Spoken Multiple-Choice Question Answering Using Multi-turn Audio-extracter BERT

Shang-Bao Luo, Chia-Chih Kuo, Kuan-Yu Chen
National Taiwan University of Science and Technology, Taiwan
E-mail: {M10615012, M10815022, kychen}@mail.ntust.edu.tw

***Abstract*— In spoken multiple-choice question answering (SMCQA) task, given a passage, a question, and multiple choices all in the form of speech, the machine needs to pick the correct choice to answer the question. A common strategy is to employ an automatic speech recognition (ASR) system to translate speech contents into auto-transcribed text. Therefore, a SMCQA task is reduced to a classic MCQA task. Under the strategy, bidirectional encoder representations from transformers (BERT) can achieve a certain level of performance despite ASR errors. However, previous studies have evidenced that acoustic-level statistics can compensate for text inaccuracies caused by ASR systems, thereby improving the performance of a SMCQA system. Accordingly, we concentrate on designing a BERT-based SMCQA framework, which not only inherits the advantages of contextualized language representations learned by BERT, but integrates acoustic-level information with text-level information in a systematic and theoretical way. Considering temporal characteristics of speech, we first formulate multi-turn audio-extracter hierarchical convolutional neural networks (MA-HCNNs), which encode acoustic-level features under various temporal scopes. Based on MA-HCNNs, we propose a multi-turn audio-extracter BERT-based (MA-BERT) framework for SMCQA task. A series of experiments demonstrates remarkable improvements in accuracy over selected baselines and SOTA systems on a published Chinese SMCQA dataset.**

I. INTRODUCTION

The arising popularity of audio sharing websites and social networks have led to significant growth in spoken content nowadays. Apart from that, the development of multimedia technology has promoted the popularity of voice assistant applications, which are now frequently installed in a variety of mobile phones, home devices, and so on. Therefore, spoken question answering (SQA) has been an emergent challenge in recent years. Especially, the machine comprehension of spoken contents is an important technology in need. To solve the related tasks, a common strategy is to employ an automatic speech recognition (ASR) system to decode speech contents into auto-transcribed text. In this way, an SQA task is reduced into a classic text-based QA task. Although various text-based methods can be easily applied to the auto-transcribed text, it is inevitable that ASR errors will harm the performance of the simple strategy. Owing to the natural relationship between spoken contents and auto-transcribed text, we believe the acoustic-level information in the speech may provide additional cues to compensate for ASR errors. In other words, if we can find some ways to distill and leverage the acoustic-level information, we may enhance the performance of SQA.

In this study, we focus on the spoken multiple-choice question answering (SMCQA) task, where passages, questions, and choices are all given in the form of speech. The major contributions are at least twofold. First, in order to distill suitable cues from speech, we propose a novel framework, multi-turn audio-extracter hierarchical convolutional neural networks (MA-HCNNs), which encodes acoustic-level features under various temporal scopes. Second, inspired from the success of the bidirectional encoder representations from transformers (BERT) [1], the paper strives to develop an effective SMCQA framework based on BERT. To fully utilize the great potential of BERT, we propose a multi-turn audio-extracter BERT-based (MA-BERT) framework, which crossly assembles the acoustic-level information extracted from speech by MA-HCNNs and the text-level information inferred by each transformer layer in BERT. Evaluated on the data of "Formosa Grand Challenge – Talk to AI", a Mandarin Chinese SMCQA contest held in 2018, the proposed MA-BERT framework can outperform various SOTA systems by a large margin.

II. RELATED WORK

A. The Language Representation Methods

Because of the impressive successes in many NLP-related tasks, language representations have become a popular research recently. Generally speaking, the research spectrum can be classified into two main schools according to the usages for the downstream tasks [1]: (1) feature-based models and (2) fine-tuning methods.

The most representative and well-practiced feature-based models are the word embedding methods. The neural network language model [2] is the most-known seminal study, which mainly concentrates on estimating an *n*-gram language model while inducing word embeddings as a by-product. Follow-up extensions develop similar methods for probing syntactic regularities and latent semantic in the representation of words. The learned word embeddings are usually treated as feature vectors for various downstream tasks. Representative methods include continuous bag-of-words model [3], skipgram model [3], global vector model [4], and ELMo model [5].

In the latter school, the leading fine-tuning methods include OpenAI GPT [6], BERT [1], XLNet [7], RoBERTa [8], and ALBERT [9]. The fine-tuning methods usually consist of two parts: pretraining and task-specific parameters finetuning. Formally, such a school of methods usually leverage a self-

supervised objective to obtain a pretrained model, and then a minimal set of task-specific parameters is introduced. After that, all (or only the task-specific) of the model parameters are trained toward the objective of the downstream task [6].

B. The Multiple-choice Question Answering

In a text-based multiple-choice question answering (MCQA) task [10–14], the input to the model includes a passage, a question, and several answer choices. The passage usually consists of several sentences, while the question and each answer choice are almost a single sentence. A question answering model is designed to select a correct answer from multiple choices based on the information given in the passage and question. Previous studies usually concentrated on utilizing lexical and syntactic information in the passage to infer the answer [15–18], while recent research has turned to present various MCQA models based on neural networks [11–14]. Classic methods include hierarchical attention-based CNN [19], parallel-hierarchical neural model [20], and hierarchical attention flow model [21], to name just a few. Although several elaborative mechanisms have been proposed based on deep neural networks, query-based attention CNN (QACNN) model [13] can be considered as a representative. Specifically, QACNN first computes the similarity matrices between passage and question, also passage and choice. Then, a CNN layer with query-based attention mechanism is employed to learn the location relationship pattern from the similarity matrices. Thus, the model crossly compares through passage, question and choice, and eventually decide an answer choice.

Opposite to the conventional MCQA task, passage, question, and choices are all in the form of speech in a spoken multiple-choice question answering (SMCQA) task. A naïve but easy solution is to first transcribe these speech utterances into text using an ASR system. Thereafter, a text-based method (e.g., QACNN) can be readily applied to the auto-transcribed text. Such a strategy only considers text-level information, while it is obvious that the audio may contain useful cues for answer prediction. Hence, several studies have been proposed to cope with the SMCQA task by considering both text-level and acoustic-level features. CNN-based hierarchical multistage mutimodal (HMM) framework [22] and SpeechBERT [23] model are representatives. The former tries to explore both the text-level and the acoustic-level relationships between a pair of passage and choice as well as a pair of passage and question by CNN-based attention mechanism. The latter assumes that the passage is given in the form of speech, while the question is in the form of manual transcription (i.e., without recognition errors). A concatenation of question and passage can then be fed into a BERT model, which makes it possible to explore the relationship between acoustic-level and text-level cues with the self-attention [24] mechanism. Although the HMM model seems to equip comprehensive ability for SMCQA task, it doesn't leverage the merits in recent language representation models (e.g., BERT). On the other hand, SpeechBERT, which takes acoustic features as additional inputs to the BERT model, may suffer from the input length limitation of BERT so as to downgrade the performance and make the model inflexible.

III. METHODOLOGY

A. Vanilla BERT method

Recently, BERT [1] has attracted much interest due to its superior performances in several NLP-related tasks [25–27], including question answering task [28]. When BERT comes to the MCQA task, a naïve but effective way is employing BERT to encode a concatenation token sequence of a passage, a question, and one of choices (a PQC pair). Then, a classifier is introduced to predict which choice is the correct answer to the question, as shown in Fig. 1.

A SMCQA task is a MCQA task while all PQC pairs are given in the form of speech. Since BERT is only capable of encoding texts, a common strategy is to take auto-transcribed text of PQC pairs as its input. Formally, we build a concatenation sequence X_s from the wordpiece token sequences of passage $P = \{P_1, P_2, \dots, P_{|P|}\}$, question $Q = \{Q_1, Q_2, \dots, Q_{|Q|}\}$, and s^{th} choice $C_s = \{C_1, C_2, \dots, C_{|C_s|}\}$:

$$X_s = \{[CLS], P, [SEP], Q, C_s, [SEP]\}, \quad (1)$$

where "[CLS]" is a special token used for feature extraction and "[SEP]" is a separator token. $|P|$, $|Q|$, and $|C_s|$ denote the token sequence lengths of P, Q and C_s, respectively.

Next, the concatenation token sequence X_s is embedded as $E_s^{token} \in \mathbb{R}^{h \times |X_s|}$, where h is the hidden size of BERT. On top of E_s^{token}, we add the position embeddings $E_s^{position} \in \mathbb{R}^{h \times |X_s|}$ and the segment embeddings $E_s^{segment} \in \mathbb{R}^{h \times |X_s|}$. Thus, we obtain $H_s^0 \in \mathbb{R}^{h \times |X_s|}$:

$$H_s^0 = E_s^{token} + E_s^{position} + E_s^{segment}. \quad (2)$$

After that, we pass H_s^0 through U transformer layers of BERT. For the u^{th} transformer layer, we retrieve $H_s^u \in \mathbb{R}^{h \times |X_s|}$:

$$H_s^u = Transformer_{encoder}(H_s^{u-1}) \; \forall u \in [1, U]. \quad (3)$$

Then, we extract the vector corresponding to the "[CLS]" token from H_s^U, and pass it through a fully-connected feed-forward layer with parameters $W_1^{FC} \in \mathbb{R}^{1 \times h}$ and $b_1^{FC} \in \mathbb{R}^{1 \times h}$. By doing so, a relevance score r_s for the s^{th} choice can be obtained:

$$r_s = W_1^{FC} H_s^U [CLS] + b_1^{FC}. \quad (4)$$

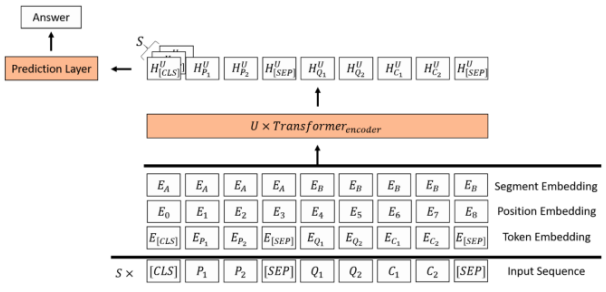

Fig. 1 A Vanilla BERT SMCQA system.

Finally, the training objective of the Vanilla BERT model is to maximize the likelihood of the correct choices by stacking a softmax function upon all the relevance scores of S candidate choices:

$$P(C_s) = \frac{exp(r_s)}{\sum_{s'=1}^{S} exp(r_{s'})}. \quad (5)$$

For testing, the candidate choice with the highest relevance score (i.e., r_s) will be selected as the answer.

B. *Multi-turn Audio-extracter Hierarchical Convolutional Neural Networks (MA-HCNNs)*

In a SMCQA task, text inaccuracies caused by the ASR system could degrade the performance of a text-based system. To compensate for the ASR errors, a hierarchical multistage multimodal (HMM) framework [22], which incorporates acoustic features with a CNN-based MCQA model has been proposed. Though BERT has a large number of parameters and a deep network architecture, it was not designed to blend acoustic-level and text-level information. To make BERT fully utilize acoustic features, we propose multi-turn audio-extracter hierarchical convolutional neural networks (MA-HCNNs), which stack multiple layers of CNNs to extract acoustic features in various temporal scopes.

In the same way of the Vanilla BERT method, we first tokenize auto-transcribed texts into wordpiece token sequences of passage P, question Q, and s^{th} choice C_s. At the pre-processing stage, we first segment the speech to align with these tokens. Then, we compute the acoustic features (e.g. MFCCs) of the segmented speech. Hence, for each PQC pair $\{P, Q, C_s\}$, we retrieve their acoustic features $A^P = \left\{A_1^P, A_2^P, \dots, A_{|A^P|}^P\right\}$, $A^Q = \left\{A_1^Q, A_2^Q, \dots, A_{|A^Q|}^Q\right\}$, and $A^{C_s} = \left\{A_1^{C_s}, A_2^{C_s}, \dots, A_{|A^{C_s}|}^{C_s}\right\}$. We take $\{A^P, A^Q, A^{C_s\prime}\}_s$, as the input of MA-HCNNs, and predict the correct answers based on encoded acoustic features, as shown in Fig. 2.

Next, we pass $\{A^P, A^Q, A^{C_s}\}$ through a CNN layer and a max pooling layer to obtain the critical temporal features $\left\{E_P^A, E_Q^A, E_{C_s}^A\right\}$:

$$E_P^A = MaxPool(W_0^{CNN} \otimes A^P) \in \mathbb{R}^{h \times |A^P|}, \quad (6)$$
$$E_Q^A = MaxPool(W_0^{CNN} \otimes A^Q) \in \mathbb{R}^{h \times |A^Q|}, \quad (7)$$
$$E_{C_s}^A = MaxPool(W_0^{CNN} \otimes A^{C_s}) \in \mathbb{R}^{h \times |A^{C_s}|}. \quad (8)$$

where $\otimes$ means the convolution operation, W_0^{CNN} is the parameter of the CNN layer, and h is the number of output channels of the CNN layer.

Similar to the Vanilla BERT, we concatenate the features $\left\{E_P^A, E_Q^A, E_{C_s}^A\right\}$ to obtain $G_s^0 \in \mathbb{R}^{h \times (|A^P| + |A^Q| + |A^{C_s}|)}$:

$$G_s^0 = [E_P^A; E_Q^A; E_{C_s}^A]. \quad (9)$$

Then, we pass G_s^0 through U layers of CNNs. For the u^{th} CNN layer, we obtain $G_s^u \in \mathbb{R}^{h \times (|A^P| + |A^Q| + |A^{C_s}|)}$:

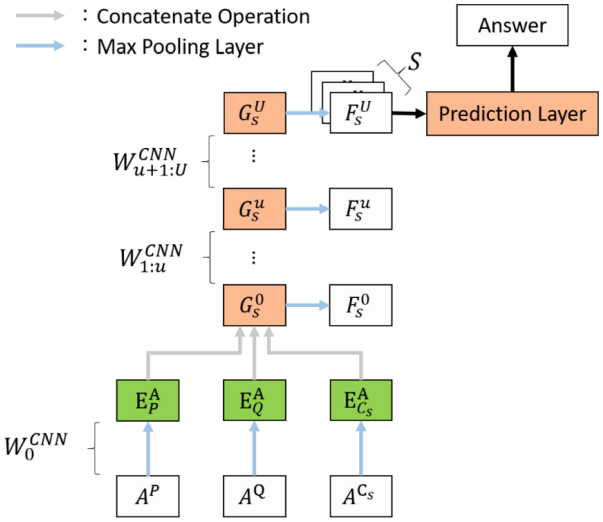

Fig. 2 The SMCQA framework of multi-turn audio-extracter hierarchical convolutional neural networks (MA-HCNNs).

$$G_s^u = W_u^{CNN} \otimes G_s^{u-1} \; \forall u \in [1, U], \quad (10)$$

where W_u^{CNN} is the parameter of the u^{th} CNN layer, and the dimension of inputs and outputs of CNNs are kept unchanged using the padding mechanism.

Likewise, we pass G_s^u through a max pooling layer to obtain the critical features $F_s^u \in \mathbb{R}^h$:

$$F_s^u = MaxPool(G_s^u) \; \forall u \in [0, U]. \quad (11)$$

Finally, we pass F_s^U through a fully-connected feed-forward layer with parameters $W_2^{FC} \in \mathbb{R}^{1 \times h}$ and $b_2^{FC} \in \mathbb{R}^{1 \times h}$. By doing so, a relevance score r_s for the s^{th} choice can be obtained

$$r_s = W_2^{FC} H_s^U[CLS] + b_2^{FC}, \quad (12)$$

Following the same training and testing approach of the Vanilla BERT method, we stack a softmax function upon all the relevance scores of S candidate choices:

$$P(c_s) = \frac{exp(r_s)}{\sum_{s'=1}^{S} exp(r_{s'})}. \quad (13)$$

C. *BERT-RNN*

The Vanilla BERT method generates a sequence of reading comprehension vectors H_s^u for each transformer layer in BERT, as stated in (3). However, only the vectors H_s^U, generated by the last transformer layer in BERT, is utilized to predict the relevance score r_s. Since each transformer layer in BERT encodes the contextual information in different scopes, it may be beneficial to utilize the outputs vectors from all transformer layers in BERT. Hence, we propose BERT-RNN, which uses a gated recurrent unit (GRU) [29] to further encode the output vectors of all transformer layers in BERT, as shown in Fig. 3.

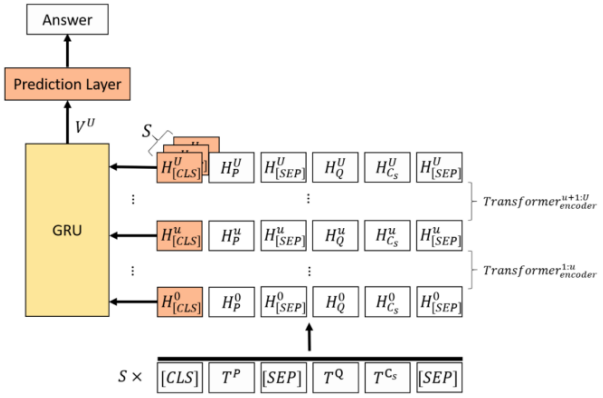

Fig. 3 The SMCQA framework of BERT-RNN.

First, we extract the vector corresponding to the "[CLS]" token from H_s^u for each transformer layer. Next, we pass the extracted vector through a GRU layer to obtain $v_s^u \in \mathbb{R}^h$:

$$v_s^u = GRU(v_s^{u-1}, H_s^u[CLS]) \; \forall u \in [0, U], \tag{14}$$

where h is the hidden size of the GRU layer.

Then, we pass v_s^U through a fully-connected feed-forward layer with parameters $W_3^{FC} \in \mathbb{R}^{1\times h}$ and $b_3^{FC} \in \mathbb{R}^{1\times h}$ to obtain. a relevance score r_s for the s^{th} choice. Following the same training and testing approach of the Vanilla BERT method, we stack a softmax function upon all the relevance scores of S candidate choices:

$$r_s = W_3^{FC} v_s^U + b_3^{FC}, \tag{15}$$

$$P(c_s) = \frac{exp(r_s)}{\sum_{s'=1}^{S} exp(r_{s'})}. \tag{16}$$

D. *Multi-turn Audio-extracter BERT (MA-BERT)*

We have introduced MA-HCNNs and BERT-RNN in previous sections. In this section, we want to further induce the SMCQA model to learn the natural relationship between text-level and acoustic-level information. To achieve the goal, we propose a multi-turn audio-extracter BERT-based (MA-BERT) framework, which fuses MA-HCNNs and BERT together with three GRUs, as shown in Fig. 4. Owing to the property of the recurrent neural network (RNN), the text-level information encoded by BERT and the acoustic-level information encoded by MA-HCNNs can be crossly combined across various temporal scopes, hence decrease the impact of ASR errors.

First, the wordpiece tokens of a PQC pair $\{T^P, T^Q, T^{C_s}\}$ are passed through BERT. For each transformer layer in BERT, we retrieve its output vectors H_s^u, as stated in (3). Then, we extract all the vectors corresponding to the "[CLS]" token from H_s^u, and stack these vectors as R_s^T, which represents the text-level reading comprehension vectors. Likewise, the acoustic features of a PQC pair $\{A^P, A^Q, A^{C_s}\}$ are passed through MA-HCNNs. For each CNN layer in MA-HCNNs, we retrieve its max-pooled output vector F_s^u, as stated in (11). Then, we stack these vectors as R_s^A, which represents the acoustic-level reading comprehension vectors.

$$R_s^T = \{H_s^u[CLS]\}_{u=1}^U, \tag{17}$$

$$R_s^A = \{F_s^u\}_{u=1}^U. \tag{18}$$

We employ two separate GRUs to encode the text-level and acoustic-level reading comprehension vectors (i.e., R_s^T and R_s^A). As the result, we obtain vT_s^u and vA_s^u:

$$vT_s^u = GRU_T\{vT_s^{u-1}, R_s^T[u]\} \; \forall u \in [0, U], \tag{19}$$

$$vA_s^u = GRU_A\{vA_s^{u-1}, R_s^A[u]\} \; \forall u \in [0, U]. \tag{20}$$

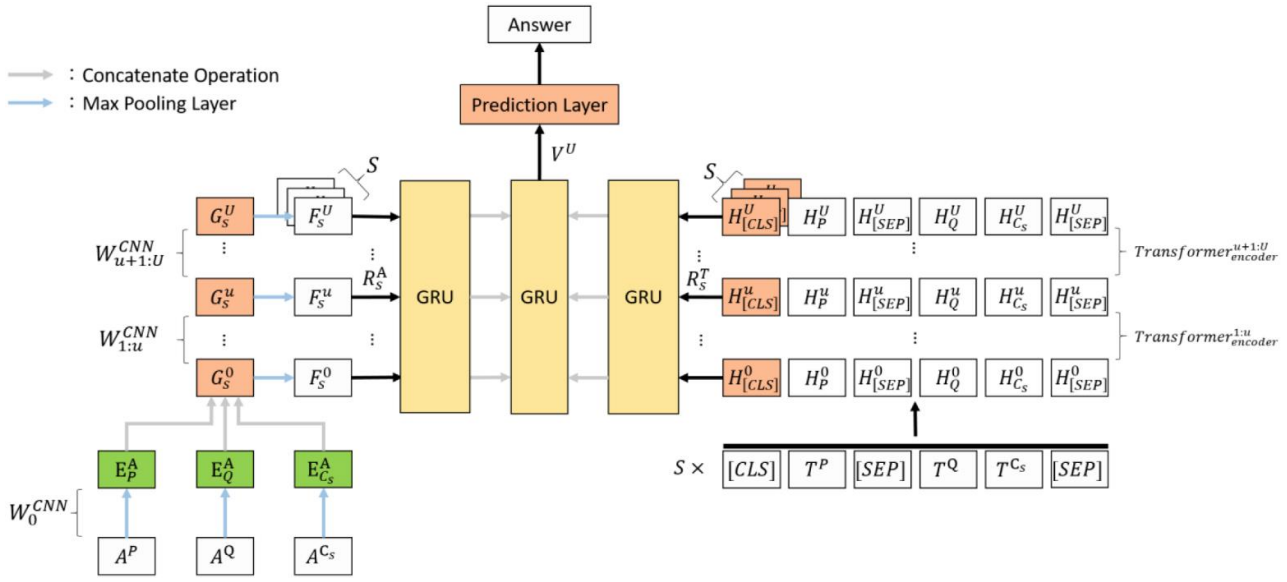

Fig. 4 The SMCQA framework of multi-turn audio-extracter BERT (MA-BERT).

Next, we concatenate vT_s^u and vA_s^u to obtain x_s^u, and pass it through a third GRU, which encodes both text-level and acoustic-level information. As the result, we obtain v_s^u:

$$x_s^u = [vT_s^u; vA_s^u], \quad (21)$$
$$v_s^u = GRU\{v_s^{u-1}, x_s^u\}\ \forall u \in [0, U]. \quad (22)$$

Finally, we take the last output vector v_s^U from the GRU, and pass it through a fully-connected feed-forward layer with parameters $W_4^{FC} \in \mathbb{R}^{1\times h}$ and $b_4^{FC} \in \mathbb{R}^{1\times h}$ to obtain a relevance score r_s for the s^{th} choice. Following the same training and testing approach of the Vanilla BERT method, we stack a softmax function upon all the relevance scores of S candidate choices:

$$r_s = W_4^{FC} v_s^U + b_4^{FC}, \quad (23)$$
$$P(c_s) = \frac{exp(r_s)}{\sum_{s'=1}^{S} exp(r_{s'})}. \quad (24)$$

IV. Experimental Setup

A. Dataset

We evaluated the proposed frameworks on the "2018 Formosa Grand Challenge – Talk to AI[1]" (FGC) dataset, which is a spoken multiple-choice question answering task in Mandarin Chinese, in the experiments. Each passage-question-choices (PQC) sets contains a passage, a question, and 4 choices, among which only one choice is the correct answer. The domain of the FGC dataset is very diverse, including science, news, and literature, to name a few. The training set consists of 7,072 PQC examplers, and there are 1,500 PQC examplers for development. An elementary and an advanced test sets were investigated in this study, and both of them contains 1,000 PQC examplers. It is worthy to mentioned that questions in the advanced test set require deep understandings for choosing correct answers.

B. ASR

Our ASR system was built up using the Kaldi toolkit [30], where the acoustic model was trained based on TDNN-F with lattice-free MMI [31, 32], followed by model refinement with sMBR [33], with 461 hours of TV and radio broadcasting speech. In audio processing, spectral analysis was applied to a 25 ms frame of speech waveform every 10 ms. For each acoustic frame, 40 MFCCs derived from 40 FBANKs, plus 3 pitch features, were used for ASR and for our proposed HMM framework. Utterance-based mean subtraction was applied to these features. The lexicon contained 91,573 Chinese words. The word-based trigram language model was trained with Kneser-Ney backoff smoothing using the SRILM toolkit [34]. The recurrent neural network language model (RNNLM) was used for lattice rescoring [35]. The training corpus was compiled from PTT[2] articles (2018) and CNA news stories (2006 ~ 2010) [36]. The character error rate (CER) of our ASR system is about 7.79%.

C. Implementation Details

The proposed frameworks were implemented by PyTorch [37], and we used the BERT model (bert-base-chinese) in the Huggingface's Transformers library [38]. Parameters of MA-HCNNs were optimized by stochastic gradient descent (SGD), while the rest were optimized by the AdamW method [39]. The hidden size of the three GRUs are set to {768, 32, 800}, respectively for encoding text-level/audio-level/both reading comprehension vectors, as stated in (19), (20), and (22). The hyperparameters: {kernel size, padding size, number of filters} of CNN layers in MA-HCNNs are set to {3, 1, 64} for W_0^{CNN}, and set to {7, 3, 768} for $W_u^{CNN}\ \forall u \in [1, U]$.

V. Experimental Result

In the first set of experiments, we evaluate various baseline systems, and the results are listed in Table 1. Systems compared in this study include a naïve baseline, a word embedding-based method and three recently proposed neural-based SOTA MCQA methods. The most naïve baseline is to choose the longest choice or the shortest choice as the answer (denoted by "Choice Length"). This method could be even worse than a random guess. In addition to the naïve baseline system, a simple strategy based on the word embeddings is investigated. The method employs the pretrained word embeddings to represent a passage-question-choice pair by averaging the embeddings of all the words in the passage-question-choice pair. Then, we can select the choice with the largest cosine similarity with the passage or the question to be the answer. The word embeddings used in this study are trained by fasttext [40] on the same corpus for ASR language model training. The dimension of the word embedding was set to 300. The results, as denoted by "Choice Similarity" in Table 1, indicate that the relationship between the question and the choice is more effective than the relationship between the passage and the choice. Next, the recently proposed neural-based methods are also compared in this study, including QACNN [13], HMM [22], and the Vanilla BERT method. It is worthy to note that both QACNN and the Vanilla BERT models only leverage auto-transcribed text for answer prediction, while HMM uses both text-level and acoustic-level information for the SMCQA task.

Valuable observations can be drawn from the results in Table 1. First, as expected, QACNN, HMM and the Vanilla BERT models performed much better than the "Choice Length" and "Choice Similarity" methods, which also reveal the ability and the potential of the neural-based methods for SMCQA task. Next, we can observe that HMM outperforms QACNN in all cases. The reason should be that the HMM model integrates both text-level and acoustic-level information for answer prediction, while the QACNN model only leverages the text-

[1] Formosa Grand Challenge - Talk to AI: https://fgc.stpi.narl.org.tw/activity/techai2018

[2] PTT: https://www.ptt.cc/index.html

level information. Moreover, the Vanilla BERT can achieve the best results in all baselines, which witnesses again the giant successes of the research on language representations.

In the second set of experiments, we make a step forward to compare the proposed frameworks with all of the baseline systems, and the experimental results are also presented in Table 1. Based on the results, several worthwhile observations can be made from the comparisons. First, MA-HCNNs can achieve a certain level of performance using acoustic-level statics only, which indicates the potential benefit of utilizing acoustic-level information in SMCQA. Second, we find that MA-BERT outperforms all of the baseline systems in all cases, which signals that it can indeed make use of both acoustic-level and text-level statistics in a systematic and theoretical way for SMCQA. Third, it is definitely helpful to take advantage of the contextual information encoded in all transformer layers in BERT, since results for BERT-RNN are better than results for Vanilla BERT in all cases. Fourth, results for the advanced test set are worse than results for the elementary test set in almost all cases, which reveal that questions in the advanced test set require deep understandings for choosing correct answers. Fifth, thanks to the large-scale pretraining of BERT models, the Vanilla BERT and the proposed BERT-RNN and MA-BERT can absolutely outperform other neural-based methods (i.e., QACNN and HMM), especially in the test sets. To sum up, by subtly manipulating both text-level and acoustic-level information, the proposed MA-BERT framework is the affirmative choice for the SMCQA task.

At the last stage, in order to exam the effect caused by the ASR errors for SMCQA task, we take Vanilla BERT, which only considers text-level information for answer prediction, as a subject. As the upper bound, the Vanilla BERT model is trained on manual-transcribed PQC sets, and the development and test sets are also in the form of manual-transcribed text. Orthogonal to the upper bound system, a model trained with erroneous transcripts by the ASR system is obtained, and the performances of the MCQA task with recognition errors are evaluated. All of the results are summarized in Table 2. The results indicate a significant performance gap between the upper bound system and the system based on auto-transcribed text, which shows that the recognition errors inevitably mislead the predictions for the Vanilla BERT model so as to degrade the MCQA performance. Accordingly, the analysis suggests that extra information, besides auto-transcribed text, should be explored to improve the SMCQA system. In summary, the proposed MA-BERT is a preferable vehicle for utilizing acoustic-level and text-level characteristics in the SMCQA task.

VI. CONCLUSION

In this paper, we have presented a multi-turn audio-extracter BERT (MA-BERT) framework, which jointly considers the acoustic-level and text-level statistics for the SMCQA task. The proposed framework has been evaluated on the 2018 Formosa Grand Challenge (FGC) dataset. The experimental results demonstrate its remarkable superiority than other strong baselines compared in the paper, thereby indicating the potential of the framework.

Table 1: *Performance (in accuracy (%)) of different systems.*

Model	Dev	Test	
		Elementary	Advanced
Choice Length			
Longest	39.14	29.58	32.36
Shortest	19.74	23.23	20.94
Choice Similarity			
Passage-Choice	26.25	25.08	24.95
Question-Choice	47.09	38.08	35.07
QACNN	63.12	71.23	39.07
HMM	66.14	72.00	40.98
Vanilla BERT	68.07	77.63	44.09
MA-HCNNs	40.02	42.72	30.27
BERT-RNN	68.33	77.84	**45.69**
MA-BERT	**70.66**	**80.34**	45.39

Table 2: *Performance (in accuracy (%)) of the vanilla BERT method with respect to manual transcriptions (Manual) or auto-transcribed texts (ASR).*

Data Usage		Experimental Result		
Training	Dev & Test	Dev	Test	
			Elementary	Advanced
Manual	Manual	77.73	88.70	50.80
ASR	ASR	68.07	77.63	44.09

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, 2019.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of the 1st International Conference on Learning Representations*, 2013.

[4] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[5] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 2227-2237, 2018.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.

[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly

optimized BERT pretraining approach", *arXiv preprint arXiv:1907.11692*, 2019.

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[10] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the right facts in the crowd: factoid question answering over social media categories and subject descriptors," *Proceedings of the 17th International Conference on World Wide Web*, pp. 467–476, 2008.

[11] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 633–644, 2014.

[12] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "LSTM-based deep learning models for non-factoid answer selection," *arXiv preprint arXiv:1511.04108*, 2015.

[13] T. Liu, Y. Wu, and H. Lee, "Query-based attention CNN for text similarity map," *arXiv preprint arXiv:1709.05036*, 2017.

[14] A. Chaturvedi, O. Pandit, and U. Garain, "CNN for text-based multiple choice question answering," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 272–277, 2018.

[15] M. Sachan, K. Dubey, E. Xing, and M. Richardson, "Learning answer-entailing structures for machine comprehension," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 239–249, 2015.

[16] K. Narasimhan and R. Barzilay, "Machine comprehension with discourse relations," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 1253–1262, 2015.

[17] E. Smith, N. Greco, M. Bošnjak, and A. Vlachos, "A strong lexical matching method for the machine comprehension test," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1693–1698, 2015.

[18] H. Wang, M. Bansal, K. Gimpel, and D. McAllester, "Machine comprehension with syntax, frames, and semantics," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 2, pp. 700–706, 2015.

[19] W. Yin, S. Ebert, and H. Schütze, "Attention-based convolutional neural network for machine comprehension," *Proceedings of the Workshop on Human-Computer Question Answering*, pp. 15–21, 2016.

[20] A. Trischler, Z. Ye, X. Yuan, J. He, and P. Bachman, "A parallel-hierarchical model for machine comprehension on sparse data," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 432–441, 2016.

[21] H. Zhu, F. Wei, B. Qin, and T. Liu, "Hierarchical attention flow for multiple-choice reading comprehension," *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 6077–6084, 2018.

[22] S. Luo, H. Lee, K. Chen, and H. Wang, "Spoken multiple-choice question answering using multimodal convolutional neural networks," *Proceedings of Automatic Speech Recognition and Understanding 2018*, 2018.

[23] Y. Chuang, C. Liu, and H. Lee, "SpeechBERT: Cross-modal pre-trained language model for end-to-end spoken question answering," *arXiv preprint arXiv:1910.11559*, 2019.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.

[25] Y. Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.

[26] W. Yang, H. Zhang, and J. Lin, "Simple applications of BERT for ad doc document retrieval," *arXiv preprint arXiv:1903.10972*, 2019.

[27] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of BERT in ranking," *arXiv preprint arXiv:1904.07531*, 2019.

[28] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *arXiv preprint arXiv:1907.10529*, 2020.

[29] K. Cho, B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," *Proceedings of Automatic Speech Recognition and Understanding 2011*, 2011.

[31] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," *Proceedings of Interspeech 2018*, pp. 3743–3747, 2018.

[32] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," *Proceedings of Interspeech 2016*, pp. 2751–2755, 2016.

[33] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," *Proceedings of Interspeech 2013*, pp. 2345–2349, 2013.

[34] A. Stolcke, "SRILM: An extensible language modeling toolkit," *Proceedings of Interspeech 2002*, 2002.

[35] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned RNNLM lattice-rescoring algorithm for automatic speech recognition," *Proceedings of the 2018 International Conference on Acoustics, Speech and Signal Processing*, pp. 5929–5933, 2018.

[36] D. Graff and K. Chen, "Chinese Gigaword (LDC2003T09)," 2003.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

[38] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "HuggingFace's Transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[39] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2019.

[40] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.