASR+transformer pipeline, our system outperformed all other submissions in the XLLM Shared Task II. Future work will explore improving robustness to ASR noise and enhancing low-resource adaptability via data augmentation and semi-supervised learning.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Ali Balali, Masoud Asadpour, and Seyed Hossein Jafari. 2021. Cofee: A comprehensive ontology for event extraction from text, with an online annotation tool. *CoRR*, abs/2107.10326.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2023. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. *Preprint*, arXiv:2304.06248.

Hao Fei, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2024. Xnlp: An interactive demonstration system for universal structured nlp. *Preprint*, arXiv:2308.01846.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. If you use spaCy, please cite it as below.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. *Preprint*, arXiv:2106.09232.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Bin Wang, Meishan Zhang, Hao Fei, Yu Zhao, Bobo Li, Shengqiong Wu, Wei Ji, and Min Zhang. 2024. Speechee: A novel benchmark for speech event extraction. *Preprint*, arXiv:2408.09462.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2024. Next-gpt: Any-to-any multimodal llm. *Preprint*, arXiv:2309.05519.