Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

# Augmenting Online Video Lectures with Topically Relevant Assessment Items

**Subhayan ROY[a*], Plaban Kumar BHOWMICK[b]**
[a]*Centre for Educational Technology, Indian Institute of Technology Kharagpur, India*
[b]*Centre for Educational Technology, Indian Institute of Technology Kharagpur, India*
*subhayanroy5@gmail.com

**Abstract:** In this paper, we present a prototype system for augmenting online video lectures with assessment items generated by analyzing the corresponding text transcripts. A video lecture of longer duration typically covers a number of topics. With linear discourse segmentation approach, we segment a video lecture transcript into topical segments. Inter and intra sentential structures of individual segments are analyzed to generate different types of questions. In this work, the question categories are restricted to factual questions (realized through Multiple Choice Questions-MCQs) and non-factual questions (why, how etc.) that demand higher level cognitive efforts in learner's part. We have presented evaluation of important modules involved in design of the proposed system. The experimental study has been performed with dataset of 192 video lectures (each having 1-hour duration approximately) covering 5 computer science courses from National Programme on Technology Enhanced Learning (NPTEL) project.

**Keywords:** MOOCs, topical segmentation of video lecture, automatic question generation, discourse-based question generation, MCQ distractor selection

## 1. Introduction

Massive Online Open Courses (MOOCs) have proliferated in a rapid rate into today's educational system pervading geographic, temporal boundaries. Unbounded participation spawns the difficulty in assessing the performance of huge learner population in a course. Thus, most of the courses restrict the assessment items to either Multiple Choice or range type questions. Some of the courses have experimented with peer grading of subjective answers (Suen, 2014). However, the presentation schedule of assessment items is controlled by the course instructors. This assessment model may be perfect in a physical classroom scenario as the learners are able to interact with the instructors in real time. On the other hand, due to asynchronous nature of delivery, learners in a virtual classroom do not have access to the instructors while s/he is viewing a lecture. This hinders self-paced learning that is one of the objectives of online learning.

One remediation to this problem is to insert questions in all possible parts of a video lecture, guided by the concepts of topics that are discussed in different parts of the lecture. This approach demands a large number of questions to be generated, looking at all possible scenarios. Generating such a huge number of assessment items manually may be tedious for an instructor. With the availability of syntactic and discourse parsers in Natural Language Processing domain, different researchers have started exploring the task of generating questions automatically from natural language text (Chali & Hasan, 2015) (Mazidi & Nielsen, 2014) (Afzal, 2014). This technology, though at its nascent stage, forms the foundation of our work.

The objective of our work is to augment the learning experience of a learner over the video lectures by:

- automatically generating assessment items relevant to a given video lecture, and
- inserting the generated questions in appropriate place of the video

Though our proposed system is founded on question generation technology, there are several key contributions of our work.

- *Deciding assessment timings:* Typically, questions are asked after completion of the text module, for example, completion of a topic. A topical segmentation-based approach has been adopted in our work to identify appropriate places for inserting the questions.
- *Handling the noisy transcript:* Most of the question generation systems in literature have considered well-formed text (e.g., book paragraph, web article etc.) as source. However, manually generated video transcripts are conversational in nature and in another extreme, the transcript is not available at all. Automatic Speech Recognizers (ARSs) may be employed to generate transcriptions. As a result, the transcripts either are conversational or are noisy. This poses a significant challenge to generate question from noisy text.
- *Generation of the questions:* Some previous work covers factual question generation with information available at intra-sentence level. In this work, we intend to generate questions that demand higher level cognitive skills on the learners' part. Generation of such questions may use inter-sentence relations guided by discourse theories.
- *Distractor selection for MCQs:* In this work, external resources like Wikipedia have been used to extract distractors for generated MCQs.
- *Choosing appropriate questions:* A question ranking scheme has been devised to present good quality questions to the learners.

With the above-mentioned contributions, the proposed work integrates different resources to develop an end-to-end question generation system to augment learners' experience in online video lectures.

## 2. System Overview

In this section, we present an overview of the modules involved in implementation of the proposed system (see Figure 1).

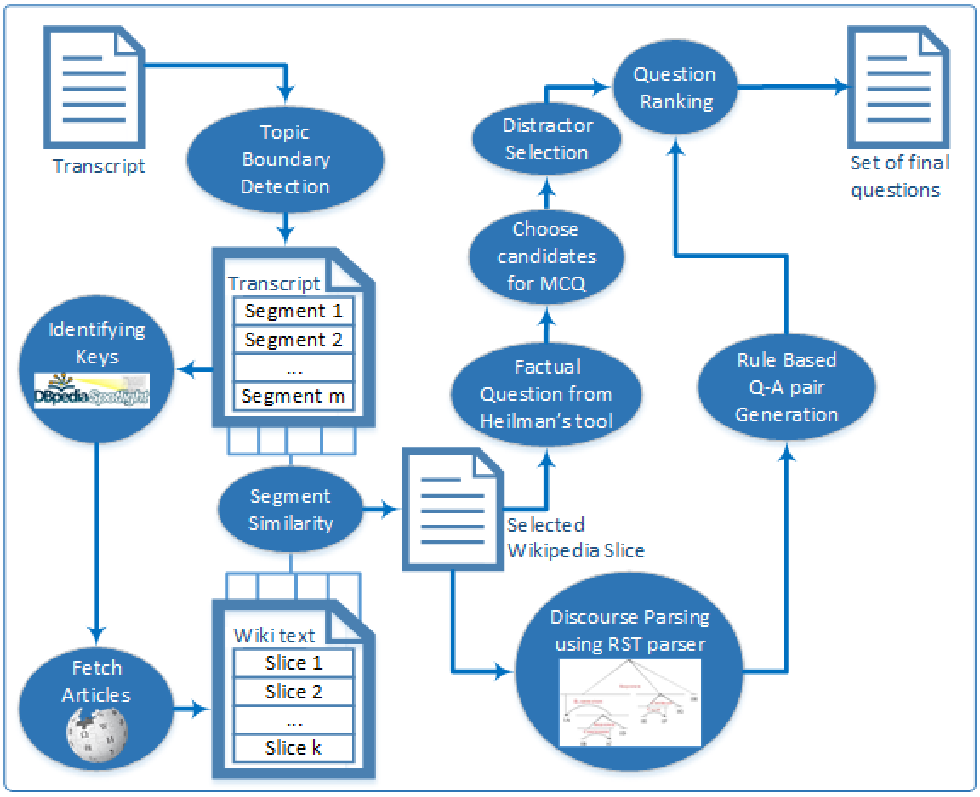
Transcript
Topic Boundary Detection
Transcript
Segment 1
Segment 2
...
Segment m
Identifying Keys
DBpedia Spotlight
Segment Similarity
Wiki text
Slice 1
Slice 2
...
Slice k
Fetch Articles
Selected Wikipedia Slice
Discourse Parsing using RST parser
Factual Question from Heilman's tool
Choose candidates for MCQ
Distractor Selection
Question Ranking
Rule Based Q-A pair Generation
Set of final questions

*Figure 1. Overview of the proposed system*

A video lecture can be seen as a sequence of segments, in each of which an instructor covers a specific topic. End of each topic is a potential place for inserting assessment items, as a self-motivated learner may want to test his/her understanding on the topic that s/he has covered recently. Thus, the first module in the architecture takes a transcript of a video lecture as input and passes it to the topic boundary detection module. The architectural overview of the system is shown in Figure 1. Different modules and processing flow of the system are as follows:

- *Topical n:* Even if a typical video lecture covers a particular topic, it can be divided into several sub-topics. This module looks at the topical distribution of input video lecture transcript and marks potential topical boundaries.
- *Retrieving Well-formed Similar Text:* To deal with the noisy nature of the transcripts, the system finds well-formed text, that is semantically similar to the input transcript segment. Wikipedia has been used as the source for well-formed text. However, it may be replaced by any other resources. This module outputs a text slice from Wikipedia that is semantically similar to the transcript segment.
- *Non-factual question generation:* At this juncture, the processing flow forks into two paths, namely, generation of factual MCQs and generation of non-factual questions. The generation of non-factual questions involves inter sentence relations. A rule-based approach has been adopted towards generating this kind of questions.
- *Choosing candidates for MCQs:* CMU question generation tool[1] has been used to generate factual questions from the text. Among those questions, we choose some of the questions as MCQ, for which the distractor can be generated.
- *Generating distractors for MCQs:* The distractor or wrong alternatives for each of the MCQs are generated using external knowledge base or ontology.
- *Question ranking:* Due to different uncertain measures or heuristics taken by the preceding sub-systems, there may be a possibility of having syntactically or semantically erroneous questions as output. The question-ranking model helps to filter out the malformed or irrelevant questions.

## 3. Topical Boundary Detection

Identification of topical boundaries is crucial for posing questions at appropriate temporal co-ordinates of the video lecture. In the present study, such boundaries are detected using TopicTiling algorithm (Riedl & Biemann, 2012) which uses topic models to determine topical shift. For effective topical distinction, topic models are trained on Wikipedia articles sampled from subject domains of the targeted lecture. According to the algorithm, the transcript is split into minimum text units, i.e. sentences. A coherence score (�) at each sentence boundary (�) is computed by comparing the distribution of topics in two adjacent blocks separated by �. These coherence scores are computed by defining a window around each of the sentence boundaries. A window consists of a left block, sentence boundary and a right block. Each block is represented as a *T*-dimensional vector (assuming LDA model consists of � topics), where the �th dimension represents topic the frequency of topic � in the block. Coherence score ($\text{�}_p$) is measured using cosine similarity of vectors of two adjacent blocks. High similarity score indicates stronger coherence between two adjacent blocks. The similarity scores are plotted and depth scores ($\text{�}_p$) are computed at minima points of the plot (refer to TextTiling algorithm (Hearst, 1997)). The points having depth score beyond a threshold are considered to be segment boundaries. The output of this module will be a set of transcript segments: $TS=\{T_1, T_2, T_3,\ldots, T_m\}$.

## 4. Retrieving Well-formed Similar Text

As discussed earlier, video lectures may contain noise in terms of grammatical error and/or homonyms (in case of ASR generated transcript). Directly generating questions from this text may produce erroneous and irrelevant questions. Hence, a preliminary step is required to deal with noisy transcript. This can be achieved using one of the following approaches:

- *Remove the noise from the transcript:* The conversational texts can be removed from the transcript by developing a classifier that classifies each sentence either into conversational and non-conversational sentence.

---

[1] CMU Question Generation Tool. http://www.cs.cmu.edu/~ark/mheilman/questions/

- *Replace the transcript with a semantically similar document:* Irrespective of the type of the noise, replace the text with a semantically similar document taken from source where the text is well-formed.

There are several issues with the first approach.

- Removing the conversational sentences from transcript segment may break the coherence of the text. This may interfere with discourse-based question generation module as abrupt or rough transition of discourse relations will be observed.
- An informative sentence may contain some conversational cues. This may confuse the classifier and consequently informative text may be filtered out.

Apprehending the above-mentioned issues with the first approach, we have adopted the second strategy that aims at finding similar and well-formed text slices from other sources.

We replaced each of the segments in �� with a semantically similar Wikipedia section. In order to find the proper replacement for the targeted transcript segment $��_� \in ��$, the concepts are first identified using DBpedia's Spotlight service (Mendes, Jakob, García-Silva, & Bizer, 2011).

Let $� = \{�_1, �_2, �_3, \ldots, �_n\}$ is the list of concepts in $��_�$. For each $�_b \in �$, Wikipedia article that is linked to $�_b$ is fetched and divided into slices according to the Wikipedia article's section and sub-section headings. Some of the sections are not taken as candidates for replacement, like, *External_links, Further_reading, References, See_also, Notes, Footnotes, History* etc. Let the candidate slices of Wikipedia article $�_b$ are: $W_b = \{W_{b1}, W_{b2}, W_{b3}, \ldots, W_{bk}\}$. For the targeted transcript segment $��_�$, a set of Wikipedia slices as $W_� = \bigcup_{k=1}^{n} W_k$ are collected. In order to select the most semantically similar slice from $W_�$, we have taken distributional semantics based approach. Latent Semantic Analysis (LSA) model is used to collect distributional information and to get semantic similarity in terms of vector/cosine similarity of the given text. Similarity scores between �� and $W_� \in W_�$ are calculated using the Cosine similarity as follows:

$$\text{��������}(\vec{��}, \vec{W_�}) = \text{���}(\theta) = \frac{\vec{��} \cdot \vec{W_�}}{\|\vec{��}\| \|\vec{W_�}\|} \ldots\ldots\ldots\ldots (1)$$

$$W_� = a\text{�������������}_{W_�}(\vec{��}, \vec{W_�}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The Wikipedia slice with the maximum similarity score ($W_�$) is selected as the replacement of the targeted transcript segment ��.

## 5. Question Generation and Ranking

This module takes a Wikipedia slice (found similar to a transcript segment) and generates MCQ-based factual questions and non-factual questions.

### *5.1 Text Preprocessing*

It has been observed that anaphoric expressions present in the retrieved Wikipedia slice pose difficulty in generating meaningful questions. For example:

| | |
|---|---|
| *Original text* | This has the advantage that incorrect candidate system designs can be revised before a major investment has been made in actually implementing the design. |
| *Generated question* | What are the advantages of this? |

Where, the word "*this*" referred to "*formal verification technique*", after anaphora resolution the question will become:

| | |
|---|---|
| *Revised Question* | What are the advantages of formal verification technique? |

In order to deal with this issue, a pre-processing step that performs pronoun resolution (Reconcile[2] has been used) is applied over the text slice.

### *5.2 Generation of Non-Factual Questions*

While factoid questions are good at testing learners' knowledge level skills (lower order cognitive skill), non-factual questions demand higher order cognitive skills such as inference, synthesis,

---

[2] Reconcile – Coreference Resolution Engine. https://www.cs.utah.edu/nlp/reconcile/

application etc. Typically, the answers to those questions are formed by connecting a set of knowledge items that are dispersed in different sentences

Discourse theory provides a framework through which the sentences in natural language can be stitched in a coherent manner. This motivates us to analyze the discourse relations present in an input slice in order to generate questions for this category. Discourse relations are extracted using Rhetorical Structure Theory (RST) (Mann & Thompson, 1988) style discourse parser[3] (Feng & Hirst, 2012). According to the RST framework, a discourse tree can be formed for a given coherent text. The leaves of a discourse tree are minimal text units (called text spans) that are non-overlapping. These minimal text units are called elementary discourse units (EDUs). The adjacent nodes in the tree are related by discourse relationships of two types: mononuclear and multi-nuclear. In case of mononuclear relations, the central text-span is named as the nucleus (denoted by [N]), and the other span is called a satellite (denoted by [S]). Whereas in multi-nuclear relationship both the text spans are equally central. A discourse based approach was proposed for generating "why" questions from text (Prasad & Joshi, 2008). In their work, they found that 71% of independently developed data set of "why" questions can be correlated with causal relations.

Another work shows that the discourse connectives are also important in order to generate questions (Agarwal, Shah, & Mannem, 2011). In their work, firstly, the relevant part of the text was identified, followed by sense disambiguation, identification of question type and application of syntactic transformations on the content.

We have adopted a rule based strategy to generate non-factual questions. The rule-base contains a set of rules, each of which is characterized by a discourse relation and a discourse cue. The rules are defined using the following template:

Relation: <*name_of_the_discourse_relationship*>
Connective: <*discourse_connective*>
Precondition: <*relationship*>*[text_span][text_span]*
Post-condition: *Question-Answer (QA) pair generation rule*

The rules are described using the following notations:

| | |
|---|---|
| [aux-verb]: | Auxiliary verb |
| [X-sub]: | Subject of [X] |
| [X-verb]: | Main verb of [X] |

Where, X denotes nucleus [N] or satellite [S].

Following example shows one question generated from one rule defined over *Explanation* relation:

<table>
<tr><td><i>Source Text</i></td><td colspan="2">Companies like Oracle and Microsoft provide their own APIs so that many applications are written using their software libraries that usually have numerous APIs in them.</td></tr>
<tr><td><i>Discourse spans</i></td><td colspan="2">Companies like Oracle and Microsoft provide their own APIs <b>[N]</b> so that many applications are written using their software libraries that usually have numerous APIs in them <b>[S]</b>.</td></tr>
<tr><td><i>Relationship</i></td><td colspan="2">Explanation[N][S]</td></tr>
<tr><td><i>Connective</i></td><td colspan="2">so</td></tr>
<tr><td rowspan="2"><i>Rule for QA pair generation</i></td><td><i>Precondition</i></td><td><i>Relation</i>: explanation [N][S]<br><i>Connective</i>: so / because</td></tr>
<tr><td><i>Post condition</i></td><td>Q: Why [aux-verb] [N]?<br>A: [S]</td></tr>
<tr><td><i>Example QA pair</i></td><td colspan="2">Q: Why do Companies like Oracle and Microsoft provide their own APIs?<br>A: Many applications are written using their software libraries that usually have numerous APIs in them.</td></tr>
</table>

Some other rules are presented in Table 1.

---

[3] RST-style Discourse Parser. http://www.cs.toronto.edu/~weifeng/software.html

*Table 1: Rules for generating question answer pair.*

| *Sl.* | *Relationship* | *Discourse connectives* | *Rules for generating QA pairs* |
|---|---|---|---|
| *1* | *Explanation*<br>*[N][S]* | *in order to, so, because* | Q: Why [aux-verb] [N]?<br>A: [S] |
| *2* | *Elaboration*<br>*[N][S]* | *which, that* | Q: What is [N-sub]?<br>A: [S] |
| | | | Q: What [aux-verb] [N-sub] [N-verb]?<br>A: [S] |
| *3* | *Joint*<br>*[N1][N2]* | *and* | Q: What [aux-verb] [N1-sub] [N2-verb]?<br>A: [N2] |
| *4* | *Attribution*<br>*[S][N]* | *on, that* | Q: What [aux-verb] [S] about [N-sub]?<br>A: [N] |
| *5* | *Condition*<br>*[N][S]* | *if, there* | Q: What happens if [S]?<br>A: [N] |
| | | | Q: Where [N]?<br>A: [S] |
| *6* | *Same-unit*<br>*[N1][N2]* | *in* | Q: What [aux-verb] [N1-verb] in [N1]?<br>A: [N2] |

Examples of some more QA pairs generated using the above rules are shown in Table 2.

*Table 2: Example questions generated by the rules.*

| *Relationship* | *Connective* | *Question-Answer pair* |
|---|---|---|
| *Explanation*<br>*[N][S]* | *because* | Q: Why Security breaches on application service provider applications are a major concern?<br>A: because application service provider can involve both enterprise information and private customer data. |
| *Elaboration*<br>*[N][S]* | *that* | Q: What are the advantages of formal verification technique?<br>A: that incorrect candidate system designs can be revised before a major investment has been made in actually implementing the design. |
| *Condition*<br>*[S][N]* | *there* | Q: Where there may be no need for a pretty graphical user interface, leaving the application leaner, faster and easier to maintain?<br>A: If an application is only going to be run by the original programmer and/or a few colleagues |
| *Same-unit*<br>*[N][N]* | *in* | Q: What will happen in the absence of an experienced architect?<br>A: there is an unfortunate tendency to confuse the two architectures, the engineer thinks in terms of hardware and software and the technical solution space, whereas the user may be thinking in terms of solving a problem in a reasonable amount of time and money. |
| *Attribution*<br>*[S][N]* | *that* | Q: What does many people believe about software engineering?<br>A: software engineering implies a certain level of academic training, professional discipline, adherence to formal processes, and especially legal liability. |

### *5.3 MCQ generation and distractor selection*

The factual questions are presented as MCQs in the present system. The methodology presented by Michael Heilman generates factual questions from syntactically complex sentences (Heilman, 2011). In his work, firstly the simplified factual statements are extracted from different syntactic transformations. Then the factual question-answer pairs (QA pair) are generated. A subset of the questions generated by CMU question generator tool is selected as all are not observed to be appropriate MCQ candidates. The selection strategy uses answer associated to a question generated by CMU tool. The question is a candidate for MCQ if an associated Wikipedia page for the corresponding answer phrase can be found; otherwise, it is discarded.

To generate the distractors or the wrong alternatives of the MCQ questions, domain specific ontology based approach is already proposed (Alsubait, Parsia, & Sattler, 2015). In present implementation, we rely on the Wikipedia article-category hierarchy for this task. The categories of the correct answer (say $�_c$) have been extracted first, let, the categories for $�_c$ are: $�_w = \{�_1, �_2, \ldots, �_n\}$. Titles of the articles belonging to these categories can be possible candidates for distractors. These candidates are ranked and top ones are selected as distractors. We have used the distribution of the candidates over the answer categories as ranking function. All the article titles belonging to the categories in $�_w$ are collected and frequencies of the titles are extracted. The top three article titles are chosen as the final distractors. For example, the factual question with the correct answer is as follows:

| | |
|---|---|
| *Question* | Who introduced the key concept of modularity and information hiding in 1972 to help programmers deal with the ever increasing complexity of software systems? |
| *Answer* | David Parnas |

The Wikipedia categories for the article "*David Parnas*" are: *Canadian computer scientists*, *Formal methods people*, *Software engineering researchers* etc. Articles under those categories are as follows:

$\{$[illegible]$\ldots$[illegible]$\} \in$ [illegible]

$\{$[illegible]$\ldots$[illegible]$\} \in$ [illegible]

$\{$[illegible]$\ldots$[illegible]$\} \in$ [illegible]

The sorted list of candidates based on frequency count is: *David Harel(2), Eric Hehner(2), Hakan Erdogmus(2), Joe Stoy(1), Jit Bose(1).* So, the final distractors are (top 3): *David Harel, Eric Hehner, Hakan Erdogmus*.

### 5.4 Question Ranking

The questions generated may have issues with respect to their acceptability level. These issues range from syntactic to semantic level, as follows:

- *Syntactic validity:* The rules for generating questions have been devised by inspecting a bounded set of text segments. Thus, there may be cases that are not handled or handled erroneously by the system due to lack of coverage of the rule base. Because of this imperfection in the rule base, some of the generated questions may be grammatically incorrect.
- *Ambiguous questions:* Due to several text properties (e.g., overuse of nouns), some of the generated questions are observed to be semantically ambiguous. For example:

| | |
|---|---|
| *Original text* | As of 2004, in the U.S., about 50 universities offer software engineering degrees, **[N]** which teach both computer science and engineering principles and practices **[S]**. |
| *Question* | What is software engineering degrees? |

- *Irrelevant to the transcript:* Some questions are syntactically correct, but irrelevant to the video lecture.

To handle the above scenarios, we have assigned scores against each of the questions, based on some weighted features.

- *Grammatical correctness ($�$):* The feature measures syntactic validity of the generated questions and is quantified with the number of grammatical errors. LanguageTool[4] is used to compute this feature.

---

[4] LanguageTool. https://www.languagetool.org/

- *Vagueness (�):* This feature indicates some of the patterns present in awkward questions. It considers following attributes of a question:
    - The ratio between the number of nouns and the length of the source text.
    - Presence of subordinate clause in the source text.
    - Number of verb phrases in the question.
- *Relevance to the transcript segment (�):* It helps to identify the irrelevant questions. This feature is quantified by the LSA similarity score between the transcript segment and the generated QA pair.
- *Anaphoric expressions (�):* This feature counts the number of co-reference expressions in a question.
- *Question length (�)*: For non-MCQs, we have considered the ratio between the number of tokens/words in the source sentence(s) and the answer phrase.

Final score is calculated as follows:

$$\mathrm{score}(�) = \sum �_i . �_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Where, $�_i$ denotes the weight (estimated empirically) of the $i^{�h}$ feature.

## 6. Evaluation of Components

In this section, we provide quantitative evaluation of different components of the proposed system. Through our evaluation study, we intend to analyze the performance of the following:

- Performance of topic boundary detection
- Quality of the generated questions
- Quality of the generated distractors

### *6.1 Data Set*

We have considered five video courses from NPTEL in order to perform aforementioned evaluation studies. The names and distribution of the lectures are presented in Table 3.

*Table 3: List of courses taken for case study.*

| *Sl.* | *Course Name* | *No of video lectures* |
|---|---|---:|
| *1* | Software Engineering | 39 |
| *2* | Introduction to Computer Graphics | 35 |
| *3* | Internet Technology | 40 |
| *4* | Computer Architecture | 38 |
| *5* | Computer Networks | 40 |

### *6.2 Performance of Topic Boundary Detection*

The performance of topic boundary detection module is evaluated using WinPR (Scaiano & Inkpen, 2012) measure. Some of the video lectures in NPTEL portal are marked with topic boundaries. We use those boundaries-marked data as reference boundaries. The system generated topic boundaries are compared against the reference data provided by NPTEL.

We calculated the WindowDiff, WinP, WinR values (Scaiano & Inkpen, 2012) for each of the video lectures taken from five courses. All these measures require a window size to be specified in their computation. The strategies followed for selecting window size are as follows:

*Strategy 1:* $�1 = \min(�������\_�����h)$

*Strategy 2:* $�2 = \dfrac{\max(�������\_�����h) - \min(�������\_�����h)}{2}$

*Strategy 3:* $�3 = a���a��(�������\_�����h)$

Where, $�������\_�����h$ = number of sentences in the segment. Following table (see Table 4) shows the average WindowDiff, WinP, WinR for each of the courses.

*Table 4: WindowDiff, WinP, WinR values for individual courses.*

| | **WindowDiff** | | | **WinP** | | | **WinR** | | |
|---|---|---|---|---|---|---|---|---|---|
| | ***k1*** | ***k2*** | ***k3*** | ***k1*** | ***k2*** | ***k3*** | ***k1*** | ***k2*** | ***k3*** |
| Course 1 (36 lecs) | 0.31 | 0.29 | 0.10 | 0.29 | 0.71 | 0.49 | 0.10 | 0.26 | 0.24 |
| Course 2 (37 lecs) | 0.28 | 0.72 | 0.68 | 0.28 | 0.65 | 0.59 | 0.13 | 0.32 | 0.29 |
| Course 3 (34 lecs) | 0.30 | 0.66 | 0.63 | 0.35 | 0.76 | 0.73 | 0.16 | 0.31 | 0.31 |
| Course 4 (38 lecs) | 0.35 | 0.71 | 0.66 | 0.36 | 0.76 | 0.68 | 0.13 | 0.28 | 0.25 |
| Course 5 (35 lecs) | 0.34 | 0.65 | 0.61 | 0.42 | 0.75 | 0.72 | 0.18 | 0.31 | 0.30 |
| Average | 0.316 | **0.606** | 0.536 | 0.34 | **0.726** | 0.642 | 0.14 | **0.296** | 0.278 |

It is observed that *Strategy 2* outperforms the others and precision value is at acceptable level. It is also noted that recall of segment boundary detection module is comparatively low than precision. It can be inferred that the system may fail to retrieve many segment boundaries. However, most of the identified boundaries are true boundaries.

### *6.3 Quality of the generated questions*

The quality judgment of the generated questions has done by human annotators using Likert scale with the five-level Likert items: *Totally Unacceptable, Unacceptable, Neutral, Acceptable, Perfectly Acceptable*. Five human annotators have been engaged in annotating 30 questions generated by our system. The questions are uniformly distributed over different rules in rule base. In Table 5,we present, for different discourse relations, distribution of questions in different levels in Likert scale.

*Table 5: Distribution of questions in different levels of Likert scale.*

| *Discourse Relationship* | *Totally Unacceptable* | *Unacceptable* | *Neutral* | *Acceptable* | *Perfectly Acceptable* |
|---|---|---|---|---|---|
| *Explanation* | 2% | 6% | | 46% | 46% |
| *Elaboration* | 1% | 8% | 4% | 36% | 51% |
| *Joint* | 5% | 40% | | 45% | 10% |
| *Attribution* | | 20% | 15% | 40% | 25% |
| *Condition* | 2% | 2% | 40% | 46% | 10% |
| *Same-unit* | 10% | 10% | | 60% | 20% |

For the relationships same-unit, joint and attribution, we found the acceptability to be lower than that for other relations. This may be caused due to the generic connective words like, ‘and’, ‘that’, etc. associated with these relations.

### *6.4 Quality of the MCQ distractors*

The quality of the MCQ distractors was again judged by 5 human annotators. Each annotator was asked to judge 20 system generated questions with their respective correct answers and 3 system generated distractors for each question. For each distractor, the annotators were requested to mark if the distractor is a good one or not (Boolean marking). Table 6 presents the response data. The first column of the table shows four *response categories*. Each category represents the number of good distractors per question (‘X good distractors’ indicates X number of valid distractor in a single question). Other columns in this table show the distribution of 20 question in different response categories for a given annotator.

*Table 6: Distribution of questions in different response categories.*

| *Response Category* | *Annotator-1* | *Annotator-2* | *Annotator-3* | *Annotator-4* | *Annotator-5* |
|---|---|---|---|---|---|
| *3 good distractors* | 6 | 5 | 8 | 5 | 4 |
| *2 good distractors* | 2 | 3 | 3 | 6 | 6 |
| *1 good distractor* | 8 | 7 | 7 | 5 | 4 |
| *0 good distractor* | 4 | 5 | 2 | 4 | 6 |

In some cases, the categories of the Wikipedia cover a wide range of concepts, which makes them less relevant. For example, the categories for the article "*Alan Turing*" are "*1912 births*", "*Philosophers of mind*", "*English inventors*", which would result in less specific distractors.

## 7. Conclusion and Future Scope

In this work, we have taken a discourse theory motivated approach towards automatic question generation to augment online video lectures facilitating self-paced learning. A rule-based approach has been adopted to generate questions from discourse structure of a text segment. The current rule base covers most of the discourse relations. However, there are further scopes of improvement as far as the discourse connectives used in the rules are concerned.

The question generation module works on the Wikipedia article slices that are semantically similar to a topical segment from input video lecture transcript. Application of state-of-the-art semantic similarity measures like word2vec (Mikolov, Chen, Corrado, & Dean, 2013) in retrieving semantically similar text segments is in future line of work. We also intend to investigate on filtering noisy transcript data to generate questions from them directly.

## References

Afzal, N. &. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing, 18*(7).

Agarwal, M., Shah, R., & Mannem, P. (2011). Automatic question generation using discourse cues. *Innovative Use of NLP for Building Educational Applications* (pp. 1-9). Association for Computational Linguistics.

Alsubait, T., Parsia, B., & Sattler, U. (2015). Ontology-Based Multiple Choice Question Generation. *KI-Künstliche Intelligenz*, 1-6.

Chali, Y., & Hasan, S. A. (2015). Towards Topic-to-Question Generation. *Computational Linguistics, 41*(1).

Feng, V. W., & Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. *50th Annual Meeting of the Association for Computational Linguistics, 1*, 60-68.

Hearst, M. A. (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics, 23*(1), 33-64.

Heilman, M. (2011). *Automatic factual question generation from text.* Doctoral dissertation, Carnegie Mellon University.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse, 8*(3), 243-281.

Mazidi, K., & Nielsen, R. D. (2014). Linguistic Considerations in Automatic Question Generation. *ACL (2),* (pp. 321-326).

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. *7th International Conference on Semantic Systems, ACM,* (pp. 1-8).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Prasad, R., & Joshi, A. (2008). A discourse-based approach to generating why-questions from texts. *Question Generation Shared Task and Evaluation Challenge,* (pp. 1-3).

Riedl, M., & Biemann, C. (2012). TopicTiling: a text segmentation algorithm based on LDA. *ACL '12 Proceedings of ACL 2012 Student Research Workshop*, 37-42.

Scaiano, M., & Inkpen, D. (2012). Getting more from segmentation evaluation. *The north american chapter of the association for computational linguistics: Human language technologies,* (pp. 362-366).

Suen, H. K. (2014). Peer assessment for massive open online courses (MOOCs). *The International Review of Research in Open and Distributed Learning, 15*(3).