QUESTION ANSWERING FOR SPOKEN LECTURE PROCESSING

Merve Ünlü[a,b] *Ebru Arisoy*[c] *Murat Saraçlar*[a]

[a] Boğaziçi University, Istanbul, Turkey
[b] Galatasaray University, Istanbul, Turkey
[c] MEF University, Istanbul Turkey

ABSTRACT

This paper presents a question answering (QA) system developed for spoken lecture processing. The questions are presented to the system in written form and the answers are returned from lecture videos. In contrast to the widely studied reading comprehension style QA – the machine understands a passage of text and answers the questions related to that passage – our task introduces the challenge of searching the answers on longer text where the text corresponds to the erroneous transcripts of the lecture videos. Our initial experiments show that searching answers on longer text degrades the performance of the QA system drastically. Therefore, we propose splitting the transcriptions of lecture videos into short passages and determining passage-question matching using question aware passage representations. The proposed approach lets us utilize competitive neural network-based reading comprehension models for our task and improves the performance of the developed QA system.

Index Terms— Spoken question answering, spoken lecture processing, automatic speech recognition.

1. INTRODUCTION

The dramatic increase in the availability of online video lectures provides flexible and easily accessible learning opportunities to learners outside the classroom. Online video lectures have been widely used in formal education as well as in life-long learning. The widespread usage of lecture videos introduces the need for effective dissemination and utilization of video lectures, which can be accomplished through speech and language processing technologies.

Automatic Speech Recognition (ASR) has been widely used to automatically transcribe academic lecture recordings to enhance the learning experience of students as well as to increase accessibility for the hearing impaired students [1, 2, 3]. In addition to speech recognition, speech retrieval has also been used to facilitate learning through academic lecture recordings [4, 5]. Complementary to the previous research efforts, we developed a question answering system for spoken lecture processing.

The main idea in question answering is to automatically find the relevant answers to textual or spoken questions from a text or spoken document. Especially, the machine comprehension task – understanding a passage of text and answering questions related to that passage – has been investigated by many researchers and significant improvements have been obtained on the SQuAD datasets [6, 7]. Thanks to the neural network-based approaches used in machine comprehension [8, 9, 10, 11], the performance of the available systems are getting closer to the human performance. Machine comprehension task has also been investigated on spoken content [12, 13]. Spoken machine comprehension is a more difficult problem than machine comprehension on text both due to the erroneous transcripts provided by ASR systems and limited amount of available datasets.

In our research, we developed a QA system that focuses on finding the relevant answers of written questions from lecture videos. Note that the proposed task is more challenging than the spoken machine comprehension task due to the following reasons: i) The concept of a passage and the questions related to that passage are not explicitly determined in the lecture videos. In general, a passage can correspond to the whole lecture video transcription. ii) Answers to the questions can be longer than couple of words, i.e., on average 22 words per question in our lecture test data. iii) The task is very domain specific so it requires in-domain QA data for training the models. In order to tackle some of these issues, we first generated a small machine comprehension training dataset for the lecture domain where the domain of the training data matches with the domain of the test data. Then we defined pseudo passages for the transcriptions of the test data and determined passage-question relevancy using neural network-based question aware passage representations. This approach turns the testing phase of our task into machine comprehension style QA and allows us to use the competitive machine comprehension models developed for SQuAD dataset also for our task. Even though the questions have quite long answers, the answers are spans of words from the given passages and this makes the lecture QA task more tractable.

QA for lecture videos was also investigated in [14, 15].

This work is supported by the TÜBİTAK-ARDEB 3501 Program (Project No: 117E202).

However, these approaches are mostly based on extracting queries from questions using natural language processing and applying information retrieval techniques for finding the relevant answers. A recent work [16] also uses similar ideas for QA on text for the lecture domain. Unlike the previous QA systems developed for textual or spoken lecture content, our system is an end-to-end approach. To our knowledge, our work is the first in applying machine comprehension type QA to lecture videos.

This paper is organized as follows: Section 2 explains the methods used for QA and passage-question matching. The details of the ASR and the QA data and the experiments with this data are described in Section 3. Section 4 concludes the paper and gives some future research directions.

2. METHODS

This section summarizes the neural networks used for the question answering and passage selection systems.

2.1. MatchLSTM with Answer Pointer

MatchLSTM with Answer Pointer architecture is an end-to-end neural network proposed for the machine comprehension task [11]. The model achieved competitive results for SQuAD using attention and answer pointer mechanism. Given a passage and a question, MatchLSTM finds an answer span from the passage by sequentially processing the passage. There are three layers in the architecture.

1. **Preprocessing:** For a given passage **P** and a question **Q**, a (forward) uni-directional LSTM calculates the hidden representations ($\mathbf{H}^p$, $\mathbf{H}^q$), as shown below:

$$\mathbf{H}^p = \overrightarrow{LSTM}(\mathbf{P}), \mathbf{H}^q = \overrightarrow{LSTM}(\mathbf{Q})$$

2. **MatchLSTM:** This layer processes the passage sequentially and generates the attention vector in both forward and backward directions. For token *i* from the passage, the forward attention vector $\overrightarrow{\alpha_i}$ is:

$$\overrightarrow{\mathbf{G}}_i = \tanh(\mathbf{W^q H^q} + (\mathbf{W^p h_i^p} + \mathbf{W^r}\overrightarrow{h}_{\mathbf{i-1}}^{\mathbf{r}} + \mathbf{b^p}) \otimes e_Q)$$
$$\overrightarrow{\alpha}_i = \text{softmax}(\mathbf{w^T}\overrightarrow{G}_\mathbf{i} + b \otimes e_Q)$$

where $\mathbf{W^q}, \mathbf{W^p}, \mathbf{W^r}, \mathbf{b^p}, \mathbf{w}, b$ are the parameters to be learned and $\otimes e_Q$ repeats the vector or scalar for Q times to match the matrix or vector dimensions for summation. The hidden state $\overrightarrow{\mathbf{h}}_i^r$ is calculated as :

$$\overrightarrow{z}_i = \begin{bmatrix} \mathbf{h}_i^p \\ \mathbf{H}^q \overrightarrow{\alpha_i}^T \end{bmatrix}$$
$$\overrightarrow{\mathbf{h}}_i^r = \overrightarrow{LSTM}(\overrightarrow{z}_i, \overrightarrow{\mathbf{h}}_{i-1}^r)$$

$\overrightarrow{\mathbf{H}}^r \in \mathbb{R}^{l \times P}$ contains the hidden states $[\overrightarrow{\mathbf{h}}_1^r, .., \overrightarrow{\mathbf{h}}_P^r]$ as columns where l is the hidden dimension and P is the passage length. Using the same calculations in reverse order $\overleftarrow{\mathbf{H}}^r \in \mathbb{R}^{l \times P}$ is obtained. The concatenation of these two matrices gives the representation $\mathbf{H}^r$.

3. **Answer Pointer:** This layer is based on the Pointer Network [17]. Answer pointer layer takes the representation $\mathbf{H}^r$ as the input and predicts the start and the end tokens of the answer in the given passage. The word sequence between these two tokens is considered to be the answer.

2.2. Passage-Question Relevance Scoring

We propose a simple one hidden layer neural network to obtain a score between the passages and the question using the internal representation $\mathbf{H^r}$ from MatchLSTM. The last and the first hidden state vectors are extracted from forward and backward representations ($\overrightarrow{\mathbf{h}}_P^r$, $\overleftarrow{\mathbf{h}}_1^r$) respectively and concatenated to form the input vector $\mathbf{h_{rel}}$. For a passage and a question, the network calculates a score as shown below:

$$\mathbf{D} = \tanh(\mathbf{W_{rel}^h h_{rel}} + \mathbf{b_{rel}^h})$$
$$\theta = \text{sigmoid}(\mathbf{W_{rel}^o D} + b_{rel}^o)$$

where $\mathbf{W_{rel}^h}, \mathbf{b_{rel}^h}, \mathbf{W_{rel}^o}, b_{rel}^o$ are the parameters. The network is trained using positive and negative passage-question pairs to optimize binary cross entropy and to predict whether the given passage contains the answer to the given question. The scores from the network are used to select a single passage (with the maximum score) for each question.

3. EXPERIMENTS

This section explains the ASR and QA experiments as well as the details of the data used in these experiments.

3.1. Data

The lecture videos used in this research were prepared for flipped learning at MEF University. Each lecture video is a short clip, on average 5 minutes long, explaining basic concepts about the lecture topic. The lecture videos were shot in the recording studio of the university. The synchronized audio was recorded with a high quality close-talking microphone.

In our research, we used the lecture videos coming from 4 different courses from Electrical and Electronics Engineering Department for the acoustic data. These videos were prepared by the same instructor so there is a single speaker in the acoustic data. The 15 lecture videos for the "Signals and Systems" course were set apart as the test data (1.2 hours) and the other videos were used as the acoustic training data (2.7 hours) for the ASR system.

Paragraph: Now just as with the Fourier transform there are a number of properties of the Laplace transform that are extremely useful in describing and analyzing signals and systems. For example one of the properties that we in fact took advantage of in our discussion last time was the linearly the linearity property which says in essence that the Laplace transform of the linear combination of two time functions is the same linear combination of the associated Laplace transforms. Also there is a very important and useful property which tells us how the derivative of a time function rather the Laplace transform of the derivative is related to the Laplace transform in particular the Laplace transform of the derivative is the Laplace transform x(t) multiplied by s and as you can see by just setting s equal to j omega in fact this reduces to the corresponding Fourier transform property.

Question 1: What is the linearity property in the Laplace transform?
Answer 1: The linearity property which says in essence that the Laplace transform of the linear combination of two time functions is the same linear combination of the associated Laplace transforms.

Question 2: How is the Laplace transform of the derivative of a time function related with the Laplace transform of this time function?
Answer 2: The Laplace transform of the derivative is the Laplace transform x(t) multiplied by s.

Fig. 1. Question-answer pairs for a sample paragraph. Answers are consecutive word sequences from the given passage.

For building the language model for the ASR system, we collected text data related with the test lecture domain, mainly the reference transcriptions of the "Signals and Systems" course offered in MIT OpenCourseWare [1]. These reference transcriptions contain around 100K words. In addition to these transcriptions, we also used the reference transcriptions of the acoustic model training data (22.3K words) and the text coming from the lecture slides of the test data (3.2K words).

Even though the amount of acoustic and text data used in building the ASR system is limited, having the same speaker both in the training and the test data, as well as collecting text from the same lecture domain make the system plausible.

Since the domain of the publicly available QA datasets are very different than our lecture domain, we also generated a new dataset for training the QA system using the reference transcriptions of the MIT OpenCourseWare "Signals and Systems" course videos. These reference transcripts were first divided into 1309 short passages. We went through all the passages and generated 310 question-answer pairs for 259 passages. Answers were selected as consecutive word sequences from these passages. In the training data, the average passage length is 72 words, the average question length is 11 words and the average answer length is 24 words. A sample passage with question-answer pairs from the training data is given Figure 1. For QA test data, we also annotated the reference transcriptions of the test data and obtained 175 question-answer pairs. In the test data, the average question length is 11 words and the average answer length is 22 words. Compared to the popular SQuAD1.0 dataset where the dev partition passages contain on average 123 words and answers contain on average 3 words, our lecture QA dataset introduces a more difficult task mainly due to having long answers (on average 22 words) and long passages (on average 652 words) for each lecture video. Note that the transcription of each lecture video can be considered as a single passage.

[1] https://ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/video-lectures/

3.2. Results

3.2.1. Automatic Speech Recognition

We built an ASR system using the acoustic and the text data explained in Section 3.1. The acoustic models were trained using the Kaldi toolkit [18]. The language model is a 4-gram language model trained using the SRILM toolkit [19]. The word error rate (WER) results for 3 different acoustic models (GMM-si: speaker independent GMM, GMM-sa: speaker adaptive GMM and DNN: deep neural network model) were obtained as 13.0%, 10.5% and 6.7% respectively. In order to evaluate the lecture QA system for different WERs we report the performance of all three models.

3.2.2. Question Answering

The QA system was implemented using PyTorch 0.4.1 [20]. The lecture QA system is a more challenging task than the reading comprehension style QA task either on text or spoken context due to long passages coming from the ASR transcripts and questions with long answers. In order to emphasize these challenges and show the effectiveness of the proposed passage-question relevancy approach, we prepared several train-test scenarios. The QA model for each scenario was trained for 30 epochs with 32 samples in one batch using 150-dimensional hidden vectors. All models were tested both on the reference transcriptions and the erroneous ASR transcriptions of the test data. The question-answer pairs for the ASR transcripts were obtained by aligning the ASR transcripts of the test data with the corresponding reference transcriptions. The train-test scenarios are as follows:

1. **short - short:** The QA system was trained with the training data explained in Section 3.1. Test data was manually divided into short passages based on the transcriptions corresponding to each lecture slide in the video. This results in 144 test passages, 94 of which

Table 1. QA results for different train-test scenarios. GMM-si, GMM-sa and DNN represent ASR transcriptions obtained with these acoustic models and Ref represents the reference transcriptions.

Train-Test Scenarios	Test Set F1 Score							
	with known passage-question pairs				with passage-question pair selection			
	GMM-si	GMM-sa	DNN	Ref	GMM-si	GMM-sa	DNN	Ref
short - short	56.38	55.62	57.02	60.47	43.65	46.10	47.74	49.12
short - long	23.39	23.91	24.21	25.51	–	–	–	–
long - long	27.84	28.69	29.70	29.87	–	–	–	–
window - window	38.76	40.47	42.84	42.63	33.05	33.53	34.31	35.59

have associated questions. These 94 passages contain on average 81 words and 2 questions.

2. **short - long:** The QA model is the same with the one trained in the first scenario. However, each passage in the test data corresponds to the transcription of a lecture video. This results in 15 test passages where each passage contain on average 652 words and 12 questions.

3. **long - long:** Short passages in the training data were concatenated to obtain longer passages containing on average 686 words. The QA system was trained with this new training data. The test data is the same with the one in the second scenario.

4. **window - window:** The train and the test data were divided into pseudo passages using around 200 consecutive words per passage by taking sentence boundaries into account. This results in 44 test passages, of which only 2 do not have any associated questions. Each passage contains on average 233 words and 4 questions.

The QA results with these scenarios are given in Table 1. In the table, GMM-si, GMM-sa and DNN columns contain the results of the QA systems on the ASR transcripts with 13.0%, 10.5% and 6.7% WERs respectively. The Ref column contains the results of the same systems on the reference transcriptions. The overlap between the predicted and the ground truth answers were measured using word level F1 score.

When the passage-question pairs are known, in other words, the questions are searched in their related passages, we obtained the best result on the reference transcripts with the short-short scenario where the test passages were obtained manually. The performance of the QA system degrades with increasing WER. Note that this scenario makes the assumption that we know the passage that contains the answer to each question, which does not hold in a real test scenario. This assumption holds in part for the short-long and long-long scenarios where each test passage contains the transcription of a whole lecture video. However, the QA performance for these scenarios degrade significantly mostly due to the increasing passage length. Using pseudo passages, window-window scenario, also requires knowledge of the window to be searched and degrades the performance compared to the short-short scenario. This can be due to increasing the passage length as well as introducing irrelevant words to the passages due to using almost fixed length pseudo paragraphs.

Even though short-long and long-long scenarios are somewhat realistic, a more realistic scenario requires searching questions in all the related videos but using longer test passages degrades the QA performance further. Therefore, we converted the task into a more realistic, but still tractable, scenario by assuming that questions related to one of the chapters of the lecture will be searched only in the videos of the same chapter. To make the task tractable, we used the window-window scenario together with the proposed passage-question relevance scoring algorithm, explained in Section 2. First, questions were assigned to a single pseudo passage based on the relevance scores and then the locations of the answers in this passage were determined. This two-stage approach improves the QA performance (the last four columns at window-window row) compared to the short-long and long-long scenarios. However, there is room for improvement as the selection mechanism is imperfect and the performance is better when the short passage or window containing the answer is known (see the columns with known and selected passage-question pairs in Table 1) which is of course not realistic.

4. CONCLUSION

In this paper we developed a QA system for spoken lectures in the signal processing domain. The system is based on competitive neural network based reading comprehension models. We proposed a passage-question matching stage to handle a realistic scenario where the answer for each question is searched in a chapter of the course lectures. We showed that the proposed system improves the performance and analyzed the degradation due to ASR errors.

In the future, we plan to extend our data to increase diversity. We also plan to make use of ASR transcripts for training the QA system. Finally we intend to add unanswerable questions in the spirit of SQuAD2.0 [7].

5. REFERENCES

[1] Lori Lamel, Gilles Adda, Eric Bilinski, and Jean-Luc Gauvain, "Transcribing lectures and seminars," in *Proceedings of Interspeech*, 2005, pp. 1657–1660.

[2] Isabel Trancoso, Rui Martins, Helena Moniz, Ana Isabel Mata, and M Céu Viana, "The LECTRA corpus–classroom lecture transcriptions in European Portuguese," in *Proceedings of International Conference on Language Resources and Evaluation*, 2008.

[3] Thomas Pellegrini, Helena Moniz, Fernando Batista, Isabel Trancoso, and Ramon Astudillo, "Extension of the LECTRA corpus: classroom lecture transcriptions in European Portuguese," *Speech and Corpora*, 2012.

[4] James R Glass, Timothy J Hazen, D Scott Cyphers, Ken Schutte, and Alex Park, "The MIT spoken lecture processing project," in *Proceedings of HLT/EMNLP on Interactive Demonstrations*, 2005, pp. 28–29.

[5] James Glass, Timothy J Hazen, Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proceedings of Interspeech*, 2007.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.

[7] Pranav Rajpurkar, Robin Jia, and Percy Liang, "Know what you don't know: Unanswerable questions for squad.," in *Proceedings of the Association for Computational Linguistics*, 2018.

[8] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," *CoRR*, vol. abs/1804.09541, 2018.

[9] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi, "Bidirectional attention flow for machine comprehension," *CoRR*, vol. abs/1611.01603, 2016.

[10] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, vol. 1, pp. 189–198.

[11] Shuohang Wang and Jing Jiang, "Machine comprehension using match-lstm and answer pointer," *CoRR*, vol. abs/1608.07905, 2016.

[12] Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee, "Odsqa: Open-domain spoken question answering dataset," 2018.

[13] Bo-Hsiang Tseng, Sheng-syun Shen, Hung-yi Lee, and Lin-Shan Lee, "Towards machine comprehension of spoken content: Initial TOEFL listening comprehension test by machine," *CoRR*, vol. abs/1608.06378, 2016.

[14] Jinwei Cao, J.A. Robles-Flores, D. Roussinov, and J.F. Nunamaker, "Automated question answering from lecture videos: Nlp vs. pattern matching," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005.

[15] Stephan Repp, Serge Linckels, and Christoph Meinel, "Question answering from lecture videos based on automatically-generated learning objects," in *Advances in Web Based Learning - ICWL 2008*, Frederick Li, Jianmin Zhao, Timothy K. Shih, Rynson Lau, Qing Li, and Dennis McLeod, Eds., Berlin, Heidelberg, 2008, pp. 509–520, Springer Berlin Heidelberg.

[16] Caner Derici, Yiğit Aydın, Çiğdem Yenialaca, Nihal Yağmur Aydın, Günizi Kartal, Arzucan Özgür, and Tunga Güngör, "A closed-domain question answering framework using reliable resources to assist students," *Natural Language Engineering*, vol. 24, no. 5, pp. 725–762, 2018.

[17] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 2692–2700. Curran Associates, Inc., 2015.

[18] Daniel Povey et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society.

[19] Andreas Stolcke, "SRILM–An extensible language modeling toolkit," in *Proceedings of ICSLP*, Denver, 2002, vol. 2, pp. 901–904.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.