# Automatic Questionnaire and Interactive Session Generation from Videos

V. C. Skanda, Rachana Jayaram, Viraj C. Bukitagar, N. S. Kumar

# Automatic Questionnaire And Interactive Session Generation From Videos

Skanda VC[1,2][https://orcid.org/0000-0002-3849-9981], Rachana Jayaram[1,3][https://orcid.org/0000-0001-8357-8331], Viraj C Bukitagar[1,4][https://orcid.org/0000-0001-8904-7013], and N S Kumar[1,5][https://orcid.org/0000-0001-8831-0625]

[1]Dept. of CSE, PES University, Bengaluru, India
{[2]skandavc18,[3]rachana.jayaram,[4]viraj.bukitagar}@gmail.com
[5]nskumar@pes.edu

**Abstract.** In this paper, we present a tool that interleaves lengthy lecture videos with questionnaires at optimal moments. This is done to keep students' attention by making the video interactive. The student will be presented with MCQ type questions based on the topic covered so far in the video, at regular intervals. The questions are generated based on the transcript of the video lecture using machine learning and natural language processing techniques. In order to have continuity and proper flow of teaching, a LDA-based (Latent Dirichlet Allocation) model has been proposed to insert those generated questions at appropriate points called logical points.

**Keywords:** Interactive videos, Question generation, Speech to text, Logical points, Latent Dirichlet Allocation, Natural language processing, Machine learning.

## 1 Introduction

According to a study by Microsoft, the average human attention span is decreasing [1]. Decreasing attention span can have a huge impact on the learning outcomes of children. Educational/lecture videos are generally monotonous with little to no interaction. Thus, in order to keep students attentive, the videos must be made interactive. So we present a method in order to generate and insert questions automatically at appropriate points.

## 2 Literature Survey

Interactive teaching is known to be very effective. In Richard Hake's landmark 1998 study on the effectiveness of lecture-based instruction, he showed that interactive classes outperform traditional classes when it comes to learning effectiveness and concentration retention [2].

In order to generate and insert questions, transcription has to be done. In 2014 Coates et al. introduced a state of the art speech recognition system with a 84% accuracy using end-to-end deep learning [3].

In 2011, Crossno et al. compared topic modelers and found that LDA performed better than LSA especially for smaller document sizes [4]. Our approach uses LDA for topic modelling as a part logical point detection.

In his 2010 study on automated question generation, Heilman [5] delves into the intricacies of generation of factual questions from text. Our approach to cloze question generation relies heavily on machine learning techniques as well natural language processing.

In 2010, Altabe and Maritxalar presented a corpus-based approach to domain-based distractor generation [6] which is quite similar to the approach to MCQ option generation presented in this paper.

## 3 Current Work

The best way to make the videos interactive is to insert questions based on the video topic at appropriate "logical points". Logical points are those time points in the video which mark the beginning or the end of a topic or a sub topic in the video. Here a topic is defined as a collection of related paragraphs.

If logical points are too sparsely distributed, there will be very few questions in the video. To maintain a balance in the length of time between questions, the logical points are found such that they are evenly distributed throughout the video. But this interval between questions can also be set manually if needed.

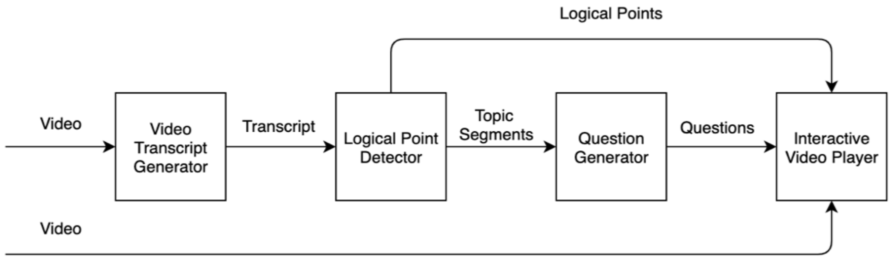

**Fig. 1.** System architecture

The questions are based on the topics covered so far in the video. They are generated from the video transcript using natural language processing and machine learning techniques. We have proposed a solution wherein the questions and the logical points at which a question must be inserted are extracted from the transcript itself.

The overall architecture of the solution has been shown in Fig. 1. The tool consists of 4 main parts which work in sequence:

1. Transcript Generator
2. Logical Point Detector
3. Question Generator
4. Interactive Video Player

### 3.1 Transcript Generation

The audio is first extracted from video. The retrieved audio file is split into multiple parts based on silence. Here silence implies that the audio level has fallen below a certain threshold. The threshold is a tunable parameter whose default value is set to 16 dbFS. Splitting the audio file is necessary as transcribing a huge audio file leads to bad transcription. Instead of splitting the audio into equal intervals, the splits are based on silence. A "silent" point is a good indication of a logical point. After splitting, the audio files are transcribed using existing transcription tools.

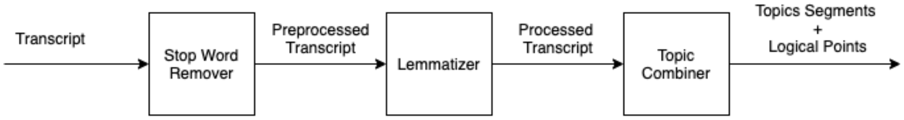

**Fig. 2.** Logical point detection architecture

### 3.2 Logical Point Detection

Text preprocessing is done on the generated transcripts before detecting logical points. The first step is to remove stop words. Stop words are articles (a, an, the), verbs (like is, was, were, etc.), pronouns (like he, she, it, they, etc.). Then lemmatization is done in order to remove different forms of the same word.

Next, in order to organize the transcripts (which were earlier generated from the audio) and detect logical points, we first find the topics for each transcript document.

LDA (Latent Dirichlet Allocation) topic modeler was used in order to extract the topics from the transcript documents. LDA is a three-level hierarchical Bayesian model, in which each transcript document is viewed as a mixture of topics. The LDA algorithm maps the topics with the documents such that words in the documents are mostly captured by those topics [7]. It returns a list of topics and their relative importance in the given document.

The extracted topics are used for deciding whether two consecutive paragraphs can be combined together or not. In order to check whether a given paragraph can be combined with its preceding paragraph, common topics among both are searched for. If there are no topics in common, then the paragraphs are not combined. This signals a logical point between the paragraphs.

If there are common topics, then we check the extent of similarity by accumulating the difference in the relative weights assigned to topics in both the paragraphs. If the accumulated difference is greater than zero, then the two paragraphs are taken to be belonging to same topic and combined together. Otherwise it is considered as a different topic and is taken to be different paragraph. Thus, it is pushed to the stack along with corresponding time stamp as a logical point. This is summarized by the Topic combiner algorithm described in Algorithm 1.

**Algorithm 1** Topic Combiner

```
function TOPIC-COMBINER(D, T)
// D: list of all pre-processed transcript documents generated
// T: list of end timestamps of transcripts
D = LDA(D)    // performs LDA on each transcript documents
doc_stack = [ ]
logic_point_stack = [ ]
doc_stack.push(D[0])
logic_point_stack.push(T[0])
index = 0
for all document in D do
    topic_similarity = 0
    for all topic in document.topics do
        if topic in doc_stack.top.topics then
            topic_similarity += (doc_stack.top.topics[topic].weight - document.topics[topic].weight)
        end if
    end for
    if topic_similarity > 0 then
        doc_stack.top.concat(document)    // concat document to top of stack
    else
        doc_stack.push(document)    // push current document to stack
        logic_point_stack.push(T[index])    // push current timestamp to stack
    end if
    index += 1
end for
return logic_point_stack, doc_stack
end function
```

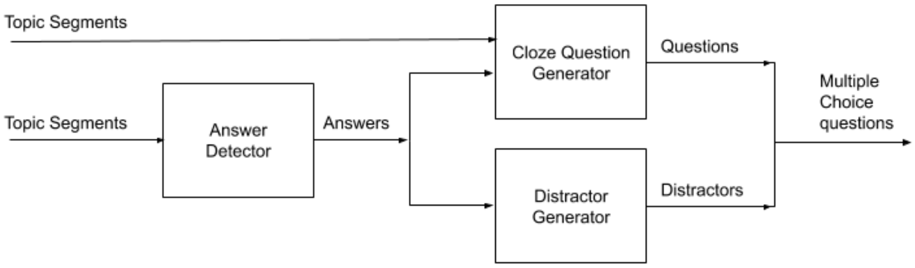

**Fig. 3.** Question generation architecture

### 3.3 Question Generation

The question generation consists of three steps:

1. Answer Detection - Given a block of text we first find all the tokens of the document that could potentially be an answer. This is similar to keyword detection. The classification of a token as an answer or not is done using a Naïve Bayes classifier trained on the Stanford Question Answering Dataset [8]. The attributes considered are part-of-speech of the token, whether the token is a named entity, tf-idf of the token, dependency of the token in its abstract syntax tree and shape of the token. Here dependency attribute of a token gives its syntactic dependency on the head token. The shape attribute gives information about capitalization, punctuation, digits in the token. It is in essence a transform on the token's string done in order to learn more about its orthographic features. The transform involves the following mappings –
   a. Lower case alphabetic characters (a-z) are mapped to 'x'.
   b. Upper case alphabetic characters (A-Z) are mapped to 'X'.
   c. Numeric characters are mapped to 'd'.
   d. Post mapping sequences of 5 or more of the same replacement characters are truncated to length 4. For instance, 'Xxxxxxx' becomes 'Xxxxx'.

   As an example, consider the sentence "Clifford is a big red dog.". The attributes of the token "dog" are listed as follows:
   a. Part of speech – Noun
   b. Named entity – False
   c. Dependency – Attribute
   d. Shape – xxx

   Consider the attributes of the token "Clifford" from the same sentence:
   a. Part of speech – Proper Noun
   b. Named entity – True
   c. Dependency – Nominal subject
   d. Shape – Xxxxx

2. Question generation - Given a sentence that contains a word categorized as an “answer”, we generate a fill-in-the-blank type question by replacing the occurrences of the answer with a blank. For example, if “dog” was categorized as an answer, a sentence in the input text containing “dog” would be transformed as follows: “Clifford is a big red dog.” becomes “Clifford is big red ____ .”.

3. Distractor generation - To build an MCQ type question, we have to generate options or distractors. Given an answer to a question, 3 words most similar to it in a relevant vocabulary to use as distractors. This is implemented using word vectors. For example, words similar to “dog” are “cat”, “wolf” and “fox”. These three distractors would be presented as options along with the correct answers.

Thus, for the sentence “Clifford is a big red dog.”, the question generated is:

Clifford is big red ____.
a. dog b. cat c. wolf d. fox

Question validation – A cosine similarity check is done between the answer of the question and the topic of the given text (provided in the LDA stage). Questions with the highest similarities are presented to the student and the rest are discarded.

## 4 Results

With this paper, we could achieve:
a. Logical segmentation of lecture videos into topics using LDA.
b. Automated generation of questions from the transcripts.
c. Generation of distractors to form MCQs.
d. Insertion of questionnaires in the lecture at logical points.

In order to know the efficacy of the overall methods proposed, we tested them on two videos. The first test was on a C++ video lecture from NPTEL [9]. 20 students were made to watch the video in a interactive video player with the UI functionality to pause the lecture and display the questionnaires at logical points. The students’ responses were collected. The video lecture was 18-minutes long. 4 logical points were detected. 6 questions were generated by question generator. Some of the questions are as follows:

1. We use ’printf’ from the _____ library and print the hello world on to the terminal or which is formally set to with the stdout file.

   a. stdin b. stderr c. stdio d. stdout

2. C strings are actually a collection of _____ in string.h

   a. Functions b. Objects c. Class d. Constructor

3. ’212’ in _____ will be considered a const int

   a. C98 b. C99 c. C97 d. C96

The statistics of student performance for the first lecture are in Table 1.

**Table 1.** Student performance statistics for the first video.

| Question number | Correctly answered | Incorrectly answered |
|---|---|---|
| 1 | 12 | 8 |
| 2 | 8 | 12 |
| 3 | 9 | 11 |

The second video was on an introduction to literary history from NPTEL [10]. It was a 22- minute video. Totally 4 logical points were detected and 10 questions were generated. Some of the questions which were generated for the video are as follows:

1. The hundred years war and the wars of the _____ accordingly had defined the fortunes of the nation.

   a. roses b. tulips c. orchids d. lilies

2. The Elizabethan period spans over _____ years from the ascension of queen Elizabeth from 1558 to the death of king James 1 1625.

   a. 67 b. 68 c. 69 d. 66

3. In many different ways England becomes a leader from the time of the reign of queen _____ I.

   a. Elizabeth b. Mary c. Anne d. Margaret

4. In many different ways England becomes a leader from the time of the reign of queen _____ I.

   a. Utopia b. utopian c. dystopia d. collectives

The statistics of student performance for the second lecture are in Table 2.

**Table 2.** Student performance statistics for the second video.

| Question number | Correctly answered | Incorrectly answered |
|---|---|---|
| 1 | 16 | 4 |
| 2 | 5 | 15 |
| 3 | 11 | 9 |
| 4 | 12 | 8 |

## 5 Future Work

Currently, the appropriateness of the location of logical points are validated manually. In order to automate the process, a machine learning model can be trained on a dataset created by manual tagging.

Logical points are currently detected by using silence and topic clustering. The appropriateness of logical points can be improved by taking into consideration the reason for silence as well. This can be achieved by using Video Analytics.

In order to improve distractor quality, vocabularies relevant to the video topics can be compiled by training GloVe models on data accumulated from online articles specifically related to the video subject [11].

The system currently in place for question validation involves performing a cosine similarity test. A more robust system can be developed by training a machine learning model to validate the questions generated. This would require building a new dataset of manually formulated questions from transcripts and training a machine learning model on the same.

## References

1. Attention spans, consumer Insights, Microsoft. http://dl.motamem.org/microsoft-attention-spans-research-report.pdf, last accessed 2019/11/21.
2. Hake, R.R.: Interactive-engagement versus traditional methods: A six thousand-student survey of mechanics test data for introductory physics courses. In: American Journal of Physics 66, 64 (1998).
3. A. Hannun, C. Case, J. Casper, et al.: Deepspeech: Scaling up end-to-end speech recognition. In: ArXiv e-prints, 2014.
4. Blei, DM., Ng, AY., Michael, JI., Lafferty, J.: Latent Dirichlet Allocation. In: Journal of Machine Learning Research. 3, pp. 993–1022 (2003).
5. Heilman, M.: Automatic factual question generation from text. In: Carnegie Mellon University. 195 (2011).
6. Aldabe, I., Maritxalar, M.: Automatic Distractor Generation for Domain Specific Texts. In: Loftsson H., Rögnvaldsson E., Hel-gadóttir S. (eds) Advances in Natural Language Processing. NLP 2010. Lecture Notes in Computer Science, vol 6233. Springer, Berlin, Heidelberg (2010).
7. Crossno, P.J., Wilson, A.T., Shead, T.M., Dunlavy, D.M.: TopicView: Visually Comparing Topic Models of Text Collections. In: 23rd International Conference on Tools with Artificial Intelligence, pp. 936-943. IEEE, Boca Raton, FL, (2011).
8. SQuAD: 100,000+ questions for machine comprehension of text, https://rajpurkar.github.io/SQuAD-explorer, last accessed 2019/11/21.

9. Module 1: Recap of C (Lecture 01), NPTEL, https://nptel.ac.in/courses/106105151, last accessed 2019/11/21.
10. Introduction to literary history (Week 1), NPTEL, https://nptel.ac.in/courses/109/106/109106124, last accessed 2019/11/21.
11. Pennington, J., Socher, R., Manning, R.D.: GloVe: Global Vectors for Word Representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014).