

## Developing a Collaborative MOOC Learning Environment utilizing Video Sharing with Discussion Summarization as Added-Value

Mohannad Al-Mousa<sup>1</sup> and Jinan Fiaidhi<sup>2</sup>

<sup>1</sup>Computer Science Department, Lakehead University  
Thunder Bay Ontario, Canada

<sup>2</sup>Computer Science Department, Lakehead University  
Thunder Bay Ontario, Canada

<sup>1</sup>[malmousa@lakeheadu.ca](mailto:malmousa@lakeheadu.ca), <sup>2</sup>[jfiaidhi@lakeheadu.ca](mailto:jfiaidhi@lakeheadu.ca)

### Abstract

*Massive Open Online Courses (MOOC) platforms provide a rich environment for knowledge creation through its massiveness and inherited collaborative tools. However, it also restricts spontaneous knowledge sharing by the existing LMS barriers between the main multimedia content and the collaborative tools. None the less, the collaboration still massive due to the number of participants. The separation of the multimedia content and the discussion tools is the first focus point of this paper. Moreover, this article is presenting a new added value to the MOOC architecture so to link the learner's discussions and its summary with the multimedia contents. The added-value component involves a summarization algorithm that summarizes the shared collaborative textual discussion collected from the various learners viewing relevant MOOC multimedia/video contents. The affectivity of the summarization component was tested using the popular ROUGE software package from University of Southern California. The new MOOC architecture represents an enhanced learning environment that enables learners to share the multimedia information along with its annotated collaborative information with the power of summarizing the final outcome of the presented annotations relevant to a specific shared multimedia content.*

**Keywords:** MOOC, Collaborative Learning, Multimedia Sharing Forums, Video Annotation and Summarization

### 1. Introduction

With the fast growing of the MOOC community and the increase in the number of Learning Management Systems (LMSs) available online, the amount of shared information is massive. Current LMS - in particular MOOC providers - offer advanced content delivery techniques to enhance the pedagogical process such as: interactive video, active retrieval practices, and quizzes. The main knowledge creation assets within MOOCs are encapsulated in other tools like discussion forums, blogs, and wikis. However, these tools exist as separate entities within the platform and still following traditional usage techniques. We believe these tools need to be fully integrated with the main content to encourage spontaneous collaboration. From our experience with some MOOC course, the amount of collaboration and information sharing are still overwhelming due to the massive number of participants and the limited window of collaborative tools. However, most of the shared information could be redundant or irrelevant to the main content. These information need to be processed to provide the most succinct knowledge. Therefore, we need to link these tools to the main content and summarize its information to provide a succinct knowledge that is most relevant

to the course content. This paper addresses this shortcoming by suggesting a two stage system to accomplish this task. In the first stage, we link the discussion tools to the main course content. Then, in the second stage, we apply textual summarization techniques to present a summary of all shared content. We use techniques like Term Frequency, Inverse Document Frequency (TFIDF), stemming algorithms, Vector Space Models, and Cosine Similarity to rank the sentences. We then tune the TFIDF values and boost the sentence ranks using the main content. At the last step, we choose the most relevant sentence to build our summary. Finally, we evaluate our result using Recall-Oriented Understudy for Gisting Evaluation (ROUGH) metric to compare our automated summary with other human summaries. The obtained results demonstrate that we can achieve high improvement compared to the baseline and other similar techniques.

## 2. Related Work

LMSs and MOOC providers do integrate advanced discussion tools within their platforms. While these tools can be displayed in the main content page, they are yet to be integrated to the actual course content (i.e. video, audio, text lectures). WizIQ<sup>1</sup> and other MOOC providers offer live chat linked to the live video lecture. The chat is usually for asking questions and providing additional notes or reading materials. The main knowledge is still within the live video. Outside the educational field, Sound Cloud<sup>2</sup> site offers micro-blog annotation to the audio time line; and that was our first intimation to create this tool.

For the summarization task, summarizing the text within forums, blogs, or wikis is different from fully structured document summarization. The text in these tools is usually not or semi-structured, and falls under the definition of microtext [1]. Therefore, the summarization process may require different techniques and approaches like the ones in [2], [3], [4], and [5]. We have followed sentence extraction summarization technique with different sentence ranking approach than in [4].

## 3. Implementation

Integration and summarization are the two focus tasks in this paper. The examined tools are YouTube video lecture and discussion forums as a content delivery and collaborative tool respectively. While the proposed system is a MOOC Forums-Media Context Summarization (MFMCS) system as an added value MOOC tool, created in two stages. In the first stage, we will create a video content - discussion forums integration plugin we called JavaScript Media Player Wrapper (JSMPW). The JSMPW is created using JQuery and JavaScript on the client side and its main purpose is to integrate discussion forums with video time line. In the second stage we will exploit the JSMPW features to apply an enhanced NLP summarization task on the shred discussions on each video lecture. Finally present the summary to the users through the JSMPW interface. Figure 1 illustrates the overall architecture our MFMCS architecture.

---

<sup>1</sup> <http://www.wiziq.com/>

<sup>2</sup> <https://soundcloud.com/>

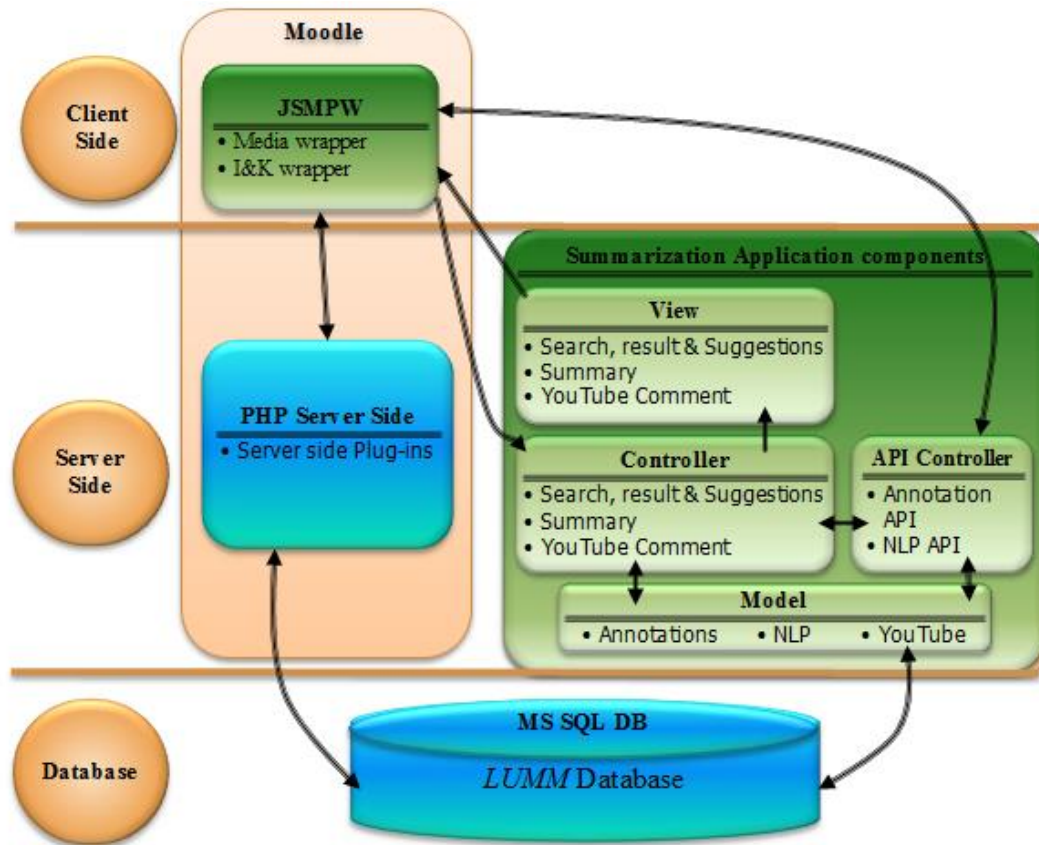


Figure 1. The Overall MOOC Architecture

### 3.1. JSMPW integration tool

The JSMPW represent the GUI interface on the MFMCS system. It is designed in a class like Object Oriented Programming (OOP) architecture using the jQuery class library. The JSMPW is built as a plugin to boost its reusability within different platforms. The JSMPW consist of two main wrappers and three backend handlers:

- **Video Wrapper:** responsible for the integration part of the MFMCS system. This wrapper encourages spontaneous collaboration in the duration of the video lecture. It does that by allowing users to create new discussion forum post as an annotation on the video time line. Therefore, this enables users to create a discussion within a timed context of the video manuscript.
- **Information and Knowledge (I&K) Wrapper:** responsible for mainly presenting the shared discussions on the current video and the summary of these discussions; one summary for all discussions for each video lecture. Additional features are also included in this wrapper like retrieval of discussions based on a query search, retrieval of the latest 25 YouTube Comments posted on the YouTube page for the current video.

The JSMPW facilitates the connection between the video content and the discussion forums within the MOOC platform. For example, while users are watching the video lecture they can initiate a new discussion forum directly from the video wrapper's annotation menu. The JSMPW pauses the playing video lecture and opens an embedded discussion forums page within a dialog modal on top of the video. Once the user fills the discussion forum and saves the changes, the JSMPW closes the dialog modal, creates an annotation at the time of the

pause, and display an annotation icon in an annotation bar within the wrapper below the video.

In addition to the above two wrappers the JSMPW tool contains the following three backend handlers:

- **YouTube handler**<sup>3</sup>: responsible for initializing YouTube player and obtaining the YouTube video metadata (title, description, and closed caption).
- **Annotation API handler**: responsible for the communication between the JSMPW interface and the Summarization Application server side Annotation API. The annotation API handler requests the annotation information from the API, sends a new annotation requests to the API, and loads the annotation icons in the annotation bar.
- **Media dialog handler**: is responsible for displaying the embedded discussion forum page for creating a new discussion, viewing, and replying to existing posts. The media dialog uses the jQuery dialog modal class library to create the popup modal box on top of the content page.

Figure 2 displays the main interface components mentioned above of the JSMPW.

### 3.2. Summarization Application (Web App and APIs)

The Summarization Application is a server side component of the MFMCS system. It contains two main sub applications:

- **Web application**: the web application contains the web pages that are rendered to the JSMPW interface to be displayed to the users. The web application is designed using the MVC design pattern and it contains five main pages; search form, search results, search suggestions, YouTube comments, and discussions summary. The search form page allows users to search all posts linked to the current video lecture. While the results page is the response of the search query requested from the search form page. The search suggestion page provides the users with search suggestions, from the existing posts, while they are typing the search query. The YouTube comments page is responsible for calling an external YouTube API to retrieve the latest 25 YouTube comments from the YouTube site. The discussion Summary page is responsible for calling the NLP API to retrieve a summary of all discussions on the current video.
- **APIs**: we have built two APIs; the Annotation API, and the NLP API. The Annotation API's main responsibility is to response to all enquiries related to the annotations and discussion posts from the following pages: search form, results, and suggestions. In addition to the direct requests from the JSMPW such as: creating a new annotation, or retrieving the annotation's list. The NLP API contains the main enhanced summarization algorithms to create an extracted summary of all discussions on the current video. The NLP API reacts to the summary page request.

---

<sup>3</sup> We use YouTube videos as the host for video lectures that can be embedded into an iFrame within our course content page.



**Figure 2. JSMPW GUI Interface**

**NLP API summarization process:**

The NLP API is responsible for the NLP summarization process that is represented in the following main steps:

**Data extraction, parsing, and stemming:** discussion posts are stored in the database in text format. Each discussion post is stored as a record in the database. We treat a full discussion as one document while discussion posts/threads as paragraphs. The API extracts the media related posts to build the document (discussion) objects. Then starts the parsing process using the parser classes we built; see the main parser classes and parameters in Figure 3. During the parsing process we use regular expressions to segment sentences and tokenize terms. Finally at the lowest class level (phrase class) we apply Porter’s stemming algorithm to obtain the stem of each phrase (term).

**Retrieve media metadata:** this step is required to contextualize the final summary based on the video lecture manuscript. As pointed out in the previous section, we used YouTube as the main host for the video lectures. Therefore we used YouTube APIs to obtain the following media metadata: video title, description, and closed caption and used them in further summarization steps below.

**Compute TFIDF:** the TFIDF is still the best known weighting model in IR [6] in particular the length normalized TFIDF modal [7]. We computed the IDF value by using Google Search engine and the web as a training dataset. If we consider web pages as documents, then based on [8], the total number of indexed webpages in Google search engine is approximately 47 billion. We then used Google web search API to retrieve the count of web pages containing a specific term. Then we compute the TF and normalize it by the length of the document. Finally compute the TFIDF using the known length normalized TF model as in equation (1):

$$w_{td} = \frac{tf_{td}}{l_d} * \log \frac{N}{1 + n_t} \tag{1}$$

Where:

- $w_{td}$ : Weight of term  $t$  in document  $d$ .
- $tf_{td}$ : Number of occurrences of a term  $t$  in document  $d$ .
- $N$ : Total number of Google's indexed web pages.
- $n_t$ : Number of web pages containing term  $t$ .
- $l_d$ : the number of unique vocabularies in the document.

**TFIDF penalization:** one of the main problems faced when summarizing microtext multi-threaded discussions is topic drift [9]. Therefore we exploited the JSMPW feature to obtain the term frequency of the video lecture's manuscript, from the closed caption. We compute the TF for each term in the closed caption. Then normalize it by the maximum TF in the closed caption. Finally we scale the computed TFIDF in the previous step based on the normalized closed caption TF. The final equation for the penalized weighted TFIDF is in equation (2) below:

$$pw_{t,cc,d} = \frac{tf_{tcc}}{\max(tf_{tcc})} * \frac{tf_{td}}{l_d} * \log \frac{N}{1 + n_t} \quad (2)$$

Where:

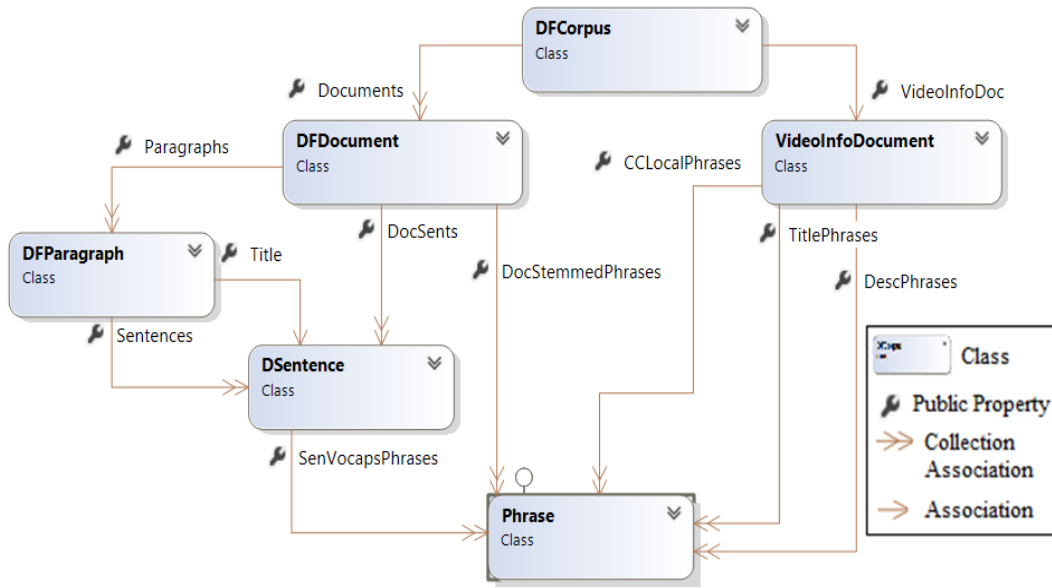
- $tf_{tcc}$ : Term frequency of term  $t$  in the closed caption text  $cc$ .
- $pw_{t,cc,d}$ : The new penalizes weight of term  $t$  in closed caption  $cc$  and document  $d$ .

**Sentence ranking:** for each document we used cosine similarity method to rank the sentence within each document by obtaining the average sentence vector in the document vector space. Each sentence is presented as a vector in the document space, where its elements are the penalized weights of each of its terms. Then we apply the cosine similarity algorithms for the normalized vector of each sentence and all other sentences in the same document. Then we compute the sum of all similarity scores for each sentence see equation (3) to obtain an initial sentence score. Eventually, these scores will be boosted and used to select the summary candidate sentences.

$$\text{score}(S_i) = \sum_{k=0}^n \vec{v}(s_i) \cdot \vec{v}(s_k) \{k \neq i, 0 < [k, i] < n\} \quad (3)$$

Where:

- $n$ : is the number of sentences
- $S_i$ : is the  $i^{\text{th}}$  sentence where  $\{i, 0 < i \leq n\}$



**Figure 3. NLP Parser Class Diagram**

**Boost sentence scores:** in this step we again exploit the JSMPW component to obtain the manuscript context using the discussion annotation time in the video time line. We use a boundary configuration value (default is 30 seconds) to select the closed caption text 30 seconds before the annotation time of the discussion and 30 seconds after; we refer to this text as local sentence closed caption (local S<sub>cc</sub>). Then we compute the TFIDF for terms in the (local S<sub>cc</sub>) and create the sentence vector in the document vector space. Finally we accumulate the cosine similarity score of the (local S<sub>cc</sub>, video title, and video description) with each sentence in the discussion. Equation (4) represents the boosted sentence score.

$$BScr(S_i) = sim(s_i, local\ s_{cc}) + sim(s_i, s_{title}) + \sum_{x=0}^{n_{desc}} sim(s_i, s_x) + \sum_{k=0}^n sim(s_i, s_k) \{k \neq i\} \quad (4)$$

Where:

- local S<sub>cc</sub>: the local closed caption around the annotation time
- S<sub>title</sub>: the media title sentence
- S<sub>x</sub>: the x<sup>th</sup> sentence in the description
- n<sub>desc</sub>: the number of sentences in the description

Figure 4 is the algorithms for the boost sentence score step above.

**Sentence exclusion and selection:** at the final step we exclude question and exclamation sentences as they shouldn't appear in a summary. Then we select the sentences above a configurable threshold value. The threshold value is a specified percentage above the average sentence scores within each document/discussion.

#### 4. Experiment setup

Our NLP API within the Summarization Application utilizes the annotations from the JSMPW and the video metadata to provide an enhanced contextualized summary. As pointed in section 3, currently there is no such a tool that links the discussion forums with the video content within MOOC and LMS platforms. Therefore, we had to create our testing dataset with a real life implementation of both systems.

**Test dataset:** we setup a Moodle LMS to serve as our MOOC platform. Then we created a one week course that contains a YouTube video as a main course deliverable content.

Then we called for registration for this course and encouraged the participants to enrich the discussion forums while watching the video lecture. At the end of the course we had 18 students registered for the course and seven of them were active in the discussion forums. We ended up with 7 main discussions, with total of 21 posts.

**Evaluation measures:** we followed the standard evaluation practice in the field of automated summarization. An automated system summary is evaluated by comparing it to an ideal summary created by humans. Wherefore, we have requested 4 writers to watch the video lecture and then summarize the discussions in the context of the video. In other words, if the video lecture is about Global Warming, yet the lecturer does not address its economic issues, then no sentences about Global Warming economic issues should be included in the summary.

Once we obtained the ideal and automated summaries we compare them using the Recall-Oriented Understudy of Gisting Evaluation (ROUGE) system [10]. Although the initial ROUGE is based on the Recall metric yet the latest version of ROUGE (v 1.5.5) includes an additional two metrics: Precision and F-Score. The following Contingence table (Table 1) explains demonstrates how these three metrics are computed in the equations (5), (6), and (7).

```

1. Boost Sentence Score() {
2.   Tokenize Title from Video Info Document object
   // use the same RegEx tokenizer
3.   Tokenize description from Video Info Document
   object
4.   Get document's annotation time //
   annotation time in the video from the JSMPW
5.   Get Local/context closed caption surrounding
   annotation time
6.   Tokenize Local closed caption
7.   For each sentence in the document {
8.     // Compute the Cosine Similarity between the
     current sentence in the loop and video
     information document sentence (title,
     description, and Local closed caption) and add
     to the sentence score calculated in the
     previous step.
9.     Sentence score += Compute cosine similarity
     between sentence & title
10.    Sentence score += Compute cosine similarity
     between sentence & description
11.    Sentence score += Compute cosine similarity
     between sentence & Local closed caption
12.  } // End for each Loop
13. }
    
```

Figure 4. Boost Sentence Score Pseudo Code

Table 1. 2 by 2 Sentence Contingence Table

System \ Ideal	In Summary	Not In Summary
In Summary	True Positive (TP)	False Positive (FP)
Not In Summary	False Negative (FN)	True Negative (TN)



$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F - score = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{B^2P + R} \quad (7)$$

## 5. Results

To evaluate the JSMPW, we requested all registered users at the end of the course to send us feedbacks and comments on the MFMCS system in general and specific on the JSMPW integration tool. We received very positive feedback from all users, one of the users commented: "I love this idea, very interesting and ideas like this are needed to pave the way for the future of education". For the summarization application, we have tested many automated summaries to obtain the best configurations that result in the best summary using the ROUGE<sup>4</sup> software package for evaluation of affectivity of summarization. Two key configurations were tested: the local closed caption sentence size, specified in seconds in the video time line, and the sentence score selection threshold, specified as a percentage above the sentences score average within each discussion. Figure 5 demonstrates one of the ROUGE measures (ROUGE-L) for two sets of summaries (Enhanced, and basic). The enhanced set represents summaries that have our contribution (TFIDF Penalization and Boost Sentence Score) algorithms enabled. While the basic set has the baseline algorithm (no enhancements). For each set we have, on the X-axis, four different threshold values between 0% and 40% above average. In this result set we have the closed caption boundary set to 30 seconds around the annotation time of the discussion. From Figure 6, we demonstrate that our enhanced algorithm generates better summaries, and the best configurations that generates the best summary is when we select all sentences above the final score average (Threshold = 0.0%) and have the closed caption boundary set to 30 seconds.

Once we obtained the best MFMCS summary, we ran ROUGE system with different configurations to test our MFMCS automated summary with different ROUGE measures. We have in Figure the different ROUGE measures for the following common ROUGE configurations: (-m) to apply Porter stemming algorithm, (-s) excluding Stop Words, (-p 0.25) favor Recall when calculating F-Score, and (-f B) get the best matched ideal summary compared to (-f A) to get the average scores of all ideal summaries.

---

<sup>4</sup> <http://www.berouge.com/Pages/default.aspx>

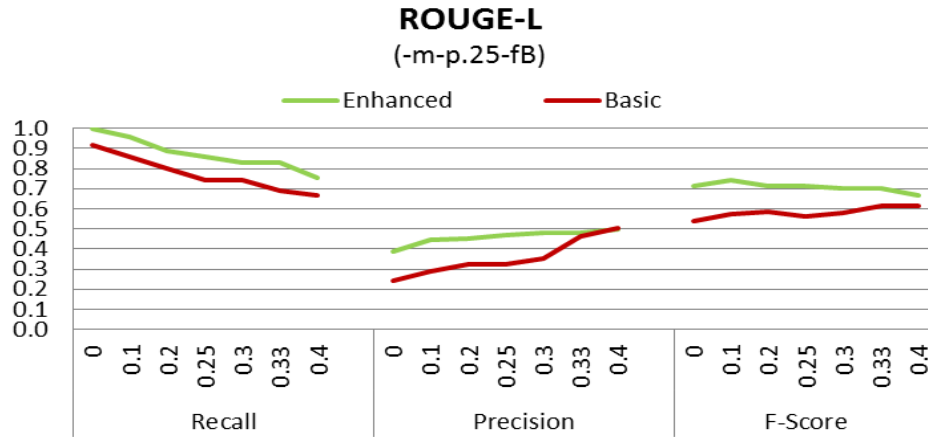


Figure 5. ROUGE-L for 4 Different Configurations of the MFMCS Summaries

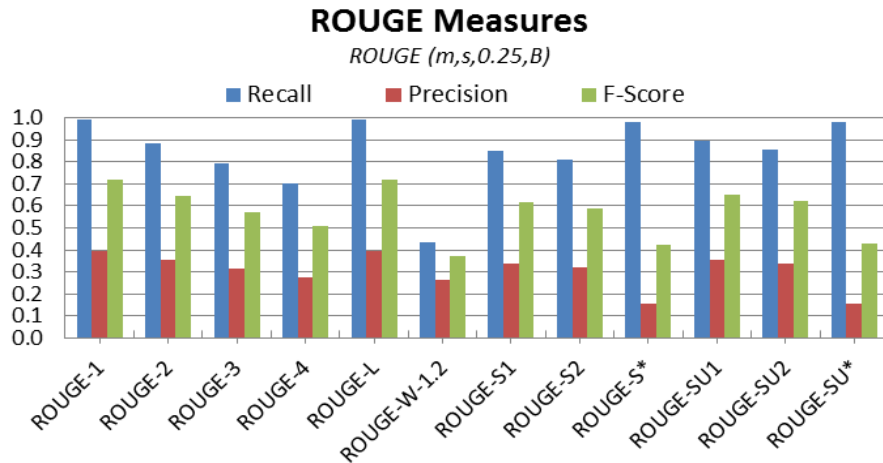


Figure 6. Different ROUGE Measures for the best MFMCS Summary

Finally Figure 7 shows ROUGE-L measure for a comparison between our MFMCS automated summary, the Dragon Toolkit summary<sup>5</sup> (based on the LexRank algorithms), Mead system summary<sup>6</sup>, and five online summarization tools: Automatic Text Summary<sup>7</sup>, Free Summarizer<sup>8</sup>, Online Summarize Tool<sup>9</sup>, Sumplify<sup>10</sup>, and Open Text Summarizer<sup>11</sup>. The purpose of this comparison is to find out the performance of our system compared to other similar extraction summarization systems.

<sup>5</sup> <http://dragon.ischool.drexel.edu/default.asp>

<sup>6</sup> <http://www.summarization.com/mead/>

<sup>7</sup> <http://autosummarizer.com/>

<sup>8</sup> <http://freesummarizer.com/>

<sup>9</sup> <http://www.tools4noobs.com/summarize/>

<sup>10</sup> <http://sumplify.com/>

<sup>11</sup> <http://www.textcompactor.com/> based on the OTS: <http://libots.sourceforge.net/>

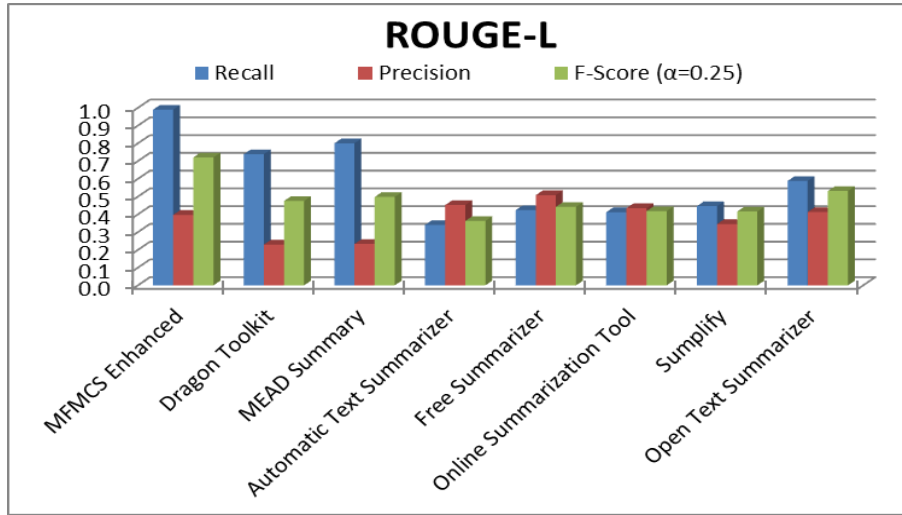


Figure 7. Comparison of MFMCS with other Online Systems

## 6. Conclusion

This article presented an enhanced MOOC architecture. The new MOOC Learners are able to share video contents and engage into cycles of collaborative discussions. The summarization component of the developed MOOC provides learners at the end of the collaboration cycle with the summary of their discussion as a learning outcome. The new MOOC have been tested using representative sample of learners who share interest on multimedia video contents used by some of the university regular courses like “global warming” where the learners expressed their liking as it helps them to write the required essays on these issues. They also noted that this feature made it easier to share their thoughts about a specific point in the video lecture. Figure 8 illustrate the current use case of the developed MOOC where group of learners can share any video content from YouTube and start collaborative discussion where they end with the summary as learning outcome. However, we need to mention that the summarization process is built on the bag of words model and the authors aim to enhance it using sophisticated NLP techniques involving analyzing annotations using part of speech (POS) and other dependency methodologies. That will be left to our future research work.

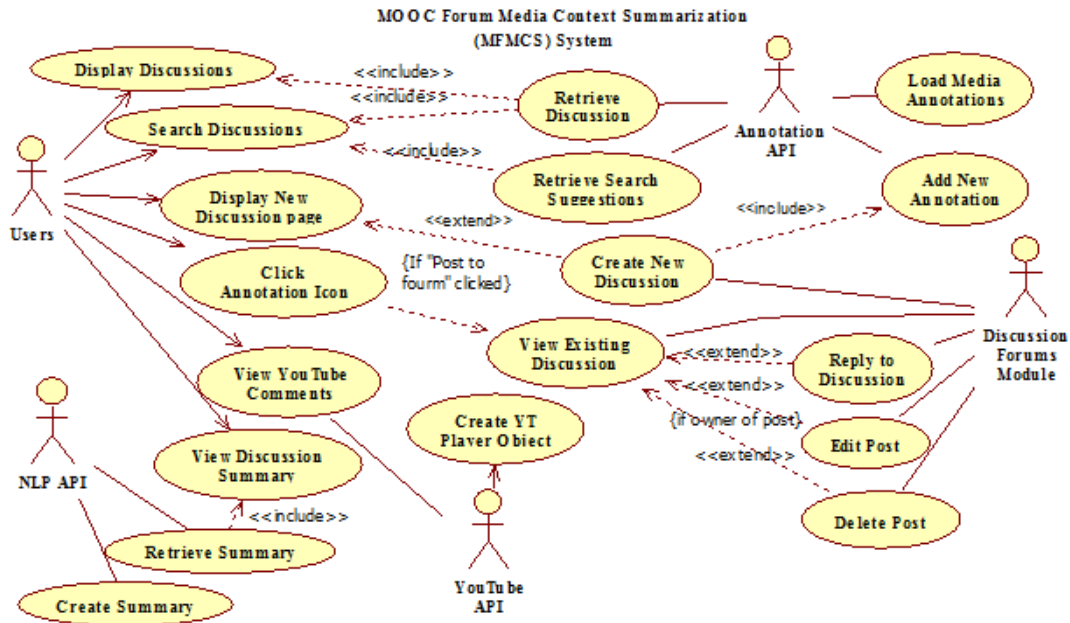


Figure 8. The Use Case Diagram of Our Developed MOOC

## Acknowledgment

This work has been extracted from the MSc Thesis work of Mohannad Al-Mousa under the supervision of Dr. Jinan Fiaidhi.

## References

- [1] J. Ellen, "All about microtext - A working definition and a survey of current microtext research within artificial intelligence and natural language processing", Third International Conference on Agents and Artificial Intelligence, (2011).
- [2] A. Doko, M. Štula and D. Stipaničev, "A recursive TF-ISF based sentence retrieval method with local context," International Journal of Machine Learning and Computing, vol. 3, no. 2, (2013), pp. 195-200.
- [3] B. Sharifi, M.-A. Hutton and J. Kalita, "Summarizing microblogs automatically," in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, (2010).
- [4] M. Hu, A. Sun and E.-P. Lim, "Comments-oriented blog summarization by sentence extraction," in Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, (2007).
- [5] L. Zhou and E. H. Hovy, "On the summarization of dynamically introduced information: online discussions and blogs," in AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, (2006).
- [6] C. D. Manning, P. Raghavan and H. Schütze, "Scoring, term weighting & the vector space model," in Introduction to Information Retrieval, Cambridge University Press, (2009), pp. 109-133.
- [7] J. H. Paik, "A novel TF-IDF weighting scheme for effective ranking," in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, (2013).
- [8] M. d. Kunder, "The size of the World Wide Web," 2 June 2014. [Online]. Available: <http://www.worldwidewebsite.com/>. [Accessed 3 June 2014].
- [9] J. R. Hobbs, "Topic drift," in Conversational organization and its development, vol. 38, Ablex Publishing Corporation, (1990), pp. 3-22.
- [10] C.-Y. Lin, "Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough," in In Proceedings of the NTCIR Workshop, (2004).