DoSSIER at MedVidQA 2022: Text-based Approaches to Medical Video Answer Localization Problem

Wojciech Kusa[1†] **Georgios Peikos**[2†] **Oscar Espitia**[2]
Allan Hanbury[1] **Gabriella Pasi**[2]

[1]TU Wien, Vienna, Austria
{wojciech.kusa,allan.hanbury}@tuwien.ac.at
[2]University of Milano-Bicocca, Milan, Italy
{georgios.peikos,oscar.espitiamendoza,gabriella.pasi}@unimib.it

Abstract

This paper describes our contribution to the Answer Localization track of the MedVidQA 2022 Shared Task. We propose two answer localization approaches that use only textual information extracted from the video. In particular, our approaches exploit the text extracted from the video's transcripts along with the text displayed in the video's frames to create a set of features. Having created a set of features that represents a video's textual information, we employ four different models to measure the similarity between a video's segment and a corresponding question. Then, we employ two different methods to obtain the start and end times of the identified answer. One of them is based on a random forest regressor, whereas the other one uses an unsupervised peak detection model to detect the answer's start time. Our findings suggest that for this task, leveraging only text-related features (transmitted either verbally or visually) and using a small amount of training data, lead to significant improvements over the benchmark Video Span Localization model that is based on deep neural networks.

1 Introduction

Nowadays, the number of users that turn to the Web to satisfy their health-related information needs has grown significantly. However, providing the user with textual answers is insufficient for some particular information needs because, occasionally, these answers are hard to interpret correctly. Therefore, it could be useful if the answers are accompanied by a visual aid, i.e., a part of a video (video segment), that presents the answer. Such information needs are the scope of this work, where we focus on identifying those video segments that contain the answer to health-related user questions. These user questions are written in natural language and the corresponding answers are part of an instructional video; our goal is to create a system capable of locating the corresponding answer.

While the majority of the proposed works in the literature rely on deep neural models to allocate the relevant video segments to the answer (Yu et al., 2017; Anne Hendricks et al., 2017), we explore another alternative. Specifically, we aim to study the impact of using only textual features to find the answers in the video, which also implies reducing the requirements for the amount of training data. As input features we use the information transmitted verbally by the presenter in the form of video transcript and also extract the text embedded in the video frames.

Our main contributions are as follows:

- We develop two approaches that use only textual information and few training data, to tackle the task of answer localization for instructional medical videos.
- We show that both the visually (text presented in a video's frame) and verbally (transcripts obtained from the speaker's instructions) transmitted information can be used to locate the answer in medical instructional videos.

The remainder of the paper is organized as follows: Section 2 describes in detail the studied task and the related works. Section 3 presents our methodology and assumptions. In Section 4, we present the experimental setup, our baseline and our submissions. Finally, Section 5 presents the obtained results, followed by the conclusions drawn from our participation.

2 Task Description & Related Works

This work studies the task of video segment identification for medical videos, introduced as a shared task in Gupta and Demner-Fushman (2022). In

† Equal contribution.

Proceedings of the BioNLP 2022 workshop, Dublin, Ireland, pages 432–440

particular, given a medical or health-related question written in natural language, the system must provide the user with the video segment that contains the answer. The task focuses on instructional medical videos. A characteristic of these videos is that they deliver the key information to the user both visually and verbally (Gupta et al., 2022).

In visual question answering, identifying relevant video segments given a user's questions in a natural language is a task that requires processing of both textual and visual signals. As reported by Zhang et al. (2019), a system designed to tackle this problem consists of three components, namely, feature extraction, feature fusion and answer prediction. Previously published studies exploit standard embedding models to obtain text features (Tapaswi et al., 2016), and CNN based models to extract image features (Zhou et al., 2018). Liu et al. (2019a) introduced ETM-Trans which is a deep transfer learning approach that also addresses the issue of feature fusion. In the field of visual question answering, as reported in (Lin et al., 2021) the majority of the proposed techniques employ pre-trained models for image and language encoders. Another finding reported in (Lin et al., 2021), is related to the fact that only a small portion of the proposed approaches investigate their generality and interpretability.

The introduction of large-scale multimodal datasets covering both language and vision enabled the development of efficient deep neural network techniques that bridge the gap between language, and visual understanding (Lei et al., 2018, 2019; Tapaswi et al., 2016).

While the majority of the proposed methods in the literature are based on deep neural models, our approach leverages only the textual information that can be extracted from a video without the need for extensive training. It estimates the relevance of each video segment to a given question, and ultimately it returns the starting time and duration of the answer.

3 Methodology

Our methodology exploits the characteristics of the videos in the current task. Specifically, we extract a video's transcripts. The transcripts contain the text that one can hear during the video, its start time and its duration, and correspond to a specific video segment. Moreover, we enrich this information by adding the text presented in video segments (video frames), for instance text that contains the topic, the steps of an exercise, among others. Then, given a question, we estimate distinct similarity scores for every video segment using four different models that will be described in Section 3.2. At this point, two distinct approaches can be followed to identify the answer's starting time and duration. The first one employs a multi-output regression model that inputs the similarity scores for every video segment and outputs the starting time and duration of the answer. For the second approach we set the starting time equal to the starting time of the segment that has the highest similarity score, obtained by aggregating the similarity scores obtained by four models, and hard-set the answer's duration based on the training data. The following sections present the hypothesis and assumptions behind each step of our methodology.

3.1 Converting video to text

As mentioned by Gupta et al. (2022), instructional medical videos deliver the key information both visually and verbally. We hypothesize that the speaker mentions keywords during the video that are also present in the question. For example, a phrase such as: "In the following part I will show you how to perform the [name of a specific exercise]", where the *"name of the specific exercise"* can also be found in the user's question.

Secondly, we hypothesized that video frames might contain textual information that overlaps with the question's text. However, it is also possible that the information obtained from these frames is irrelevant; i.e. frames may contain the speaker's name or affiliation. All in all, we assume that the text extracted using the two approaches mentioned above can provide a strong indication of the answer's location.

Finally, we assumed that text is not equally distributed across the video. For instance, it is common that a speaker might make a pause, e.g. to demonstrate the instruction or to change the subject. When only a video's text features are used, it is possible that some parts of the video will have no representation. In order to mitigate this issue and also to further enrich the text representation, we experiment with merging consecutive transcript lines. We ensure that when doing this, we also shift the time that corresponds to the merged text.

3.2 Estimating text-question similarity

Having the text that corresponds to a set of sequential video frames, we estimate its similarity to the question using four different models. Specifically, we employ two relevance models widely used in Information Retrieval (IR) to estimate the query-document similarity. In addition, we employ two pre-trained neural language models that are based on the Transformer architecture (Vaswani et al., 2017). We encode the questions and textual features independently for each language model and then calculate the similarity scores using a cosine similarity measure. We perform a min-max normalization of the similarity scores for each model independently. We then create an $M \times N$ matrix that contains the aggregated similarity scores for each question-video and every video segment; where M is the number of the employed models, and N is the number of video segments.

3.2.1 IR models

Regarding the IR relevance models, we employed the BM25 relevance model and a language model with a Dirichlet smoothing to overcome the problem of missing terms, which is likely to occur due to the characteristics of the studied task. In particular, the problem of missing terms occurs because the duration of the instructional videos is short and therefore it contains only few words. These models rely their estimation on some collection-related statistics, e.g. a term's inverse document frequency; to estimate these values, the models exploit an index created by concatenating the videos' texts present in a training collection.

3.2.2 Neural language models

In our experiments we employ two different language models pre-trained using different datasets, that are available in the HuggingFace transformers library (Wolf et al., 2020), namely:

- The RoBERTa model (Liu et al., 2019b) trained on the MS MARCO dataset from the *sentence-transformers*[1] framework (Reimers and Gurevych, 2019).
- The MPNet model (Song et al., 2020) trained on the SNLI and MultiNLI datasets from the *sentence-transformers*[2] framework.

[1] https://huggingface.co/sentence-transformers/msmarco-distilroberta-base-v2

[2] https://huggingface.co/sentence-transformers/nli-mpnet-base-v2

3.3 Answer localization models

This section describes two different approaches to localization of the answer time: multi-output regression and peak detection.

3.3.1 Multi-output regression (MoR)

Having created the $M \times N$ matrix described above, the answer localization can be modelled as a regression problem. To this aim, we employed the Random Forest multi-output regression model to predict the answer's starting point and duration.

The employed regression model requires a fixed-size sequence to be used as input. However, the available videos, and hence their textual representation, have varying duration. As a result, one should normalize the input length across the whole dataset. To achieve that, we formulate a method of sampling the text-question similarity models to obtain the same length for every video-question pair. In particular, we split every video into B equally spaced bins. By using these bins, we create a fixed-size representation of every video in the dataset. For every bin, independently for each model, we calculate two values: the maximum and the median values of all text-question similarity scores within the timestamps of a particular bin. Consequently, our normalization approach generates $2M \times B$ input matrix, where M is the number of models and B is fixed for the whole dataset and it contains both the maximum and median values.

3.3.2 Peak detection (PD)

Peak Detection (PD) approach also utilizes the $M \times N$ matrix described in Section 3.2 to find the video segment which is the most relevant to the question. We hypothesize that the segment with the highest topical similarity could be identified shortly before or after the true start of the answer (Figure 1). This method takes the average of the similarity scores from all text-question similarity models for every segment, and then retrieves the segment with the highest score. After identifying the segment, the start and end time of the answer can be predicted using the following formula:

$$\begin{aligned} t_s^{'} &= t_s + \beta_1, \\ t_e^{'} &= t_s + \beta_2, \end{aligned} \tag{1}$$

where t_s is the timestamp of retrieved segment and β_n are two free-parameters that are used to estimate

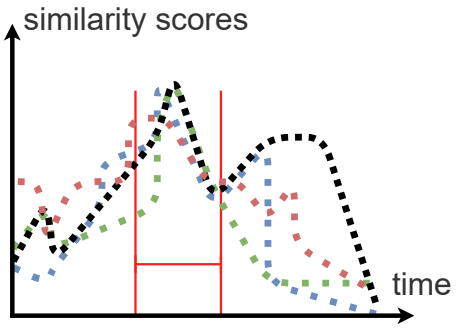

Figure 1: Illustration of the text-question similarity over a video time. Red lines mark the span of the correct answer. Maximum values of similarity for each similarity model are within the true answer span.

the duration of the answer.

4 Experiment setup and submissions

This section describes the dataset used to train and evaluate our approaches and our methodology for extracting text from videos. In addition, we provide the details of our submissions. Our code is publicly available[3].

4.1 Dataset

Model training and validation has been conducted on the MedVidQA dataset (Gupta et al., 2022), which consists 3010 questions from 899 unique videos in three different data splits, i.e., Train, Validation and Test. The submitted runs in the MedVidQA 2022 Answer Localization Shared Task (Gupta and Demner-Fushman, 2022) were evaluated on a new test dataset that consists of 153 questions covering 50 new YouTube videos, hereafter referred to as MVAL 2022.

Table 1 presents the number of videos for which we were able to extract the transcripts and the text in the video frame (embedded text) across the different datasets (rows 1-3). 98.5% of the videos contain some textual information. The missing 1.5% is primarily due to the private or protected videos for which it is not possible to obtain these features. In addition, Table 1 presents the mean and the median number of lines found in the transcripts and in the embedded text showing that medical instructional videos contain many verbal explanations and textual information embedded in the video frames.

[3] https://github.com/ProjectDossier/MedVid2022

4.2 Video to text

To extract the transcripts from a video we used the *youtube_transcript_api*[4] library. In cases where the transcript extraction was not feasible (1.5% of the videos), a placeholder text was assigned to the first second of the video. The obtained transcript lines were often just a set of words, split based on the speaker's pauses during the video, rather than complete sentences. In Section 3.2.1, we hypothesized that the problem of missing terms may occur. Indeed, it was found that various transcript lines contained only few keywords (due to speaker's pauses), in some cases only the stopwords. Therefore, for these cases, the obtained document representation was not accurate.

To overcome this issue, initially we tried to concatenate all the transcript text, and then, by using sentence splitting methods, create a set of sentences. However, due to the missing punctuation in many videos, this method was not accurate, and we decided to follow a simpler approach.

In particular, we proceed by merging subsequent transcript lines. For instance, a line i which contain the words: "now I will present" followed by a line $i+1$ containing "an exercise that helps with back pain" was merged into one single text. Moreover, we experiment with different levels of merging sequential transcript lines by joining two, three and four consecutive texts that generate three additional input representations. We refer to all the transcript features as *transcript-*n, where n is the number of original sequential transcript lines that were merged.

To download the videos we used the *pytube*[5] Python package. We use the offset of one second for the first frame as the beginning of the video is usually just a black screen. Also, we used the *tesseract*[6] engine to perform the optical character recognition (OCR) to extract the text from every video frame. Finally, we set the recognized text's duration to three seconds to follow the same data format as in the transcripts. The obtained features from this textual information are referred to as *ocr*.

An overview of all five different video-to-text representations used in our experiments is presented in Table 2.

[4] https://pypi.org/project/youtube-transcript-api/

[5] https://pypi.org/project/pytube/

[6] https://github.com/tesseract-ocr/tesseract

	MedVidQA			MVAL 2022	
	Train	Validation	Test	Test	Total
Videos (V)	800	49	50	50	949
V with transcripts	788 (98.5%)	48 (98%)	50 (100%)	49 (98%)	935 (98.5%)
V with embedded text	750 (93.8%)	48 (98%)	47 (96%)	49 (98%)	894 (94.2%)
Mean # lines in transcripts	133	142	124	123	140
Median # lines in transcripts	97.5	107.5	110.5	70	106
Mean # lines in embedded texts	20	16	25	18	17
Median # lines in embedded texts	9	8	15	9	9

Table 1: Statistics of the availability of textual information in medical informational videos. MVAL 2022 stands for MedVidQA 2022 Answer Localization Shared Task.

Feature name	Feature description	Start time	End time
transcript-1	Original transcript line i output from the video	s_i	e_i
transcript-2	Two consecutive lines of transcript merged together	s_i	e_{i+1}
transcript-3	Three consecutive lines of transcript merged together	s_i	e_{i+2}
transcript-4	Four consecutive lines of transcript merged together	s_i	e_{i+3}
ocr	OCR of the video frame i taken at second s every 3 seconds	s_i	$s_i + 3$

Table 2: Description of five different input features used in our work. s_i represents the start time of the i-th transcript line or the video frame.

4.3 Submissions

We submitted five runs for the MedVidQA 2022 Medical Visual Answer Localization (MVAL) Shared Task. A summary of our submissions is presented in Figure 2. In this section we describe these runs in detail.

4.3.1 Baseline: zero-shot extractive Q&A (1)

We use the DistilBERT-base-uncased model (Sanh et al., 2019), fine-tuned using knowledge distillation on the SQuAD dataset. We take the implementation from the HuggingFace transformers library[7]. As an input feature, we concatenate all the lines of *transcript-1* to create a consistent, single document representation of each video.

The model's output is text extracted from the video. Therefore, that extracted textual answer needs to be converted back to its start and end time. This can be done by locating its corresponding lines in the transcript. To achieve that, we employed the most greedy approach, i.e., selecting the whole transcript line if it contains at least one word from the extracted answer.

We noticed that the employed Q&A model could not correctly predict the textual answer to the question, and the retrieved answers are too short. We believe that this is because most videos do not exhibit the explicit textual answer to the question, but only the visual explanation. In order to mitigate this issue, we decided to test a simple parametrization model that stretches the predicted answer span:

$$t_s^{'} = \alpha_1 \cdot t_s,$$
$$t_e^{'} = \alpha_2 \cdot t_e, \quad (2)$$

where t_s and t_e are outputted start and end times of the answer from the Q&A system and α_n are estimated using the train dataset. After conducting an analysis on the validation dataset, we select the following values for the parametrization of the results: $\alpha_1 = 0.35$, $\alpha_2 = 0.90$.

4.3.2 Multi-output regression (2)

Our submission (2) is described in detail below:

1. We use all five input features to calculate the text-question similarity using the four models described in Section 3.2. For the BM25 and the statistical language model, the index was created using the Train data.

[7] https://huggingface.co/distilbert-base-uncased-distilled-squad

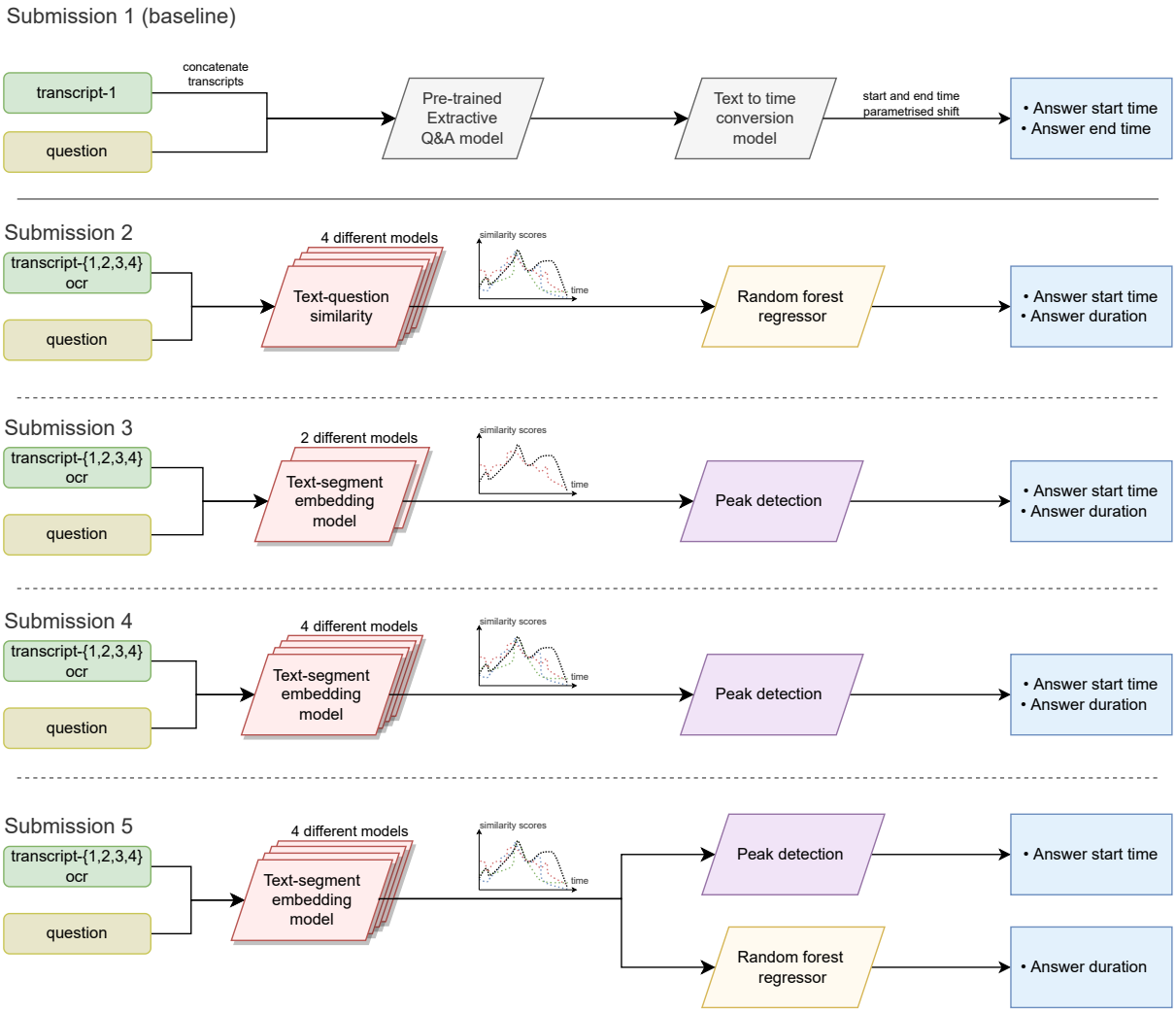

Figure 2: Summary of our five submitted runs.

2. We perform a min-max normalization independently for each model, aggregating scores from every input feature.

3. We conduct an input normalization as described in Section 3.3.1, so that the size of each feature vector is constant across all the training samples.

4. We train the random forest regressor model to predict the start time and duration of the answer.

We use the random forest regressor implementation from *scikit-learn* library (Pedregosa et al., 2011) with max depth equal to 10 and 40 estimators.

4.3.3 Answer start-time detection (3) & (4)

Submissions (3) and (4) also use the first two steps as in the submission (2) to calculate the text-question similarity and perform a min-max normalization. For submission (3), we use only the RoBERTa and the statistical language model with a Dirichlet smoothing. For submission (4), we use all four text-question similarity models.

This is followed by the step of peak detection by selecting the time when the average similarity of all models is the highest, as described in Section 3.3.2. Instead of using the start time of a segment, we take the center point of the selected segment as the most plausible starting point of the answer: $t_s = (s + e)/2$.

Finally, we calculate the answer start and end time by using Equation 1. Based on the experiments on the validation set, we select $\beta_1 = -6$ to overcome the shift between the true answer start and the similarity score peak. We use $\beta_2 = 62$ which corresponds to the mean answer duration on the training dataset.

Run	Source	Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
	Gupta et al. (2022)	VSL-BASE (FPL 800)	21.93	12.25	5.80	20.15
		VSL-QGH	25.81	14.20	6.45	20.12
1	Extractive Q&A		21.29	9.68	3.87	18.92
2	MoR		31.61	15.48	4.52	18.62
3	PD (2 models)		46.45	**29.03**	10.97	29.92
4	PD (4 models)		**48.39**	**29.03**	**11.61**	30.33
5	PD (4 models) start + MoR duration		47.10	27.74	10.97	**30.67**

Table 3: Performance comparison of our submissions on MedVidQA Test dataset from (Gupta et al., 2022).

4.3.4 Ensemble model (5)

Our last submission (5) is an ensemble model. It uses the prediction of the start time from the Peak Detection (4 models) – submission (4) and the duration from the multi-output regression model – submission (2). This method overcomes a limitation of the previous approaches, i.e., the constant parameter β_2 that defines the answer's duration (see Equation 1). In the previous approaches, this parameter had a constant value across all video-question pairs. In contrast, in this approach, the β_2 parameter for every question takes a unique value predicted by the random forest regressor used in the submission (2).

5 Evaluation and results

In this section we present the results on both the evaluation and test datasets.

5.1 Evaluation measures

We follow the evaluation measures proposed by Gupta et al. (2022) that have been chosen as the official metrics for the MedVidQA 2022 Shared Task. In particular, we evaluate our results using Intersection over Union (IoU) that measures the proportion of overlap between the predicted answer and the ground truth at three different thresholds, and mIoU that is the average of IoU calculated over a set of samples. Notice that MedVidQA adopts "R@n, IoU=μ", which denotes the percentage of questions for which, out of the top-n retrieved temporal segments, at least one predicted temporal segment intersects the ground truth temporal segment for longer than μ. Specifically, results are evaluated using $n = 1$ and $\mu \in \{0.3, 0.5, 0.7\}$.

5.2 Evaluation on MedVidQA

Validation results are presented in Table 3. Our baseline Q&A model (submission (1)), which initially was not able to retrieve any relevant information, after using parametrization it reaches 21.29 (IoU=0.3), which is on par with the performance of the Video Span Localization (VSL) benchmark model from Gupta et al. (2022). This shows that the first threshold could be reached even by a sub-optimal model whose predictions are shifted using two fixed parameters. Our best performing approach, Peak Detection (submissions (3) and (4)), achieves significant gains for each of three thresholds for the IoU measure, when compared to the best benchmark, i.e., the VSL model. Especially for the mIoU measure it obtains 10% more overlap on the Test data.

5.2.1 Impact of text extracted from the video frames

For some of the videos, the text extracted from the video frames had a significant impact on localising the correct answer.

Such an example can be seen in Figure 3. One

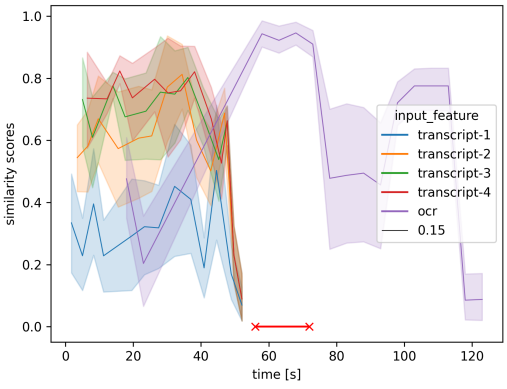

Figure 3: Mean text-question similarity plots for the Peak Detection approach with four models for question ID 2714 grouped by the input feature. Red line shows the span of a correct answer.

Run	Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
	Max	91.50	84.97	73.20	75.83
	Median	80.32	71.90	48.37	58.81
	Mean	76.04	60.80	40.37	55.79
1	Extractive Q&A	18.95	7.84	1.96	19.87
2	MoR	31.37	13.07	4.57	18.80
3	PD (2 models)	**40.52**	20.26	**10.45**	**25.26**
4	PD (4 models)	37.25	14.38	7.84	22.05
5	PD (4 models) start + MoR duration	33.33	**21.57**	9.15	23.54

Table 4: Performance comparison of the variants of our submissions on MedVidQA 2022 Test dataset. Runs 3, 4 and 5 did not contribute to the median and mean pool.

Features	IoU=0.3	IoU=0.7	mIoU
all	**48.39**	11.61	30.33
transcript-1	45.16	9.68	28.01
transcript-{1,2,3,4}	47.74	**12.26**	**30.55**
ocr	18.06	3.23	12.65

Table 5: Performance comparison of the Peak Detection approach using 4 models with different input features on MedVidQA Test dataset.

can observe that without the *ocr* feature, it is not feasible to identify the correct answer because the transcript features do not exist for the correct answer span.

To further quantify the impact of input features, we conducted an ablation study on our best performing approach: PD with four models (submission (4)). The results are summarised in Table 5. The model using all features achieves the highest IoU=0.3, which was our optimization goal. Removing the *ocr* feature slightly improves the results on IoU=0.7 and mIoU. Even though the text extracted from the video frames alone yields low results, it still can be a helpful additional feature for medical instructional videos when correctly merged with other inputs.

5.3 MedVidQA 2022 Shared Task results

The results produced by our models, along with max, median and mean values from all participants are presented in Table 4. The performance obtained by the proposed approaches is below the reported mean. However, by comparing the obtained effectiveness presented in Table 3 and Table 4 one can observe that the models have a robust behavior across the different datasets as they yield similar performance. Peak detection-based approaches yield the highest results among our submissions, confirming the results of our experiments conducted on the MedVidQA dataset.

6 Conclusion

This work investigates two different approaches for detecting answer timestamps from medical instructional videos in the context of the MedVidQA 2022 MVAL Shared Task (Task 2). Our approaches rely only on the text extracted from the videos, either as transcripts or as the text displayed in the video's frames. After extracting the text corresponding to every video segment, we estimate its similarity to the question using four different models. We employ two different strategies to map the question-text similarity to the answer timestamp, i.e. multi-output regression model based on random forest and a peak detection model.

Our best performing peak detection model achieves 40.52 IoU=0.3 on MedVidQA 2022 Shared Task and outperforms the VSL benchmark model on the MedVidQA test dataset. We also show a positive impact of using multiple video-to-text conversion methods on the overall quality of models. Our feature extraction methods could easily extend the set of features used by end-to-end deep learning models. Further analysis is needed to assess other ways of processing the text-question similarity importance for obtaining more accurate predictions.

Acknowledgements

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval – DoSSIER (H2020-EU.1.3.1., ID: 860721).

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.

Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2022. A Dataset for Medical Instructional Video Classification and Question Answering. *arXiv preprint arXiv:2201.12888*.

Deepak Gupta and Dina Demner-Fushman. 2022. Overview of the MedVidQA 2022 Shared Task on Medical Video Question Answering. In *Proceedings of the 21st SIGBioMed Workshop on Biomedical Language Processing, ACL-BioNLP 2022*. Association for Computational Linguistics.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2019. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*.

Zhihong Lin, Donghao Zhang, Qingyi Tao, Danli Shi, Gholamreza Haffari, Qi Wu, Mingguang He, and Zongyuan Ge. 2021. Medical visual question answering: A survey. *CoRR*, abs/2111.10056.

Feifan Liu, Yalei Peng, and Max P. Rosen. 2019a. An effective deep transfer learning and information fusion framework for medical visual question answering. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9-12, 2019, Proceedings*, volume 11696 of *Lecture Notes in Computer Science*, pages 238–247. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2017. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3165–3173.

Dongxiang Zhang, Rui Cao, and Sai Wu. 2019. Information fusion in visual question answering: A survey. *Inf. Fusion*, 52:268–280.

Yangyang Zhou, Xin Kang, and Fuji Ren. 2018. Employing inception-resnet-v2 and bi-lstm for medical domain visual question answering. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018*, volume 2125 of *CEUR Workshop Proceedings*. CEUR-WS.org.