SPEECHBERT: CROSS-MODAL PRE-TRAINED LANGUAGE MODEL FOR END-TO-END SPOKEN QUESTION ANSWERING

Yung-Sung Chuang  Chi-Liang Liu  Hung-Yi Lee

College of Electrical Engineering and Computer Science, National Taiwan University
b05901033@ntu.edu.tw, {liangtaiwan1230,tlkagkb93901106}@gmail.com

ABSTRACT
While end-to-end models for spoken language understanding tasks have been explored recently, there is still no end-to-end model for spoken question answering (SQA) tasks, which would be catastrophically influenced by speech recognition errors. Meanwhile, pre-trained language models, such as BERT, have performed successfully in text question answering. To bring this advantage of pre-trained language models into spoken question answering, we propose SpeechBERT, a cross-modal transformer-based pre-trained language model. As the first exploration in end-to-end SQA models, our results matched the performance of conventional approaches. As the first work for end-to-end spoken question answering model can be jointly optimized without an error bottleneck caused by ASR. As the first exploration in end-to-end SQA models, our results matched the performance of conventional approaches. As the first work for end-to-end spoken question answering model can be jointly optimized without an error bottleneck caused by ASR. Although the room for improvement (SQA), our model gets the results close to the performance of cascading ASR and QA models. As the first exploration in end-to-end SQA models, our results matched the performance of conventional approaches. As the first work for end-to-end spoken question answering model can be jointly optimized without an error bottleneck caused by ASR.

Index Terms— Spoken Question Answering, Spoken Language Understanding, Language Model Pre-training

1. INTRODUCTION
Machine comprehension (MC) on text data has a large improvement recently after large-scale self-supervised pre-trained language models appeared, such as BERT [1], GPT [2]. Instead of learning MC tasks from scratch, these model is firstly pre-trained on a large unannotated corpus to learn self-supervised representations for general language and then is fine-tuned on the downstream MC dataset. This training schedule helps the model to tackle MC tasks and achieve comparable results to human performance on SQuAD dataset [3,4].

Previous work [5] indicated that doing MC on spoken content is much more difficult than on text content, because speech recognition errors have catastrophic impact on MC. On the other hand, an end-to-end model for spoken language task, such as spoken language understanding (SLU) and speech to speech translation, is promising to solve the ASR error propagation problem, although by now the performance of most of the end-to-end models is still not as good as the performance of the corresponding natural language task.

In our work, we proposed SpeechBERT, a pre-trainable model of generic representation for speech and text tasks. By combining the pre-trained language models with a spoken audio encoder, our end-to-end model can circumvent the negative impact caused by cascading ASR and QA models that are trained separately. First, we pre-train the SpeechBERT model both on text corpus and speech audio to enable the model to extract useful semantic features on both speech and text content. When fine-tuning on the MC task, the whole model can be jointly optimized without an error bottleneck caused by ASR. As the first work for end-to-end spoken question answering (SQA), our model gets the results close to the performance of cascading ASR and QA models. Although the room for improvement exists, it is a good first step towards end-to-end SQA.

2. RELATED WORKS

2.1. Speech Segment Embedding
In a series of investigations, people want to extract semantic embeddings from speech feature segments given predefined segment boundaries. Speech2Vec [6] used a speech feature level sequence-to-sequence network to imitate the training process of skip-gram or CBOW in Word2Vec [7] to help the model extract more semantic feature. Unsupervised segmentation method [8] was also proposed to jointly learn to segment spoken words and extract the speech embeddings. After learning the speech segment embeddings, the following works [9,10] tried to map them to a regular word embedding, as we do in bilingual embedding mapping [11], either in supervised (using paired seeds) or unsupervised (using Generative Adversarial Networks [12]) approaches. Promising as these works sound, by now the mapping qualities are far from that of a bilingual embedding mapping case. The correctness of the mapping between speech and text embeddings is at most near 25% [10] even with the aid of oracle boundaries. That indicates the difficulty of disentangling semantic information from noisy speech signals without a supervised ASR model.

We also used the speech segment embedding concept in this work. The method will be introduced in 3.2.2. However, because the baseline methods of the SQA tasks already include a supervised ASR model as front end, we do not impractically pursue a fully unsupervised method for speech content extraction. Instead, we use the labels of what words are exactly represented by speech segments in the pre-training stage.

2.2. End-to-end Model for Spoken Language Tasks
Conventional methods for spoken language tasks need an ASR as a front-end module to extract semantic information from speech signals into plain text. The output of ASR will be treated as natural language data and fed into regular NLP models for downstream tasks. The end-to-end model aims to tackle the whole task from speech level features without cascading the ASR model. The end-to-end model has the benefits: 1) it optimizes the metric of the final task directly, instead of optimizing toward different targets for ASR and NLP models separately 2) it avoids error propagation problem caused by ASR bottleneck 3) the direct exposure of speech information to downstream models can help the model to capture useful information that is not shown in text transcripts.

Regular spoken language understanding (SLU) tasks like intent classification and slot filling have been explored in end-to-end methods [13,14,15,16] wider recently compared to spoken question answering (SQA) tasks [5]. However, these two tasks are at different levels of difficulty. SLU task is a sentence-level classification problem that needs to fill the slots from pre-defined classes by extracting some local information in a short utterance. As the literal
meaning of the short utterance is extracted by the SLU model, it is not far from making a correct classification. Compared to SLU, the inputs in the SQA task are much longer spoken paragraphs. Besides understanding the literal meaning, the SQA model needs to organize the global information first because the sophisticated reasoning in the paragraph is required to answer the questions. Fine-grained information is also needed to predict the exact position of the answer span from a very long context. Therefore, we will solve it by pointer networks [17] instead of classification models. These are the reasons why SQA would be a harder problem than SLU.

3. SPEECHBERT

Based on the BERT [1] model, we extend the BERT architecture with a speech segment encoder, which functions as an alternative module of the ASR model. Instead of recognizing words, the speech segment encoder aims to directly find good speech representations that can be fed into the BERT model, makes it possible to process text and speech in a shared BERT model. The model architecture and training process in illustrated in Figure 1.

3.1. BERT for Text Pre-training

BERT is a multi-layer Transformer [18] model. For the text part, given a token sequence $X_{text} = \{x_1, x_2, ..., x_n\}$, we represent the tokens with vectors $E_{text} = \{e_1, e_2, ..., e_n\}$. Then we sum $E$ and position embeddings and sentence segment embeddings to get $E_{text}$. Then $E_{text}$ will be fed into the multi-layer Transformer. At the output layer of BERT, we use the output features to do two tasks: masked language model (MLM) and next sentence prediction (NSP). MLM is to randomly replace 15% of vectors in $E_{text}$ with a special mask token vector $e_{mask}$ and predict the masked tokens at the same position of output features. NSP is to predict whether the tokens with different sentence segment embeddings are from successive sentences. However, some recent study [19, 20, 21, 22] have indicated that NSP does not improve performance but hurt it instead, so we removed this part and only trained MLM in our setting.

3.2. Speech Segment Encoder Pre-training

For the speech part, we have a speech feature sequence $X_{speech} = \{x_1, x_2, ..., x_t\}$ where $t$ denotes the number of acoustic features. The speech feature sequence is segmented into audio segments in Section 3.2.1. Given the word boundaries, we encode each segment to get speech version of word vectors $E_{speech} = \{e_1, e_2, ..., e_n\}$ where $n$ denotes the number of segments. The encoding method will be described in Section 3.2.2.

3.2.1. Speech Segmentation

Segmentation for Training Stage: To effectively extract semantic features in speech signals, we segment the Mel Frequency Cepstral Coefficients (MFCCs) sequences according to the predefined boundaries from forced alignment of an off-the-shelf ASR model.

Segmentation for Testing Stage: At the testing stage, we cannot access the ground truth labels to run forced alignment, so we use the ASR model to get the word pseudo-label sequence to run forced alignment. Even with wrong words in ASR recognition results, the boundaries found by forced alignment are usually corresponding to some other true words.

3.2.2. Phonetic-Semantic Joint Embedding

After getting the speech feature segments, we used an RNN sequence-to-sequence autoencoder to encode the segments to obtain phonetic embeddings that captures the phonetic information of acoustic words. The autoencoder training procedure makes audio that have similar phonetic features cluster together. However, due to the need to act as the inputs of the BERT model, simply fitting on pure phonetic information without considering the semantic relations between acoustic words is not desirable. Hence, we use the labels according to acoustic words to get the primary word vectors from the word embedding layer of BERT if words are not in out-of-vocabularies. For each paired audio segment and word, we add a loss term by calculating the L1-distance between the two paired vectors. By doing so, the autoencoder model can learn to fit the BERT input distribution for semantic word embedding, while keeping the acoustic information to reconstruct the original MFCC features as much as possible. This regularization helps the model learn a joint embedding space both for text and speech embedding, extracting semantic level features from speech directly.

To make the concept clear, we listed the loss terms to optimize. Given audio segment $x = \{x_1, x_2, ..., x_t\}$ as input features, the RNN encoder encode it as a vector $z$. The RNN decoder network maps $z$ to output $y = \{y_1, y_2, ..., y_f\}$. The encoder-decoder network are trained to minimize the reconstruction error:
After MLM pre-training, speech segment encoder pre-training, we used a bidirectional LSTM as the encoder and a single-directional LSTM as the decoder, both with the input size 39 (MFCC-dim) and hidden size 768 (BERT embedding-dim). Two layers of the fully-connected network are added at the encoder output to enable the encoder to transform the encoded information to fit BERT embedding space. We directly used the audio from Spoken SQuAD training set to train this encoder-decoder network.

4.2. Model Settings

4.2.1. Speech Segment Encoder

For speech segment encoder pre-training, we used a bidirectional LSTM Audio Encoder LSTM Audio Decoder...
Table 1: Experiment results. Ground truth is denoted by GT. ASR transcripts is denoted by ASR trans. GT segment means forced alignment on ground truth is used in training time. SQuAD development set and Spoken SQuAD testing set are denoted by SQuAD-dev and SpokenS-test, respectively.

| MODEL | SQuAD-dev (GT text) | SpokenS-test (ASR trans.) |
|-------|----------------------|--------------------------|
|       | EM       | F1       | EM       | F1       |
| BiDAF [26] | 58.4  | 69.9  | 37.02  | 50.9  |
| R-NET [27] | 66.34 | 76.20 | 44.75 | 58.68 |
| Mnemonic Reader [28] | 64.00 | 73.35 | 40.36 | 52.87 |
| Dr.QA [13] | 62.84 | 73.74 | 41.16 | 54.51 |
| FusionNet [29] | 70.47 | 79.51 | 46.51 | 60.06 |
| BERT [1] | 76.90 | 85.71 | 55.93 | 68.26 |
| BERT [1] w/o WordPiece | 72.24 | 82.59 | 48.71 | 66.27 |
| SpeechBERT | w/ MLM | 47.84 | 61.98 |
| SpeechBERT | w/o MLM | 46.11 | 60.23 |
| SpeechBERT w/ GT segment | w/ MLM | 49.34 | 63.27 |
| SpeechBERT w/ GT segment | w/o MLM | 47.90 | 61.97 |

As expected, the F1 and EM scores are dropped by about 1.7, showing the benefits of joint MLM pre-training before fine-tuning.

5.1.2. Quality of Segmentation

We wondered that if the performance is restricted by the quality of word boundaries which is found by forced alignment with ASR transcripts, which have 22.73% WER as mentioned in Spoken SQuAD [5]. To observe whether the quality of segmentation is the bottleneck of performance, we tested our model on the Spoken SQuAD testing set with ground truth text forced alignment used in training time, which would be more accurate than ASR transcripts forced alignment. However, the performance change higher only within 1.3 to 1.5 for F1 and EM scores. This showed that the quality of boundaries is not the main problem that causes performance lower.

5.2. Error Analysis

5.2.1. Out-of-vocabulary

Although out-of-vocabulary (OOV) is not an issue for spoken audio, we found the OOVs in question text part make the performance lower. As mentioned in the last section, to make the SpeechBERT model be able to process cross-modal input consistently both for speech and text in the same unit, we discarded the WordPiece [25] tokenizer and use the same vocabulary set as used in our SpeechBERT model. However, this modification disables the model to use WordPiece tokenizer to process name entities, which is crucial for answering correctly. To evaluate our conjecture, we trained a BERT with the same setting on the transcripts of Spoken SQuAD training set but using our new vocabulary set. Consistent with our hypothesis, the F1 and EM score consistently dropped by 2 to 7 for both Spoken SQuAD testing set and SQuAD dev set.

5.2.2. Comparison in different WER

Although our SpeechBERT model still not outperforms BERT trained on ASR transcripts, we can investigate whether SpeechBERT can beat BERT on the questions with higher recognition word error rate (WER). We split the questions into many groups according to different WER and tested the EM score for each group. We defined the “EM score ratio” as the ratio of the number of questions with EM = 0 to the number of questions with EM = 1. The higher ratio means the more questions are answered wrong compared to correctly answered questions. We calculate the ratios both for SpeechBERT and BERT, the result is shown in Figure 3. Obviously, BERT tends to have a lower ratio when WER is low and higher ratio when WER is higher, while SpeechBERT does not have this tendency and can still correctly answer the questions with extremely high WER.

6. DISCUSSIONS AND FUTURE WORKS

Though we achieved a reasonable performance on the SQA task, there is still a large room for future research. The first challenge is the usage of word boundaries. Although it is reasonable to use an off-the-shelf ASR model that acts as a segmenter under a supervised setting, it will be much more desirable if the boundaries can be provided by the end-to-end model itself. In conventional SLU tasks, it is possible to extract information from frame-level speech features to do classification tasks like slot filling. However, it will be an enormous challenge to use frame-level speech features for the SQA model which needs a pointer network to predict positions directly on very long frames. In this work, we choose an easier setting that focuses on embedding learning and language model pre-training to solve SQA with pre-computed word boundaries. One possible way to integrate segmentation into our approach is simply dividing audios by the voice intensity. Alternatively, previous work on simultaneous speech translation [30] has proposed algorithms to learn segmentation strategies that directly maximizes the performance of the machine translation system. Joint learning of segmentation and audio embedding that can mutually be enhanced by reinforcement learning [8] is another promising approach. This method can be adapted to the text and speech cross-modal language model pre-training in our work in the future.

The second goal for future research is cross-modal language model pre-training with few labels for speech corpus. While paired data was used in our pre-training stage, semi-supervised or unsupervised method can leverage the much larger unpaired corpora.

7. CONCLUSIONS

In this work, we proposed an end-to-end model for spoken question answering. Our model got the results close to the performance of cascading ASR and QA models. It is a stepping stone towards understanding speech content directly from speech information to solve QA problems.
