INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Context-Constrained Sentence Modeling for Deception Detection in Real Interrogation

Ya-Tse Wu[1†], *Yuan-Ting Chang*[2†], *Shao-Hao Lu*[1], *Jing-Yi Chuang*[3] *and Chi-Chun Lee*[1]

[1]Department of Electrical Engineering, National Tsing Hua University, Taiwan
[2]Agency Against Corruption, Ministry of Justice, Taiwan
[3]CPC Corporation, Taiwan

crowpeter@gapp.nthu.edu.tw, aac14030@mail.moj.gov.tw, rosyhoward1223@gmail.com, allison.sobra@gmail.com, cclee@ee.nthu.edu.tw

Abstract

Detecting deception in real interrogations for criminal cases is critically important. Interrogation is composed of evidence-driven conversation that calls for a need for proper integration of context, where most prior works treat it as a sequence modeling task. In this work, we propose a context-constrained sentence modeling approach for deception detection. Specifically, we introduce the use of a global context label that is defined on multi-sentences, i.e., a context label is marked as deception if any of its sentences are deceptive. Then, by using a contextual integrator that aggregates predictions on local sentences for context label prediction, we improve deception detection by jointly optimizing global and local labels. Our approach significantly outperforms other models and achieves 76.38% and 73.15% in Unweighted Average Recall (UAR) at the local and global levels, respectively. We also conducted two analyses to further demonstrate the effectiveness of our approach.

Index Terms: Deception Detection, Contextual Modeling, Real Interrogation

1. Introduction

Deception is a common behavior in our daily life, but the accuracy for detecting deception by humans is only a little better than chance at 54% [1]. Much research is thus developed with an aim to achieve better deception detection by modeling objective behavior cues, e.g., linguistic cues [2], face expressions [3], acoustic features [4], or multi-model fusion [5]. These works rely on data collected from two major types of sources; 1) recruiting subjects to engage in dialog games or recordings of the TV show, and 2) the real interrogation of criminal cases or trial data from court recordings. While the first type helps bring insights into deceptive behaviors, the ability to detect deception for the second type is desirable as the consequence of a low detection rate in these scenarios can be severe.

The studies involving real interrogation datasets are quite limited to-date [6, 7, 8]. A few notable works include: Hsiao et al. build a network with several BiLSTMs and attention layers to encode the visual, audio and transcription information for real-life trial video data [9]; on the same dataset, Zhang et al. train a graph attention cross-modal network for multi-model fusion to accomplish the same task on question-level [5]; Serban et al. develop a use of combining convolutional neural networks and multilayer perceptrons to perform voice activity and deceptive speech detection in a single system for real criminal cases [10]. These studies concentrate on single sentence modeling, ignoring the "context". In real practice, the agents question the respondents around pieces of evidence they have, which create a scenario ("context") around the back-and-forth conversation turns during interrogation. In fact, Blair et al. demonstrate the importance of situating subjects within a meaningful context and having important implications for deception theory [11].

This contextual effect has just been considered in recent works of deception detection in conversation. For example, Chou et al. extract conversational temporal dynamic features of the QA pair turns, and further fuse with acoustic-prosodic features and BERT to build a bi-directional LSTM multi-model deception detector [12]; Fornaciari et al. aggregate the prior sentences using BERT for text-pair representations and perform deception detection using a Hierarchical Transformers [13]; Bao et al. propose a context selector network in multi-turn QA, which uses cosine similarity to mask the context for deception detection task [14]. For these works, they treat context modeling as the ability to carry information from past sentences to current target sentences by utilizing variants of time-series models. However, in this work, we argue that the "context" should act more as a global-constraint on sentences in conversation rather than simple temporal progression as inspired by the practice of evidence-driven interrogation [15].

In this work, we propose a novel context-constrained sentence modeling for deception detection in real interrogation. The core idea is to improve sentence-level learning by imposing an additional aggregation loss that is derived from the global context label. This context label is assigned at the (global) level of multi-sentences, i.e., if any of the local sentences are marked as deception, the global label is set as "deception". Specifically, our deception model first takes sentence-level multimodal inputs to generate local deception probability at Question-Answer (QA) level. Then, a context integrator that is built based on a self-attention mechanism aggregates the multi-sentence probabilities to recognize the global label, acting as an additional constraint that backpropagates to the sentence-level deception model. By collaborating with Agency Against Corruption, Ministry of Justice (AAC), we evaluate our method on a real interrogation dataset. Our method achieves a binary classification result of UAR 76.38%, 73.15% at the local-level and the global-level, respectively. We also carry out an analysis on different context lengths and illustrate the importance of context for this task.

2. Research Methodology

2.1. Dataset

Our dataset consists of Chinese Mandarin audio recordings (44.1kHz, 1Ch) collected by Agency Against Corruption, Ministry of Justice. The dataset comprises 33 real interrogation cases, with a total speech duration of 31.57 hours. Each case includes multiple QA turns between one agent and one respon-

† These authors share first authorship.

10.21437/Interspeech.2023-890

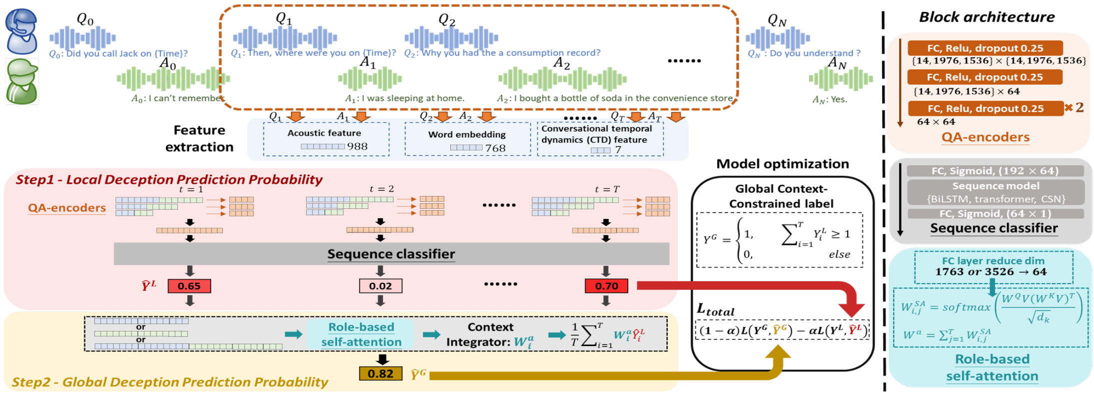

Figure 1: *Left: Our proposed framework of context-constrained sentence modeling for deception detection. Right: The block detail in our network.*

Table 1: *Label summary. N=Number; T=Truth; D=Deception.*

	T-Cases	D-Cases	Total	Avg. Q+A length
T-QA	5181	6377	11558	6.90s
D-QA	0	1707	1707	19.88s
Total QA	5181	8084	13265	8.57s

dent, with an average of 401.97 turns per interrogation. The minimum and maximum number of turns per interrogation are 65 and 890, respectively. The recordings are manually transcribed and segmented into speech segments of the agents and respondents. We then down-sample the audio to 16kHz using Librosa [16]. The dataset includes 17 deceptive and 16 truthful respondents, with interrogation duration ranging from 0.5 to 7 hours.

2.1.1. Deception Annotation

Each answering sentence in our dataset is annotated as truthful (T) or deceptive (D) by investigation agency agents based on 13 reliable pieces of evidence, including self-statements, statements from others, contact records, police surveillance, communication surveillance, financial records, departure and arrival records, household registration, motor vehicle supervision, insurance records, taxation records, documentation, and digital evidence. The agents review each QA-turn and make their annotations accordingly, adhering to the principle of secrecy in investigations. The annotations are performed six months after the case is closed to maintain confidentiality. The dataset contains a total of 13,265 QA pairs, with 11,558 truthful answering sentences and 1,707 deceptive answering sentences (detail is in Table 1). We are unable to release the dataset due to the sensitive nature of these real investigations.

2.1.2. Context-Constrained Deception Label

Deceptive respondents often switch between telling the truth and lying in response to different pieces of evidence. If respondents perceive that the agents are unable to detect their deceptions based on the available evidence, they may continue to deceive the agents throughout a short session. To properly capture this tendency, we define a global context-constrained label, denoted as Y^G, which aggregates T multiple annotated sentence labels Y^L according to the following rule:

$$Y^G = \begin{cases} 1, & \Sigma_{i=1}^{T} Y_i^L \geq 1 \\ 0, & else \end{cases} \tag{1}$$

Here, we define $Y^L = 1$ to indicate deception, and $Y^L = 0$ to indicate truth. Given T local annotations, if any of them are labeled as deceptive, the label of the context is set to deception; otherwise, it is labeled as truth.

2.2. Feature Extraction

2.2.1. Acoustic Feature

We used Opensmlie to extract Emobase as our acoustic feature [17], which was recently used in another deception study [12]. Emobase contains acoustic-prosodic properties (MFCC, fundamental frequency (F0), and voice quality envelope, etc) and their statistical functions, resulting in a feature size of 988 dimensions per utterance.

2.2.2. Word Embedding Feature

We use Bidirectional encoder representations from transformers (BERT) as our word embedding feature, which has been widely used for modeling language, even for the task of deception detection [18]. Text sequences are segmented using CKIP tools [19] and fed into a pre-trained bert-base-chinese model. We extract 768-dimensional features from the output of the last encoder layer for each sentence. If a sentence exceeds the input length limit, we split it in half and concatenate the output embeddings.

2.2.3. Conversational Temporal Dynamics (CTD) feature

- **Duration (d):** : The total turn duration of agent's questioning turn or respondent's answering turn.
- **Speed:** The speaking speed of one question or answering compute by N_{words}/d.
- **Silence duration (d_s):** The duration between the end time point of last utterance and the start time point of current utterance.
- **Silence utterance ratio:** The ratio between d and d_s on one utterance, computed by (d_s/d).
- **Duration difference:** The difference between current utterance duration and last utterance duration $(d - d_{last})$.
- **Duration addition:** The addition between current utterance duration and last utterance duration $(d + d_{last})$.
- **Duration ratio:** The ratio between current utterance duration and last utterance duration (d/d_{last}).

2.3. Context-Constrained Sentence Modeling

The overall structure of our proposed framework is depicted in Figure 1. The main idea of this approach is to jointly optimize the network using a context-constrained global label and local sentence labels. There are two step predictions in the network: (1) The QA-encoders and the sequence classifier are used to predict the local deception probability $\hat{Y}^L \in \mathbb{R}^{N \times T}$ and (2) Integrate the local deception probability to global deception probability prediction $\hat{Y}^G$ by using the context integrator W^a with the following equation.

$$\hat{Y}^G = \frac{1}{T}\Sigma_i W_i^a \hat{Y}_i^L \tag{2}$$

Since there are two levels of predictions, the total loss of the model L_{total} is computed as:

$$L_{total} = (1-\alpha)L(Y^G, \hat{Y}^G) + \alpha L(Y^L, \hat{Y}^L) \tag{3}$$

Here, $\hat{Y}^G$ is the global context-constrained label, and $\alpha = 0.5$ represents an equal weighting of the global and local loss. We will further detail each step in the following section.

2.3.1. Step1 - Local Deception Prediction Probability

There are three QA-Encoders in the network that are used to compute the QA-representations of three modalities: acoustic, word and CTD features, respectively. The QA-Encoder is a stack of 4 fully-connected layers (FC) with dropout rate 0.25 and ReLU activation function, its task is to fuse the information from questioning and answering. The input of a QA-Encoder carries a time sequence T of interrogation sentence features obtained from three modalities. The resulting concatenated representation of the three modalities is then sent through a sequence classifier followed by a FC with a sigmoid function to obtain the local deception prediction probability.

2.3.2. Step2 - Global Deception Prediction Probability

We propose a role-based self-attention weight to integrate the local prediction probability sequence. There are two roles in the interrogation, the agent and the respondent; the concatenated vector of multi-modalities features from either one or both of them is passed through a reduced dimension fully-connected layer and a self-attention layer to derive a self-attention weight. We use self-attention weight from the role-based features as the final integrator for two reasons. Firstly, it is not reasonable to use a probability from local to derive another probability-based attention weight (eventually vanishing). Secondly, by using different role-based features (from agents, from respondents, or both), we can further analyze the importance of each role in the interrogation for detecting deception. We denote the vector that we compute self-attention weight as $V \in \mathbb{R}^{T \times F}$. The integrator W^a is defined as follows:

$$W_{i,j}^{SA} = softmax(\frac{W^Q V (W^K V)^T}{\sqrt{d_k}}) \tag{4}$$

$$W^a = \Sigma_{j=1}^T W_{i,j}^{SA} \tag{5}$$

Here, $W^{SA} \in \mathbb{R}^{N \times T \times T}$ is the self-attention weight, W^Q and W^K are the components of the standard self-attention operation. The final integrator W^a is the summation over the second dimension of W^{SA}, which means that we use the self-attention weight over time to obtain the final integrator. We provide all of our source code on a github repository[1].

[1] `https://github.com/crowpeter/RealDeception`

3. Experimental Setup and Results

3.1. Experimental Setup

We conducted our experiments on a two-class deception classification task using the provided real interrogation dataset. All experiments are implemented in PyTorch 1.11.0 [20], and each model is trained on a Nvidia GeForce RTX 2080 Ti GPU for approximately 1 hour. We perform a 5-fold case-independent cross-validation. We also reserve an additional 10% of data in each training set as a validation set. We use a batch size of 64 and a maximum of 30 epochs for each model, with early stopping applied. All trainable parameters in the models are initialized using the default PyTorch settings and trained using the binary cross-entropy loss via Equation 3, which is updated using an Adam optimizer with a learning rate between $1e^{-3}$ and $5e^{-4}$. To address the label imbalance issue, we use the reciprocal of the global label class distribution as the data sampling weight during training. The probability threshold for deception is set at 0.5. The evaluation metric we used to assess model performance is the unweighted average recall (UAR).

3.2. Models and Integrator Comparison

- **Context Selector Network (CSN)** [14]: Context selector network is recently proposed for deception detection using natural language text. It comprises a word encoder (BERT), a context selector (cosine similarity masking), context encoders (two Bi-Gate Recurrent Units) and a classifier. We re-implement their network and replace the word encoder with our encoded multi-modality features of stack FC layers (mentioned in section 2.3). The number of model parameters N_m is about 5.0M.
- **BiLSTM** [12] and **transformer** [13]: We use BiLSTM and transformer as our local sentence sequence classifier. In both models, we stack 2 encoders with 2 heads in the Transformer. The embedding vector is passed into the fully connected layer with a sigmoid function for the local sentence prediction. The N_m is about 4.9M and 6.5M in Transformer and BiLSTM, respectively.
- **Different Integrator:** We compare our approach with different integrators, which is inspired by various pooling methods in the multiple instance learning framework [21].
  - **No-Integration**: No using the integrator.
  - **Max**: Using the maximum local probability in length T context as the global probability.
  - **Mean**: Using the average local probability in length T context as the global probability.
  - **Att** (Attention with average): The operation is the same as the equation 2 but the W^a is derived by feeding local probability over the T sequence to a simple fully connected layer with a sigmoid function. N_m increases by T^2 if we use **Att**.
  - **Hybrid** (Attention with maximum): Similar to the **Att**, but we change the average time pooling into a maximum pooling over time in the operation.
  - **Q**, **A**, and **QA**: Our proposed method with three different integrators. The three different integrators are derived by using features from agent (questioning, Q-Integrator), from respondent (answering, A-Integrator) and from both (question-answering, QA-Integrator), respectively. N_m increased by 180k and 390k for using Q/A-Integrator and QA-Integrator, respectively.

Table 2: *The unweighted average recall (UAR) for all models and different integrators with T=15.*

Local-level (Sentence)								
Methods	No-Integration	Max	Mean	Att	Hybrid	Q-Integrator	A-Integrator	QA-Integrator
CSN	61.54	64.04	68.70	68.86	68.09	67.97	66.30	69.33
BiLSTM	58.91	70.62	71.28	72.25	68.85	69.39	74.00	66.50
Transformer	62.94	68.32	73.21	71.03	67.11	65.85	***76.38**	67.98

Global-level (Multi-Sentences Context)								
Methods	No-Integration	Max	Mean	Att	Hybrid	Q-Integrator	A-Integrator	QA-Integrator
CSN	-	66.41	66.57	52.43	61.70	65.50	66.08	68.55
BiLSTM	-	68.80	69.63	69.63	67.73	70.43	71.27	66.13
Transformer	-	68.79	71.31	69.53	68.37	64.78	***73.15**	67.12

Table 3: *A case study in deception case.*

	Questions	Answers	Local ground truth	No Intr	A-Intr
Turn-1	OK, did you send these pictures to your group?	No, no, no.	0	0	1
Turn-2	These pictures didn't exist in your group?	I didn't send any picture or message, I didn't have it.	0	0	1
...	...	...	...	...	...
Turn-5	Thinking again.	Yes.	1	0	1
Turn-6	Are you sure?	From my memory, yes.	1	0	1
Turn-7	Again, are you sure?	Yes, I'm sure.	1	0	1
Turn-8	OK, did you remember correctly?	Yes.	1	0	1
Tuen-9	Because it seems you did?	From my memory, I didn't.	1	0	1
...	...	...	...	...	...

Table 4: *The UAR result of the transformer in different T.*

	Q-Integrator		A-Integrator	
T	Local-level	Global-level	Local-level	Global-level
5	***65.71**	***65.21**	64.23	63.91
10	***69.80**	***69.95**	64.56	64.19
15	65.85	64.78	***76.38**	***73.15**
20	64.11	64.53	***71.50**	***70.26**

3.3. Result and Analysis

3.3.1. Performance Comparison

Table 2 shows the results of binary deception classification for all sequence classifiers and different integrators with $T = 15$. At the local sentence level, we see that all of the performances using the integrator (column > 2) are better than without one (column 1), demonstrating the effectiveness of our context-constrained sentence modeling approach. Additionally, we observe that each sequence classifier achieves its best results using our proposed method (columns 6 to 8), which improved 0.47, 1.75, and 3.17 over the best results of others (columns 2 to 5) on CSN, BiLSTM, and Transformer, respectively. The best result is obtained using the combination of Transformer and the A-Integrator, with improvements of 7.05 and 2.38 compared to the best results of CSN and BiLSTM, respectively. At the global level, we observe similar trends to the local sentence level prediction, with our proposed method achieving the best results over the three models, with improvements of 1.98, 1.64, and 1.84. Again, the best combination is Transformer and the A-Integrator, which outperforms CSN and BiLSTM by 4.6 and 1.88, respectively. Overall, the UAR reach 76.38%, 73.15% at the local-level and the global-level, respectively.

3.3.2. Different Context Length Analysis

In this section, we analyze the results of our Transformer model with different T values for local-level sentence labels used in our context-constrained sentence modeling approach. Table 4 shows the results of different training parameters T with Q-Integrator and A-Integrator. We observe that as T increases ($T = 15, 20$), the performance of the A-Integrator improves when compared to the Q-Integrator. This suggests that as a larger window of contextual information is provided, the model is better capable at identifying deceptive messages from the respondents' answers. Intuitively, due to the increasing clarity of the respondent's answers, liars are often unable to maintain consistency in their stories [22]. However, we also observe that the Q-Integrator performs better than the A-Integrator in shorter contexts ($T = 5, 10$). This suggests that how the agents question the respondents may be more important than the respondents' behavior when the number of contexts is limited. This insight is similar to Chou et al. [12] for setting of dialog games.

3.3.3. Case Study

Table 3 presents a context where the addition of the A-Integrator leads to correct detection of deception. We can observe that the respondent's answers are all brief, without contextually constrained learning, it is difficult to detect those deceptive QA-pairs. With context, the ability to detect those brief sentences as deceptions is greatly enhanced. In fact, this observation aligns with known behaviors of liars who tend to simplify their word usage to avoid inconsistencies in their stories [23]. Our proposed context-constrained sentence modeling with the A-Integrator effectively captures this deceptive behavior.

4. Conclusions and Future Work

In this work, our proposed context-constrained sentence modeling approach effectively leverages both global and local contexts to improve deception detection performance. Through our experiments, we show that by incorporating contextual information significantly, it outperforms previous approaches, with the transformer followed by the A-integrator achieving the best results. Our analysis on different context lengths suggests that the type of integrator depends on the choice of context length. We also provide a case study demonstrating the importance of imposing the global context label. One limitation of this work at the moment, since the boundaries of segments relating to one piece of evidence (context) are not annotated, this makes us use a fixed T to assume the length which may not be optimal. In the future work, we are going to annotate the connection between the evidence and the QA-pairs, and further explore the modeling algorithm for this scheme.

5. References

[1] C. F. Bond Jr and B. M. DePaulo, "Accuracy of deception judgments," *Personality and social psychology Review*, vol. 10, no. 3, pp. 214–234, 2006.

[2] S. I. Levitan, A. Maredia, and J. Hirschberg, "Linguistic cues to deception and perceived deception in interview dialogues," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1941–1950.

[3] M. Ding, A. Zhao, Z. Lu, T. Xiang, and J.-R. Wen, "Face-focused cross-stream network for deception detection in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7802–7811.

[4] H.-C. Chou, Y.-W. Liu, and C.-C. Lee, "Joint learning of conversational temporal dynamics and acoustic features for speech deception detection in dialog games," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1044–1050.

[5] H. Zhang, Y. Ding, L. Cao, X. Wang, and L. Feng, "Fine-grained question-level deception detection via graph-based learning and cross-modal fusion," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2452–2467, 2022.

[6] T. Fornaciari and M. Poesio, "Decour: a corpus of deceptive statements in italian courts." in *LREC*, 2012, pp. 1585–1590.

[7] V. Pérez-Rosas, M. Abouelenien, R. Mihalcea, and M. Burzo, "Deception detection using real-life trial data," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, 2015, pp. 59–66.

[8] Y.-Y. Kao, P.-H. Chen, C.-C. Tzeng, Z.-Y. Chen, B. Shmueli, and L.-W. Ku, "Detecting deceptive language in crime interrogation," in *HCI in Business, Government and Organizations: 7th International Conference, HCIBGO 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*. Springer, 2020, pp. 80–90.

[9] S.-W. Hsiao and C.-Y. Sun, "Attention-aware multi-modal rnn for deception detection," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 3593–3596.

[10] S. Mihalache and D. Burileanu, "Using voice activity detection and deep neural networks with hybrid speech feature extraction for deceptive speech detection," *Sensors*, vol. 22, no. 3, p. 1228, 2022.

[11] J. P. Blair, T. R. Levine, and A. S. Shaw, "Content in context improves deception detection accuracy," *Human Communication Research*, vol. 36, no. 3, pp. 423–442, 2010.

[12] H.-C. Chou, Y.-W. Liu, and C.-C. Lee, "Automatic deception detection using multiple speech and language communicative descriptors in dialogs," *APSIPA Transactions on Signal and Information Processing*, vol. 10, p. e5, 2021.

[13] T. Fornaciari, F. Bianchi, M. Poesio, D. Hovy *et al.*, "Bertective: Language models and contextual information for deception detection," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2021.

[14] Y. Bao, Q. Ma, L. Wei, D. Wang, W. Zhou, and S. Hu, "Deception detection towards multi-turn question answering with context selector network," in *PRICAI 2022: Trends in Artificial Intelligence: 19th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2022, Shanghai, China, November 10–13, 2022, Proceedings, Part I*. Springer, 2022, pp. 304–315.

[15] J. K. Swanner, C. A. Meissner, D. J. Atkinson, and R. E. Dianiska, "Developing diagnostic, evidence-based approaches to interrogation," *Journal of Applied Research in Memory and Cognition*, vol. 5, no. 3, pp. 295–301, 2016.

[16] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[17] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1459–1462.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] R. Sproat and T. Emerson, "The first international chinese word segmentation bakeoff," in *Proceedings of the second SIGHAN workshop on Chinese language processing*, 2003, pp. 133–143.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[21] R. Hebbar, P. Papadopoulos, R. Reyes, A. F. Danvers, A. J. Polsinelli, S. A. Moseley, D. A. Sbarra, M. R. Mehl, and S. Narayanan, "Deep multiple instance learning for foreground speech localization in ambient audio from wearable devices," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, no. 1, p. 7, 2021.

[22] B. M. DePaulo, J. J. Lindsay, B. E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper, "Cues to deception." *Psychological bulletin*, vol. 129, no. 1, p. 74, 2003.

[23] L. A. Strömwall, M. Hartwig, and P. A. Granhag, "To act truthfully: Nonverbal behaviour and strategies during a police interrogation," *Psychology, Crime & Law*, vol. 12, no. 2, pp. 207–219, 2006.