Continuous Integrate-and-Fire Speech-Text Fusion for Spoken Question Answering

Stanford CS224N Custom Project

Jeff Brown
Department of Computer Science
Stanford University
jeffrey.brown@stanford.edu

Abstract

Machine question answering (machine QA) is one of the cornerstone problems of modern NLP [1]. The problem comes in many variants: open-domain versus closed-domain; generative versus extractive; conversational versus singular; speech-based versus text-based. In this paper, we explore the closed-domain, extractive, singular speech-based question answering problem. First, we recalculate the standard ASR-BERT cascading approach using SoTA methods for both. Then we explore a novel continuous integrate-and-fire (CIF) method for aligning and transformer rich acoustic representation to semantic representations.

1 Key Information to include

- Sharing project: CS 224S

2 Introduction

Machine question answering (machine QA) is one of the cornerstone problems of modern NLP [1]. The problem comes in many variants: open-domain versus closed-domain; generative versus extractive; conversational versus singular; speech-based versus text-based. In this paper, we explore the closed-domain, extractive, singular speech-based question answering problem. As a closed-domain problem, a passage and question set are passed to a model and the model is tasked with answering the questions based on the passage. By restricting to the extractive task, the model's goal is to return the span of words in the passage that answers a posited question. As a singular task, the model does not need to retain dialog or conversation information to answer subsequent questions; rather, each question is independent of other questions and is dependent on the passage itself. As we are interested in the speech-based QA task, the passage passed to model is some instance of spoken language data. Typically, this means the audio recording of a spoken passage, a signal transform of such data, or a transcript, either by a human or with automated speech recognition (ASR) technology. I will refer to this task from here on as spoken question answering (SQA)

According to various sources [2] [3] [4], SQA is a more difficult problem than the typical text-based QA problem. While there are likely various causes, the current consensus is that that overwhelming source is ASR-driven information loss due to errors. The analyses of [2] [3] shown an average decrease of 19.2 F1 points across six different methods for machine QA when the models were trained on text data exclusively and then applied to both text-based QA datasets and spoken-then-transcribed versions of those same datasets. Solving the SQA problem opens doors to analyzing large amounts of spoken-language corpora in much the same way we do now text-based corpora. With competent success in the SQA task posed here, we are better suited to expand to open-domain, conversational, and generative QA tasks in the future.

One obvious proposal for augmenting performance at the SQA task is improving the performance of the ASR systems used to transcribe the audio data. In the [2] analysis, they cited an average

word-error-rate of 22.73% on a spoken-then-transcribed version of the Stanford Question Answering Dataset (spoken SQuAD). Given the rapid performance rise of the state of the art (SoTA) ASR systems since 2018, I suspect better systems will show much better performance. In this analysis, I determine the performance of the RoBERTa model for question answering on wav2vec2-ctc ASR to see if better ASR techniques lessen or eliminate the gap between SQA and text QA.

Another another class of SQA proposals go beyond better transcription and look to leverage both audio and text feature of the data to improve SQA performance. Generally, these proposal look to fuse audio-based phonetic and text-based semantic representations together to generate richer overall representations [2]. [3] [5]. There are many methods proposed for such fusion, and currently the best method is undetermined. Even it is the case that better ASR performance greatly augments performance on SQA tasks, such fusion methods may still be significantly valuable for more advanced spoken language processing tasks. For instance, we can imagine a joint sentiment-classification and SQA problem where the answer depends on both what was said and how it was said. Such fusion methods provide approaches to these more difficult yet useful problems.

In this analysis, I was interested in trying a novel approach to the SQA task: a continuous integrate-and-fire based fusion method proposed by Yi et al [6] for low-resource ASR. They use continuous integrate-and-fire to fuse wav2vec2 embeddings [7] with BERT embeddings [1] to generate robust, information-rich phonetic-semantic embeddings. Given that BERT-based approach have dominated the landscape of text QA in recent year [1], I was interested in how this method would perform in the context of SQA.

3 Related Work

In the QA literature, there is an existing precedent for fusing multi-modal information to enhance task performance. In Lu et al [8], they use a cross-attention to combine image and word presentation to create visio-linguistic representations for image-text question answering. In Zhang et al[9], they use multi-headed attention to aggregate semantic and entity representations to improve on various NLP tasks.

In each of these tasks, the authors leverage self-attention transformer techniques that have shown to produce SoTA results for various NLP tasks by learning infromation-rich representations through various self-supervised learning objectives. The inaugural form of this technique comes from Devlin et al's BERT (Bidirectional Encoder Representation from Transformers) model for language representation. BERT and other BERT-like model have shown the power of rich representations generate via contrastive learning (amongst other self-supervised tasks). Taking this to the next step, [7] showed that similar techniques could produce richly representative embeddings for audio data in wav2vec 2.0.

More specifically to the SQA task, many have proposed various techniques for audio-text fusion. With SpeechBERT, [3] use encoder-decoder model to generate audio-word embeddings from Mel Frequency Cepstral Coefficients with an $l1$ distance-constraint between audio-word embedding and BERT-based text-word embeddings to create jointly phonetic-semantic embeddings. You et al [5] propose a knowledge-distillation technique that leverages ground truth text transcription and audio data to train a model that can correct for ASR errors.

As an example of a technique using both audio and text to solve problems, [10] leverage ASR transcriptions and maximum-entropy based sentiment classification to determine the sentiment of YouTube videos. They show that they can achieve success on a difficult sentiment classification task even with a relatively high ASR error rate.

4 Approach

4.1 Baseline: BERT on wav2vec2 ASR Transcripts

As there are many different approaches for ASR, choosing one is a particular design choice fir this analysis. I decided to use the wav2vec 2.0 plus connectionist temporal classification decoding (CTC) [7] for ASR transcription. Using wav2vec 2.0 embeddings and CTC decoding alone achieves SoTA performance on the Librispeech dataset. Furthermore, it provides a nice benchmark for testing my CIF approach by using the same rich wav2vec vectors in an alternative way.

Take $\mathcal{W}$ to be the transform from audio samples of context passages x_a to wav2vec 2.0 embeddings $w \in \mathbb{R}^{s_w \times h_w}$, where s_w is the sequence length and h_w is the size of the output hidden layer. These hidden state are fed through a linear layer and decoded using CTC decoding to produce a sequence of characters c. These characters are concatenated with the text-characters of the context question questions, separated by a [CLS] token. This sequence of characters is tokenized, converted to embedding vectors $E_{\text{RoBERTa}}\mathbb{R}^{s_b \times h_b}$ and then fed into a RoBERTa model, producing a sequences of hidden states $H_{\text{RoBERTa}} \in \mathbb{R}^{s_b \times h_b}$ where s_b is the sequence length of tokenized RoBERTa inputs and h_b is the output hidden size of RoBERTa. To fine tune the model for extractive question answering, start and end vectors $S, E \in \mathbb{R}^h_b$ are multiplied against the output hidden states and a softmax function is applied, producing a distribution of start and end points for the question answer. During training, a negative log likelihood loss function is used to train the output vectors.

4.2 CIF-based BERT-wav2vec Fusion

Roughly speaking, the method proposed here approaches the SQA problem with an encoder-decoder model with a CIF bottleneck. The encoder is a the wav2vec 2.0 transformer pipeine; the decoder is the RoBERTa transformer pipeline. As I have already talked about wav2vec 2.0 and the motivations for using it, I will talk brief about RoBERTa model, and the elaborate on the CIF method and motivation.

4.2.1 RoBERTa

RoBERTa was developed by [11] as a better trained version of the original BERT model. The authors found that the training methods of the original BERT model severely under exploited its potential and more sophisticated training led to better overall results. As such, I wanted to use this novel model in place of the original BERT model for my experiments.

4.2.2 Continuous Integrate-and-Fire

Continuous integrate-and-fire (CIF) methods have a long history in neural machine learning, and more recently, some have used such technique to innovate novel methods for audio-text alignment for ASR [12] [6].CIF methods are used to transform between sequences of different lengths but shared information. Mathematically, take $x_1, \ldots, x_m$ to be m embeddings in $\mathbb{R}^{|}x|$ of sequential audio data and $w_1, \ldots, w_n$ to be n embeddings in $\mathbb{R}^{|w|}$ of sequential text data. A CIF transform $C : \mathbb{R}^{|x| \times m} \rightarrow \mathbb{R}^{|w| \times n}$ transforms one sequence of representations to another by constrained combination of consecutive representations.

There are many ways to define the function C to generate CIF embeddings. Generally, a function is applied to the original sequence of representations to create a corresponding sequence of weights. Consecutive vectors are (partially) combined such that the sum of the contributed representation weights adds up to 1. This enforces a soft and monotonic alignment strategy that forces the transformed inputs into the correct shape. As an example of this sort of process, imagine you have four vectors v_1, v_2, v_3, v_4 with representation weights $0.6, 0.3, 0.2, 0.9$ and you are transforming to a sequence with 2 vectors w_1, w_2. Then $w_1 = 0.6v_1 + 0.3v_2 + 0.1v_3, w_2 = 0.1v_3 + 0.9v_4$. Dong et al use a one-dimensional convolution layer to generate the representation weights. However, in this analysis, I was curious if the simpler method proposed in [6] would be sufficient.

In [6], to generate the representation weights $a_1, \ldots, a_m$ from a sequence of audio embeddings $x_1, \ldots, x_m$, they use $a_1 = x_1[-1], \ldots, a_m = x_m[-1]$. A sigmoid function is a applied to each of the weights generating $\alpha_1 = a_1, \ldots, \alpha_m = a_m$. As such, $\alpha_i \in [0, 1]$ for all i weights. Then, a new set of CIF embeddings $u_l, l \in 1, \ldots, n$ are generated by monotonic, partial linear combination of the audio embeddings by the representation weights. The monotonicity constraint is applied because a vector x_m can only be added to a CIF embedding u_l if all other vectors $x_t, t < m$ have been added to $u_s, s \leq l$, The partiality constraint is applied because some audio embeddings will fall on the boundary between u_l and u_{l+1}; the representation weight must be split between the two embeddings.

To given the model a change to adjust to the CIF embeddings, during pretrainig, I randomly sample from the CIF embeddings and RoBERTa word embeddings and a varying rate $p(step)$ that is a function of the optimization step. I feed these embeddings in at the first layer of the RoBERTa model as surrogate word vectors.

5 Experiments

5.1 Data

To pretrain the wav2vec-CIF encoder, I use audio and transcript data from the Librispeech English 100-hour clean dataset [13]. The audio data is 16 kHz mono spoken utterances and can be loaded a one-dimensional Pytorch tensors. The text data corresponds to the ground truth transcription of the audio data. The output of RoBERTa is decoded into text and compared to these ground-truth transcripts. To fine tune on the SQA task, I put together novel dataset. While previous methods used Spoken SQuAD [2] to approach this task, this data set gave me significant trouble due to data corruption and overall size. As such, I used a combination of the Spoken Wikipedia Corpora [14] and the Stanford Question Answering Dataset (SQuAD) [15]. The former contains audio-recordings of spoken wikipedia articles with force-aligned transcriptions; the latter contains human generate question about Wikipedia articles. Between two, I found 20 overlapping Wikipedia pages containing 6687 question-answer pairs. I split the audio recording from the Spoken Wikipedia Corpora into chunks according the context passages in the SQuAD dataset.

5.2 Evaluation method

To evaluate the pretraining of the wav2vec-CIF-BERT model, I computed the overall character error rate (CER). The value of this metric is to see how well the CIF transform learns to convert to the RoBERTa input embeddings. To evaluate the baseline SQA tasks, I kept track of F1 word scores. For the wav2vec-CIF-bert method, I kept track of character overlap (CO) between the extracted and gold answers. This is a proxy for the better audio overlapping score (AOS) metric developed recorded in [2]. While CO is less accurate than AOS for measuring true answer overlap, the hope is that is a reasonable proxy to the F1 score.

5.3 Experimental details

5.3.1 Baseline

To get the model for wav2vec with CTC decoding, I used the Wav2Vec2ForCTC model from Huggingface with pretrain extension "facebook/wav2vec2-base-960h". For RoBERTa question answering, I used the RobertaForQuestionAnswering model loaded from pretrain extension "roberta-base." To get naive metrics for the performance of this cascade pipeline, I did not do any further training on either model and froze all of the parameters in each model.

5.3.2 wav2vec-CIF-BERT Pretraining

For pretraining, I loaded the Wav2Vec2ForCTC model with pretrain extension "facebook/wav2vec2-base-960h" from Huggingface. However, in this version I left weights unfrozen. After the CIF layer, there is a fully connected layer to match the hidden layer size of RoBERTa. For the RoBERTa model, I used the RobertaModel loaded from pretrain extension "roberta-base" from Huggingface. I froze the weights of the RoBERTa model. During training, I used four Tesla V100 GPUs with Distributed Data Parallel. I used gradient accumulation to generate batch sizes of 32. I trained for 3 epochs used an AdamW optimizer. I used a learning rate scheduler with a 500 step warm up phase until 5e-5 and exponential decay after 42000 steps. I decrease the value of p from 0.9 to 0.2 over the first 1000 steps and set it constant there afterward.

5.4 wav2vec-CIF-BERT Question Answering

For this phase, I froze all the model weights from pretraing and intitialized a new output linear layer for the start and end vectors. I used Adam optimization with a learning rate of 5e-5 during training, and trained for 3 epochs.

5.5 Results

In the baseline, I found the WER to be 9.43%. The corresponding F1 scores were 65.3%. After pretraining, the wav2vec-CIF-BERT model produced CER of 153%. The CO measure for question answegin using wav2vec-CIF-BERT was 5.6%.

While I was pleasantly surprised by the impressive performance of the ASR-BERT cascading system using wav2vec, the model I explored here severely underperformed. Looking into the decoded outputs of the BERT decoder shows that the model is effectively producing gibberish. As such, the CER overlap score is essentially a function of chance.

6 Analysis

Given the impressive WER of wav2vec, I was not surprised by the impressive performance on teh question answering task. Many of the mistakes do not properly count as question-answering errors but rather pipeline errors. For instance, whenever the answer to the question is a number, the audio data for the entry reads out the number but the answer is the numeric version of that number. Therefore, it is counted as an error under the guideline of the F1 score when the model goes to the start of the word number rather than the numeric number. A better approach would be to use the AOS metric to compute time overlap. If the answer were assigned to the same time window, we would know whether or not the model actually extracted the correct answer.

The performance of the novel method proposed here was greatly disapointing. As such, I sought to understand why the method performed so poorly here yet did so well in [6]. After going over their analysis, I realized the the auxiliary loss function that Yi et al use likely contribute heavily to the success of their method. In addition to compute the cross entropy loss from the logits of the BERT decoder, they are use CTC loss against the output of the wav2vec module and cross entropy loss against the logits of the fully connected layer following the CIF layer. In there loss weighting scheme, they actually weight the loss from BERT-wav2vec fusion lower than the other loss functions. As a result, the model is not appropriately penalized for poor wav2vec to BERT mapping because it is decreasing the other two losses. These other two losses are known to produce good results for ASR: wav2vec with CTC is how the original wav2vec 2.0 paper achieve SoTA results. Given this, I wanted to see if I could more directly compute the representational power of this form of CIF embedding.

As such, I made a new model where, instead of trying to maximize the character accuracy of BERT decoder using CIF embeddings, I tried to minimize the mean squared error between the CIF embeddings and the ground truth RoBERTa word vectors. I found that the model quickly plateaued in the MSE loss, showing that this scheme was never going to be sufficient for from transforming wav2vec embeddings to BERT embeddings.

7 Conclusion

In conclusion, I showed that better ASR and semantic representations significantly improve performance on the SQA task. While I experimented with a novel method for SQA through CIF embedding transformations, I showed that this particular implementation is insufficient for the task at hand.

References

[1] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186, 2019.

[2] Chia Hsuan Li, Szu Lin Wu, Chi Liang Liu, and Hung Yi Lee. Spoken SQuAD: A study of mitigating the impact of speech recognition errors on listening comprehension. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018-Septe:3459–3463, 2018.

[3] Yung Sung Chuang, Chi Liang Liu, Hung Yi Lee, and Lin Shan Lee. SpeechBERT: An audio-and-text jointly learned language model for end-to-end spoken question answering. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020-Octob:4168–4172, 2020.

[4] Chia Chih Kuo, Shang Bao Luo, and Kuan Yu Chen. An audio-enriched BERT-based framework for spoken multiple-choice question answering. *Proceedings of the Annual Conference of the*

International Speech Communication Association, INTERSPEECH, 2020-Octob:4173–4177, 2020.

[5] Chenyu You, Nuo Chen, Fenglin Liu, Dongchao Yang, and Yuexian Zou. Towards Data Distillation for End-to-end Spoken Conversational Question Answering. pages 1–13, 2020.

[6] Cheng Yi, Shiyu Zhou, and Bo Xu. Fusing Wav2vec2.0 and BERT into End-to-end Model for Low-resource Speech Recognition. (January):1–5, 2021.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *arXiv*, (Figure 1):1–19, 2020.

[8] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv*, pages 1–11, 2019.

[9] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERniE: Enhanced language representation with informative entities. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 1441–1451, 2020.

[10] Lakshmish Kaushik, Abhijeet Sangwan, and John H. L. Hansen. SENTIMENT EXTRACTION FROM NATURAL AUDIO STREAMS Lakshmish Kaushik , Abhijeet Sangwan , John H . L . Hansen Center for Robust Speech Systems (CRSS), Eric Jonsson School of Engineering ,. *Audio*, pages 8485–8489, 2013.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, (1), 2019.

[12] Linhao Dong and Bo Xu. CIF: Continuous Integrate-And-Fire for End-To-End Speech Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:6079–6083, 2020.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[14] Arne Köhn, Florian Stegen, and Timo Baumann. Mining the spoken wikipedia for speech data and beyond. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[15] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad, 2018.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc., that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.