Indian Institute of Technology
Bombay

Summer Internship 2018

EKLAVYA (Fundamental Research Group)

Content Tools for Collaborative Communities

Under the guidance of: **Prof. D.B. Phatak**

Team:
Dhanush S R
Naveen T
Saurabh Shwetabh Singh
Aadarsh Singh

Supervisor:
Nagesh Karmali
Mentors:
Firuza Aibara
Abhijit Bonik
Rohit Prasad

Summer Internship 2018 Project

Approval Certificate

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Certificate

The project entitled Content Tools for Collaborative Communities submitted by *Mr. Dhanush S R*, *Mr. Naveen T*, *Mr. Saurabh Shwetabh Singh*, *Mr. Aadarsh Singh* is approved for Summer Internship 2018 programme from 16th May 2018 to 6th July 2018, at Department of Computer Science and Engineering, IIT Bombay

PROF D.B.PHATAK
Dept. of CSE, IITB
Principle Investigator

MR. NAGESH KARMALI
Dept. of CSE, IITB
Project In-Charge

Date: July 6, 2018
Place: IIT Bombay

Contents

List of Figures

List of Tables

1. Acknowledgement

We have taken efforts in this project. However, it would not have been a success without the kind support of various individuals and organizations and we would like to extend our sincere thanks to all of them.

We would like to extend our sincere thanks and regards to *Prof D.B.Phatak* and Indian Institute of Technology Bombay for providing us this unparalleled opportunity to work on this esteemed project. We would also like to extend our sincere thanks to *Mr. Nagesh Karmali* for supervising our project. Our sincere thanks to *Firuza Aibara*, *Abhijit Bonik*, and *Rohit Prasad*, without whom the project would not be this refined and successful.

Finally our thanks to open-source community and others who have willingly assisted us in smallest possible manner in the development of the project.

2. Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dhanush S R
IIT Patna

Saurabh Shwetabh Singh
IIT (ISM) Dhanbad

Naveen T
IIT Patna

Aadarsh Singh
IIT (ISM) Dhanbad

3. Abstract

Collaborative Communities is a system which allows individual users to form communities, participate in various activities defined in the community, and thus, enhance its value for the society in terms of education. Content Tools is a Fundamental Research Group Project aimed at enhancing and expanding the Collaborative Communities system by working on the content that the users of Collaborative Communities create and interact with.

In the following sections, the design and specifications of this system are described along with the achieved results. Also, more insight about the used key technologies is given, justifying why they were necessary for the project. The details for building such a system with Python & its frameworks and Node are explained and the necessary concepts for using it are introduced.

In this project, we successfully worked on four modules viz. Wiki, Interactive Content, Collaborative Editing and Inappropriate Content Filtering. These have been integrated into the Collaborative Communities System.

4. Introduction

System of Collaborative Communities is a platform where the main aim is to bring together people from all around the world on a common platform to share educational content with the community.

For such a platform it is necessary that it should contain proper tools to upload, edit and manage the educational content they wish to share. It is also necessary that the tools must be user friendly and intelligent enough to manage the content collaborated by numerous persons. In such a system it is also necessary to make sure that the content should be appropriate for the viewing of everyone in the community.

As of now the portal for Collaborative Communities offers only 'Basic Article' creation as a resource type. This project aims at facilitating other content types like wiki page, blogs, web presentations, interactive content, real-time collaborative text editing, and testing inappropriateness of contents posted on the site.

4.1. Interactive Content

H5P is a free and open source Content Collaboration Framework based on JavaScript. H5P which stands for HTML5 Package provides tools for creating interactive content. This facilitates easy creation, sharing and reuse of HTML5 content and applications. Authors may create and edit interactive video, presentations, games, advertisements and more. Content may be imported and exported. All that is needed to view or edit H5P content is a web browser.

4.2. Community Wiki Page

A wiki page is a platform where users collaboratively create and modify content. It is written using a simplified markup language and often edited with the help of a rich-text editor. A wiki is run using wiki software, otherwise known as a wiki engine which is a type of content management system.

In Collaborative Communities, the aim is to have a wiki page for each community through which an existing user or one who is a new addition to the community can be made well aware about the community and its aims. This wiki page is open to all community members for editing and working collaboratively for properly conveying the need of the community to the new members or ones who are not yet a part of the community and are unsure of joining it.

4.3. Real-time Collaborative Text Editor

A collaborative text editor is not just a text editor but has an additional feature of accommodating the changes of various authors editing a certain article at the same time simultaneously and at the same time displaying the changes made by any author to every other author, both happening in real-time.

In this vast growing literate community such a tool is of prime importance in a platform where people come together, create and share knowledge and Collaborative Communities is one such platform. This mitigates many save conflicts which occurs when a simple non-collaborative editor is used.

4.4. Inappropriate Content Filter

Inappropriate Content Filter is a tool that can be used to check the appropriateness of a piece of information for a large public audience of various background and level of knowledge before actually publishing the information in front of them.

The tool that we have built for Collaborative Communities is a Deep Learning model that is trained to check the appropriateness of piece of text and predict its probability of being appropriate. Based on this score the publisher can decide whether to publish the article or not.

5. Interactive Content - H5P Module

5.1. Need

The existing Collaboration Communities portal renders only Basic Article creation. The aim has always been to expand, enhance and extend the existing system. Creation of Interactive Content in the portal is one of the ways to go ahead in achieving this aim. This option is made available to us by H5P, an open source package which helps in creation of richer content. This package facilitates creation of interactive videos, interactive presentations, quizzes, interactive timelines and more. The advantage of this is that along with the wide variety of content types, there are tutorials available for creation of each content type which makes it even more user-friendly.[1][2]

5.2. Basis

5.2.1. Problems encountered

H5P is a community driven project currently available only for WordPress, Drupal and Moodle, incorporated as a Moodle plugin, a Drupal module and a WordPress plugin respectively. The Collaborative Communities portal is a django project and there is no official support for creation of H5P contents for django. Thus creation or upload of H5P content is not possible for django websites directly. But the silver lining is that most of the H5P code is framework independent and it is easy to create integrations for new frameworks. Most of the code is javascript. H5P file format consists of a metadata file in JSON format, a number of library files providing features and design for the content and a content folder where textual content is stored in JSON format and multimedia is stored as files or links to files on external sites. To integrate H5P in a platform, the generic H5P code as well as interface implementations and platform specific code is needed.

5.2.2. Solution

Here comes the huge open-source community to the rescue. After a long search for some open-source plugins, H5PP - HTML5 Package Python[3] was found. This is a port for python and the web-framework django provided by Joubel, the creator of H5P which provides development and design services related to H5P. It is a django app written in django version 1.8 and python 2.7. The task in hand was to integrate this port with the Collaborative Communities portal. Working with the concept of keeping the project modular and easy to understand, the h5p module has been separated from the Collaborative Communities portal and runs on a different server and uses the same database as that of Collaboration System which facilitates automatic sharing of sessions in django.

5.3. Examples of interactive content

H5P offers a bulk of interactive content types. Here listed are some examples of interactive content types created using H5P.[4]

5.3.1. Course Presentation

This is a HTML5-based engaging presentation content type which allows users to add multiple choice, fill in the blanks, texts and other types of interactions to their presentations using only a web browser.

Course presentations consist of slides with multimedia, text, and many different types of interactions like interactive summaries, multiple choice questions and interactive videos. Learners can experience new interactive learning material and test their knowledge and memory in Course Presentations.

A typical use of the Course Presentation activity is to use a few slides to introduce a subject and follow these with a few more slides in which the user's knowledge is tested. Thus, this becomes an important tool in the Collaborative Communities portal for users to offer courses with the help of such engaging presentations.

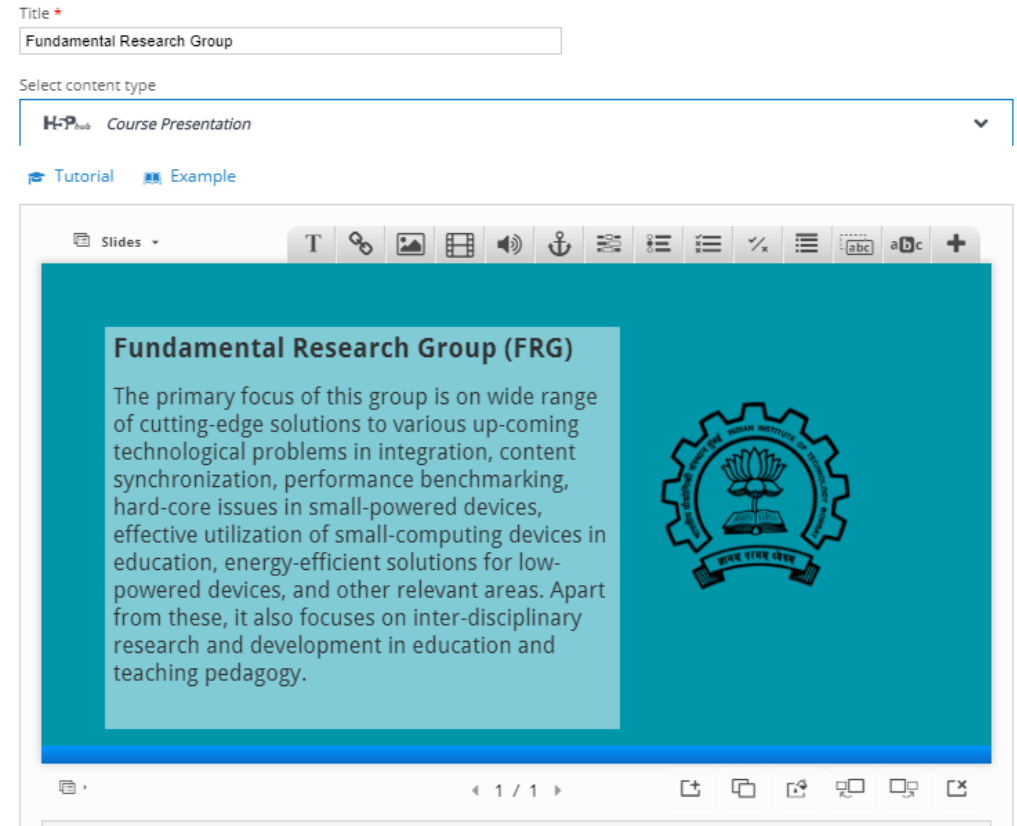

Figure 1: Creation of Course Presentation.

5.3.2. Interactive Video

In this digital era of distant learning, internet is the new classroom for any individual. There are many tools needed to build a course which encourages interactive learning. One of them is interactive video based learning. Collaborative Communities portal now offers creation of interactive video which may be enriched with activities like explanations, extra pictures, tables, Fill in the Blank and multiple choice questions. Quiz questions support adaptability, meaning that you can jump to another part of the video based on the user's input. Interactive summaries can be added at the end of the video.

Figure 2: Example of Interactive Video.

5.3.3. Find the Hotspot

This content type adds to the functionality of courses, giving the users the option to press somewhere on an image and get feedback on whether that was correct or incorrect according to the task description.

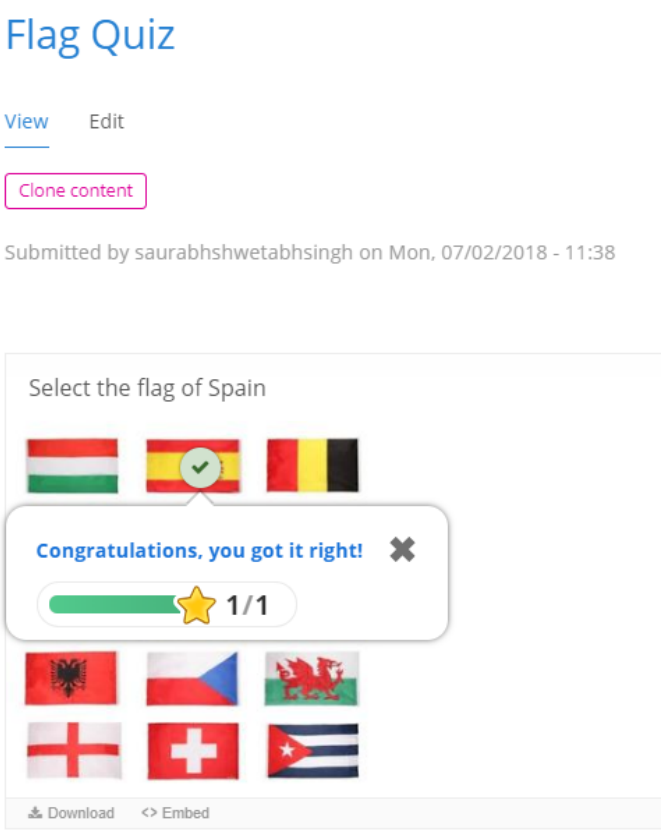

Figure 3: Example of Image Hotspot Quiz.

The teacher/author can upload an image and define various hotspots corresponding to details or sections of the image. Hotspots can either be defined as correct or incorrect, and the author provides appropriate feedback text in both cases.

5.4. Workflow

5.4.1. H5PP

H5PP - HTML5 Package Python was cloned and minor modifications were made so that it can be incorporated into the Collaborative Communities.

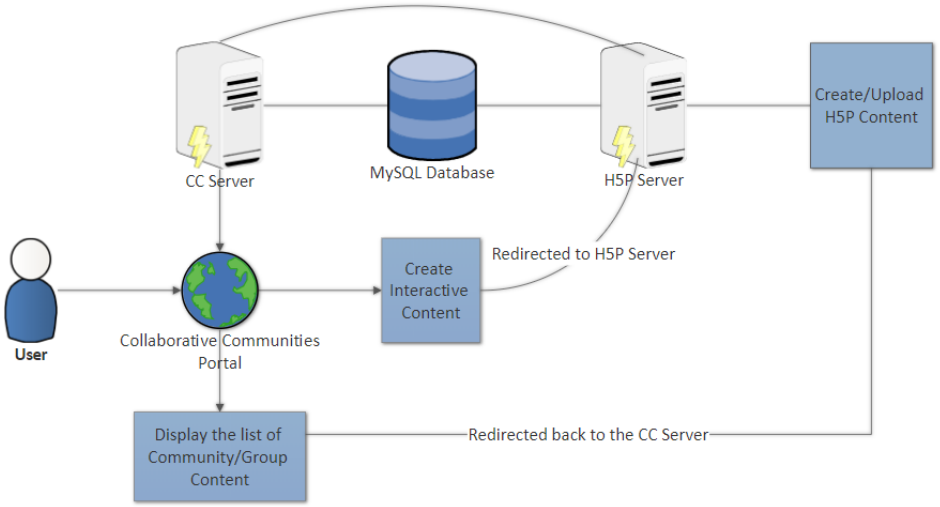

Figure 4: Workflow of the H5P Module.

The User Interface (UI) was changed to that of the Collaborative Communities. Now, a user can easily create interactive content for a community or a group.

H5PP requires the official release of h5p libraries to be uploaded in the system. This is a one-time step and has to be done once the system has been installed. After uploading the libraries in the H5P server, the system will be ready for the users to create/upload and view H5P contents in the Collaborative Communities portal.

5.4.2. Creating Interactive Content

The option for 'creating interactive content' is present under the 'create a resource' option. On clicking it, the user is redirected to the h5p server where he can create/upload interactive content from a bulk of content types. After creation, the user is redirected back to the Collaborative Communities server to the list of community/group content.

5.4.3. Viewing Interactive Content

A user who is a member of the community/group can view the interactive content from the existing community/group content list. A click on the name of the content redirects the user to the h5p server where he can view the content.

6. Wiki

6.1. Need

The communities created in the collaborative communities portal have some mission and vision which they look forward to follow for shaping a better future of the community. The aim of the community should be properly defined so that the new enthusiastic users understand the core objective of the community. Such content should be more precise and should have a proper structure to enhance readability. This spurs the need of more compact tool which has these features, easily plug able and also doesn't increase the load on the system.

6.2. Django-wiki

A wiki is a platform where users can collaboratively modify and structure their contents. Django-wiki is a wiki with permission system, revision control and file attachments. Django-wiki is easily pluggable and has a very good user interface. Django-wiki's markdown syntax makes the wiki more readable and well organised. Django-wiki has been integrated with the collaborative communities portal. When a new community is created, it's wiki page is also created along with it. All the information regarding that community can be edited in that page.

6.3. Structure of Django-wiki

The django-wiki has a tree like strucure of wiki pages. A wiki can have children wikis which can be exploited to our use of creating group wikis for a particular community. The tree stucture of wiki can be viewed in the figure 5.

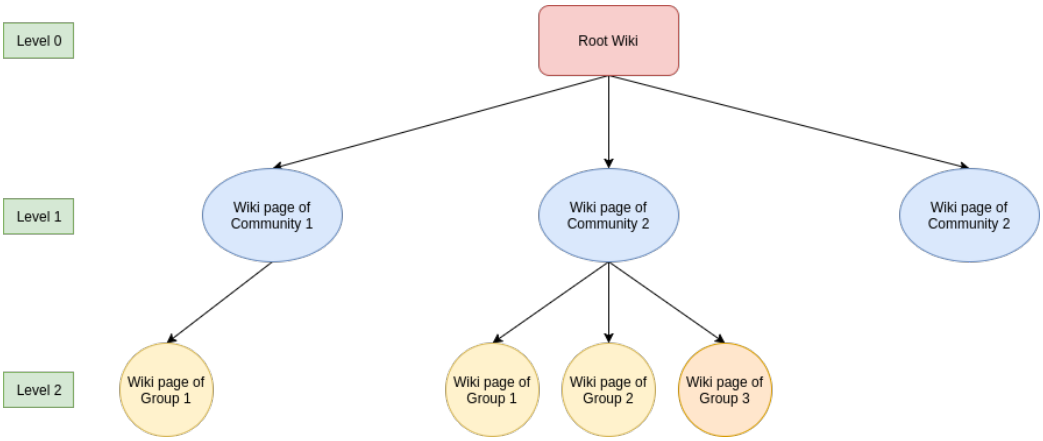

Figure 5: Structure of Django-wiki

6.4. Workflow

In the collaborative communities portal, a community can be created either directly by the admin of the portal or when the admin approves the request of creation of a community by an authenticated user. For creating a wiki, without user interference we used SQL insertion statements into the database for filling the required tables. The wiki page can be viewed by any user by going to the community's page and then to the wiki page of the community by clicking on the wiki button on the community view page. Since the django-wiki has revision control system, a user can also see the different stages of the community's wiki. This provides a better wholistic view of the community.

7. Real time Collaborative Editor

7.1. Need

The initial collaboration system had CK editor as an editor. Although it had all the required tools which a rich text-editor should have, it failed to serve the purpose of "Collaborative editing" which can be called the soul of the Collaboration system. The required objective for the project was to replace CK editor with an editor that supports collaborative editing and has an equally efficient interface.

7.2. Basis

7.2.0.1. Problem faced to implement collaborative editing

Let's say there are two friends Alice and Bob who wish to edit an article collaboratively. Consider the following diagram that represents the initial state of their document:

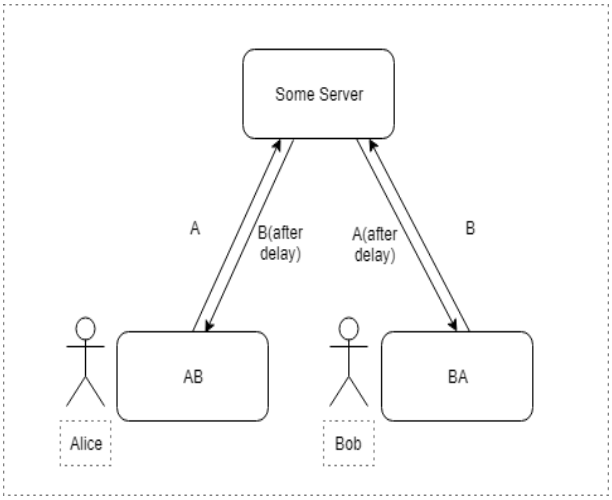

Figure 6: Example

As shown, Alice has written A while Bob has typed B. Now, after a short

time interval their insertions reach each other. So, Alice gets B and her final article becomes AB whereas Bob's becomes BA. This was unexpected. To resolve the error, a solution needs to be found that helps to maintain the coherence the documents at the two ends.

7.2.0.2. Available Solution

To counter the above problem, the information passed over the servers need to be modified. Two possible ways are as follows:

- 1. Operation Transform (OT)
- 2. Conflict-free replicated data type (CRDT)

7.3. Available Editors

Fortunately, there are umpteen open source rich text editors and plethora of libraries for operational transform. Our task was to find the best. To find out the best, the best possible was to test some of the most popular ones. So we chose the following for the test :

- Summernote
- Quilljs
- Firepad
- Prosemirror
- Etherpad

Out of the above mentioned editors, summernote does not have the feature of collaborative editing just like CK editor but we chose to test it because of the features that it provides and its simplicity. Rest of them are collaborative.The following table gives a summary of the tests that we performed:

Editor	Database Used/Compatible	Extra Features	Comments
Firepad	Firebase,	-	Firebase has limited use
Prosemirror			Complicated workflow
Summernote	-	Video, image can be inserted	
Quilljs	MongoDB		
Etherpad	Mysql, DirtyDB,	Plethora of Useful Plugins Available	

Table 1: Tested Editors

7.4. Theory

7.4.1. Changesets

The underlying foundation of etherpad is changeset. A changeset represents a change to a document. It can be applied to a document to produce a new document. When a document is represented as a list of changesets, it is assumed that the first changeset applies to the empty document. Now, to make things easier to implement, changesets are represented in the following manner:

$$(n \rightarrow n')[c1, c2, c3...]$$

where n and n' are the initial and final lengths of the document respectively, and $[c1, c2, c3..]$ is an array of n' characters that describes the document after the changes.

In a changeset, integers represent retained characters in the original document while the characters represent the insertions. For instance, say in an empty document, we have to write "$HelloWorld!$". The same can be represented using changeset in the following way,

$$(0 \rightarrow 12)["HelloWorld!"]$$

Let's call this changeset as A. If we wish to change this document to "$Hello\,guys$!", another changeset (say B) can be inserted as follows:

$$(12 \rightarrow 11)[0-5, "guys", 11]$$

Thus, the document becomes a composition of two changesets A and B represented as AB or $A.B$ So, the whole document can be represented as a changeset (say C) where $C = AB$ i.e.

$$C = (0 \rightarrow 11)["Hello\,guys!"]$$

7.4.2. Realtime collaboration with the help of changesets

7.4.2.1. Merging of Changesets

Composition of changesets, as can be seen is a cake walk but how do these changesets help in realtime document editing? For this we need to know about the merging of changesets. Suppose Alice and Bob make two different changes to the same document at the same time. It is impossible to compose these changes. For example, if we have the document X of length n, we may have

$$A = (n \rightarrow n_a)[...n_a\,characters]$$

$$B = (n \rightarrow n_b)[...n_b\,characters]$$

where

$$n \neq n_a \neq n_b$$

. It is impossible to compute $(XA)B$ because B can only be applied to a document of length n, and (XA) has length n_a. Similarly, A cannot be applied to (XB) because (XB) has length n_b. This is where merging comes in. Merging takes two changesets that apply to the same initial document (and that cannot be composed), and computes a single new changeset that preserves the intent of both changes. The merge of A and B is written as $m(A, B)$. Now, in the etherpad systems, for collaborative editing to work,

$$m(A, B) = m(B, A)$$

.

7.4.2.2. Follows function

Having gone through the concept of merging, let's jump into our initial experiment of collaboratively editing a document. Again we have Alice and Bob editing a document say X rows function. Now, after finishing her changes, Alice sees the document as XA (composition of the changeset A with the initial changeset X) whereas Bob sees it as XB. So why does merging not help us here? The reason is simple, merging could easily merge A and B into AB to make the resultant document as XAB(for both Alice And Bob) but the issue is Alice already sees the document as XA and Bob sees the document as XB.

None of them see the plain document X. So instead of merging the changeset, we need to modify the received changeset in a way such that composing the modified changeset with the other user's changeset and then with the document i.e

$$X.(A.B')$$

for Alice

$$X.(B.A')$$

for Bob does the same job as composing the initial changesets(A and B) and the initial document X i.e.

$$X.(AB)$$

for both Alice and Bob.

One major point to note is that for realtime editing to take place the following condition should be fulfilled

$$XAB' = XBA'$$

This is what the follows function helps us to achieve.The definition of follows function f is such that $Af(A,B) = Bf(B,A) = m(A,B) = m(B,A)$. When we compute $f(A,B)$

- Insertion in A becomes retained characters in $f(A,B)$
- Insertions in B becomes insertions in $f(A,B)$

- Retain whatever characters are retained in both A and B.

To explain the concept in a much better way, I would like to present an example from the document given in the Etherpad Wiki's documentation on changesets[8].

Example: Suppose we have the initial document $X = (0 \rightarrow 8)[\text{"}baseball\text{"}]$ and user A changes it to "basil" with changeset A, and user B changes it to "below" with changeset B. We have

$$X = (0 \rightarrow 8)[\text{"}baseball\text{"}]$$

$$A = (8 \rightarrow 5)[0-1, \text{"}si\text{"}, 7]$$

$$B = (8 \rightarrow 5)[0, \text{"}e\text{"}, 6, \text{"}ow\text{"}]$$

First we compute the merge $m(A, B) = m(B, A)$ according to the constraints $m(A, B) = (8 \rightarrow 6)[0, \text{"}e\text{"}, \text{"}si\text{"}, \text{"}ow\text{"}] = (8 \rightarrow 6)[0, \text{"}esiow\text{"}]$ Then we need to compute the follows

$$B' = f(A, B) \; and \; A' = f(B, A)$$

$$B' = f(A, B) = (5 \rightarrow 6)[0, \text{"}e\text{"}, 2, 3, \text{"}ow\text{"}]$$

$$A' = f(B, A) = (5 \rightarrow 6)[0, 1, \text{"}si\text{"}, 3, 4]$$

We can now double check that,

$$AB' = BA' = m(A, B) = (8 \rightarrow 6)[0, \text{"}esiow\text{"}]$$

.

7.4.3. Clients and Changesets

At any moment in time, a client maintains its state in the form of 3 changesets[8].

- The client document looks like A · X · Y , where A is the latest server version, the composition of all changesets committed to the server, from this client or from others, that the server has informed this client about. Initially

$$A = (0 \rightarrow N)[< initial\, document\, text >]$$

.

- X is the composition of all changesets this client has submitted to the server but has not heard back about yet. Initially

$$X = (N \rightarrow N)[0, 1, 2, ..., N-1]$$

, in other words, the identity, denoted as I_N .

- Y is the composition of all changesets this client has made but has not yet submitted to the server yet. Initially

$$Y = (N \rightarrow N)[0, 1, 2, ..., N-1]$$

.

7.5. Implementation

The big question that arises is - How is etherpad implemented? The answer to the above question can be given in two parts -

- Implementation of changesets
- Sending and Receiving information

7.5.1. Changeset Strings

Changesets are implemented in the form of changeset strings. Changeset strings are a way of representing changesets in the system. Here we will

discuss the form of changesets that etherpad uses to store the document. An example of such a changeset string is as follows:

$$Z : 5g > 1|5 = 2p = v * 4 * 5 + 1\$x$$

This changeset represents inserting a bold letter “x” into the middle of a line. The string consists of:

- a letter Z (format version identifier)
- a series of opcodes (punctuation) and numeric values in **base 36** (the alphanumerics)
- a dollar sign ($)
- a string of characters used by insertion operations (the “char bank”)

7.5.1.1. Operations

If we separate out the operations and convert the numbers to base 10, we get:

$$Z : 196 > 1|5 = 97 = 31 * 4 * 5 + 1\$"x"$$

Here are descriptions of the operations, where capital letters are variables:
Descriptions

- $: N$ - Source text has length N (must be first op)
- $> N$ - Final text is N (positive) characters longer than source text (must be second op)
- $< N$ - Final text is N (positive) characters shorter than source text (must be second op)
- > 0 - Final text is same length as source text
- $+N$ - Insert N characters from the bank, none of them newlines

- $-N$ - Skip over (delete) N characters from the source text, none of them newlines
- $= N$ - Keep N characters from the source text, none of them newlines
- $|L + N$ - Insert N characters from the source text, containing L newlines. The last character inserted MUST be a newline, but not the (new) document's final newline.
- $|L - N$ - Delete N characters from the source text, containing L newlines. The last character inserted MUST be a newline, but not the (old) document's final newline.
- $|L = N$ - Keep N characters from the source text, containing L newlines. The last character kept MUST be a newline, and the final newline of the document is allowed.
- $*I$ - Apply attribute I from the pool to the following $+$, $=$, $|+$, or $| =$ command. In other words, any number of $*$ ops can come before a $+$, $=$, or $|$ but not between a $|$ and the corresponding $+$ or $=$[10].

If $+$, text is inserted having this attribute. If $=$, text is kept but with the attribute applied as an attribute addition or removal. Consecutive attributes must be sorted lexically by (key,value) with key and value taken as strings. It's illegal to have duplicate keys for (key,value) pairs that apply to the same text. It's illegal to have an empty value for a key in the case of an insertion ($+$), the pair should just be omitted.

Characters from the source text that are not accounted for are assumed to be kept with the same attributes.

7.5.1.2. Additional Constraints

Consecutive $+$, $-$, and $=$ ops of the same type that could be combined are not allowed. Whether combination is possible depends on the attributes of the ops and whether each is multiline or not. For example, two multiline deletions can never be consecutive, nor can any insertion come after a non-multiline insertion with the same attributes. "No-op" ops are not allowed,

such as deleting 0 characters. However, attribute applications that don't have any effect are allowed. Characters at the end of the source text cannot be explicitly kept with no changes; if the change doesn't affect the last N characters, those "keep" ops must be left off. In any consecutive sequence of insertions ($+$) and deletions ($-$) with no keeps ($=$), the deletions must come before the insertions. The document text before and after will always end with a newline. This policy avoids a lot of special-casing of the end of the document. If a final newline is always added when importing text and removed when exporting text, then the changeset representation can be used to process text files that may or may not have a final newline.

7.5.1.3. Attribution String

An "attribution string" is a series of inserts with no deletions or keeps. For example, "$*3+8|1+5$" describes the attributes of a string of length 13, where the first 8 chars have attribute 3 and the next 5 chars have no attributes, with the last of these 5 chars being a newline. Constraints apply similar to those affecting changesets, but the restriction about the final newline of the new document being added does not apply.

7.5.2. Workflow

After having understood the theory and the system overview, let us see how Etherpad functions. Etherpad-lite uses socket connections for sending data to and fro between client and server. For creating and maintaining these socket connections it uses socket.io which is a popular open source library. It uses socket because it's a two way connection so both client and server can send message to each other.Also this is a persistent connection so sending data to and fro just happens in realtime. Between etherpad server and database, communication is through standard TCP calls.[9]

It uses the collaboration system database's store table for storing the data permanently. It stores pad content(a pad in etherpad is analogous to an article) and it's revisions in database. Besides MySQL, Etherpad-lite provides the functionality to use databases like SQLite, Redis, MongoDB, Cassandra etc. Apart from storing pad content in database it keeps the pad content in

memory as long as there is an active client on pad. So basically for opening a pad it loads the data from database and stores it in memory, then for every operation it uses the data stored in memory.It does not store the revisions content in memory. For each client it also stores the session information in memory.In session information it basically stores the pad id, revision number and author info of that client.

7.5.2.1. Message Types

Each message , which client sends to server or server sends to client, contains a message type. Using message type client or server determines that message contains what information and it is for what action.Below are the some common message types and their use:

- **CLIENT_READY**: Message of this type is sent from client to server when it asks for the pad information just after establishing the socket connection.
- **CLIENT_VARS**: Message of this type is sent from server to client when it receives CLIENT_READY from client. It contains pad information.
- **USER_CHANGES**: Message of this type is sent from client to server when client sends a changeset to server.
- **ACCEPT_COMMIT**: Message of this type is sent from server to client on accepting the client changeset.
- **NEW_CHANGES**: Message of this type is sent from server to client to let it know about other user changesets if multiple users are working on same pad.

7.5.2.2. Etherpad instance in Collaboration System Portal

Since etherpad is running on a nodejs server and Collaboration system is based on a Django server, we embedded the pad instance on the article_create and article_edit page using iframe. Now on the nodejs server side, through http calls etherpad-lite loads the supporting scripts and html.

It never loads the actual content of pad through it. When the scripts get loaded, it creates a socket connection with server. And after creating connecting it sends a message of type CLIENT_READY with information like padID and other security related token and passwords. Server on receiving this message verifies the user and store it's info in sessionInfo in local memory and then send the pad details to client with message type CLIENT_VARS . After that, corresponding communication socket joins the room of that padID. (Because server should know which all clients are connected to a pad so by adding socket to room this information remains on server.Using this room concept now you can find which all clients are connected to a particular pad and can send a message to all of them or a particular one). After reading this data which is sent with message type CLIENT_VARS browser loads the pad into memory and render it. The django server simply displays the iframe that contains this pad. A workflow diagram of the same is represented below:

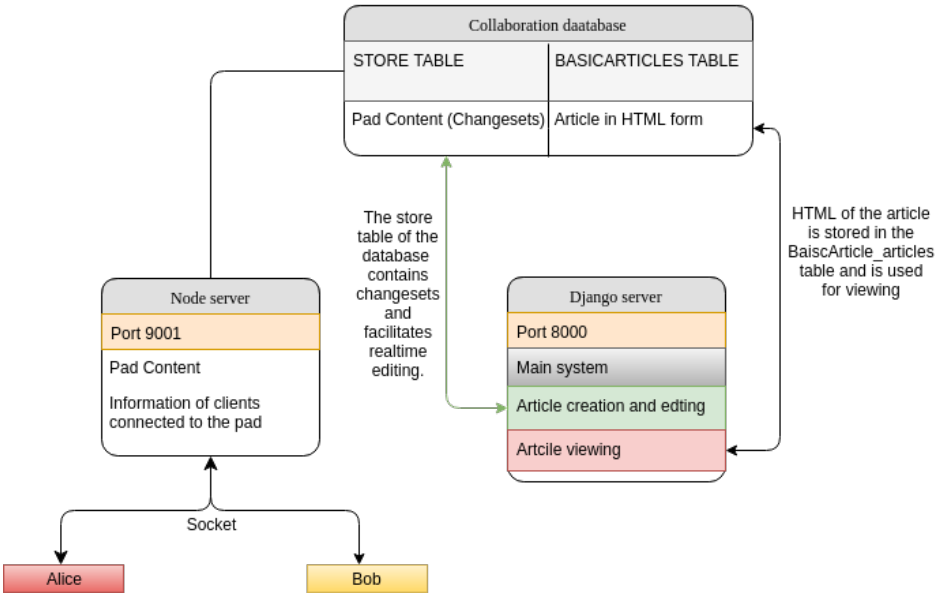

Figure 7: Workflow of collaborative editor- Etherpad

8. Inappropriate Content Filters

8.1. Need

Collaborative Communities portal provides a platform for people to collaborate from all corners of the world. In this portal, people will be sharing their views, and expressing their feelings. Though the information shared will be useful to the people around the globe, some of the contents can be vulgar, offensive, abusive or insulting. Thus, there is need for filtering the inappropriate contents using a more effective filtering mechanism.

8.2. Tools Used

8.2.1. Pytorch

PyTorch is an open source machine learning library for Python, based on Torch, used for applications such as natural language processing. It is primarily developed by Facebook's artificial-intelligence research group and Uber's "Pyro" software for probabilistic programming is built on it.[5]
The main two high-level features of Pytorch is:

- GPU acceleration for Tensor Computations
- Deep Neural Networks based Autodiff System

8.2.2. Numpy

Numpy is a Open-Source Python library for Scientific Computation in Python. This provides support for high-dimensional matrices both dense and sparse and also for high-level mathematical functions on these arrays.

8.2.3. GloVe

GloVe stands for Global Vectors for Word Representation. It is an unsupervised learning algorithm to obtain vector representation for words. It is trained on a global word-word corpus based on statistics of occurrences of words in the corpus. It also captures the meaning between various words.

E.g: king - man + women = queen

We have used a pre-trained GloVe matrix provided by Stanford NLP (released under Public Domain Dedication and License) at the following link:

http://nlp.stanford.edu/data/wordvecs/glove.6B.zip

8.2.4. Pandas

Pandas is a Python Library used for Data Manipulation and Analysis. It offers data-structures and manipulation tools for numerical tables and time-series.
Some of the important features of the library are:

1. DataFrame object for data manipulation with integrated indexing.
2. Tools for reading and writing data between in-memory data structures and different file formats.
3. Data alignment and integrated handling of missing data.
4. Reshaping and pivoting of data sets.
5. Label-based slicing, fancy indexing, and subsetting of large data sets.
6. Data structure column insertion and deletion.
7. Group by engine allowing split-apply-combine operations on data sets.
8. Data set merging and joining.

9. Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
10. Time series-functionality: Date range generation[3] and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.[6]

8.2.5. bcolz

bcolz provides columnar, chunked data containers that can be compressed either in-memory and on-disk. Column storage allows for efficiently querying tables, as well as for cheap column addition and removal. It is based on NumPy, and uses it as the standard data container to communicate with bcolz objects, but it also comes with support for import/export facilities to/from HDF5/PyTables tables and pandas dataframes.[7]

8.3. Theory

The model of Inappropriate Content Filter broadly contains 2 main parts.

- Encoder
- Fully Connected Network

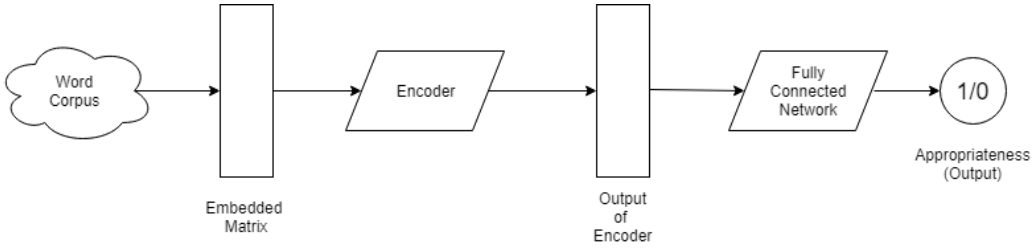

Figure 8: Visualization of the Inappropriate Content Filter Model.

8.4. Encoder

The Encoder part of the model contains two parts.

- Word Embedding
- Long Short Term Memory Network

8.4.1. Word Embedding

8.4.1.1. Why Word Embeddings are needed?

Many Machine Learninig models and most Deep Learning Models cannot read strings as input in its raw form. It can read only numerical inputs. Hence word embeddings is of prime importance in Natural Language Processing where both input and output are strings. As huge amount of data is present in text format it is also necessary to extract knowledge from it to build applications and use them for the betterment of mankind.

8.4.1.2. What is Word Embedding?

Word Embedding is a Deep learning network used to convert words from any languages in to vectors/tensors of fixed shape. There are two type of Word embedding networks viz., Word2Vector, Global vectors for Word Representation(For detailed explanation please refer *Section* 8.2.3).

8.4.1.3. Word2Vector

Word2Vec is not a single algorithm but it is a combination of two algorithms viz., CBOW(Continuous Bag of Words) and Skip-Gram Model. Both of them are shallow neural networks where target variables of a word is also word(s). They both learn weights which act as word representation.

8.4.1.4. Process of converting word to vectors.

The entire process of Word Embedding starts from converting the data from text format to column vector of length equal to the length of vocabulary used which(one hot encoding).Then feeding this matrix to the embedding network returns a tensor/vector of fixed size.

8.4.2. Long Short Term Memory

8.4.2.1. Recurrent Neural Networks

When Neural Networks are used to identify the various incidents occurring in movie it classifies them only based on current information rather than using information from little while ago along with current input. This is a major short-coming in Neural Networks.
In general, every time a human thinks he doesn't start from the scratch. He/she takes some part of the information from the previous thinking and continue on that information. In a similar fashion Recurrent Neural networks or RNNs unlike other Neural Networks, loops a part of its output into its next input allowing the information to persist.

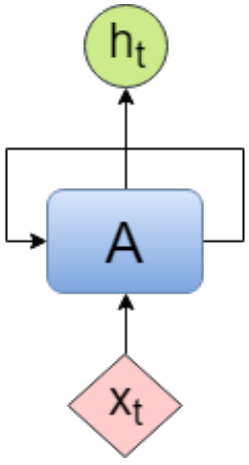

Figure 9: A Recurrent Neural network

Here the RNN(A) takes input x_t and returns output h_t and there is also a

loop within the A. This can be visualized as Series of A one after the other in a row and can be visualized as below.

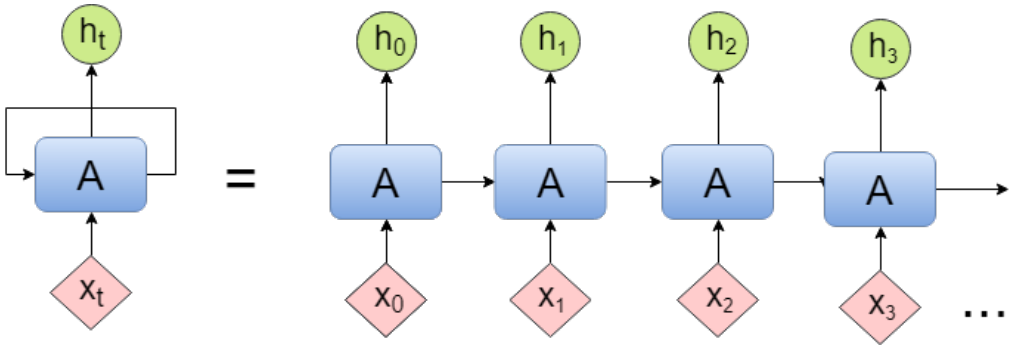

Figure 10: Unfolding of RNN Network

8.4.2.2. Problem with RNNs

Let us consider an example predicting words in sentence *"The clouds move in the sky"*. In this example, an RNN can predict that the word after *"the"* must be *"sky"* as it is obvious from the sentence. Here the RNN can handle the dependency which is over a short range which contains the relevent information.

But when it comes to the example of *"He is a poet. He writes poem"*. For predicting the word *poem*, RNN has to take care of two things. First is the nature of word to be placed after *"writes"* ie., it is a kind of literary work and then the actual word. A simple RNN might be able to predict that it must be literary work like a story or a novel or even a poem. But may not be able to predict that it is *"poem"*, as it has to handle Long range dependencies which is not possible for a traditional Recurrent Neural Network. This shortcoming of the RNN is taken care by using a *LSTM* network.

8.4.2.3. LSTM Network

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They

were introduced by Hochreiter Schmidhuber (1997), and were refined and popularized by many people in following work.They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn![12]

A regular RNN will have a $tanh$ layer as it repeating unit. But LSTMs along with $tanh$ it has 4 layers as a repeating unit. A simple visualization of an LSTM can seen in figure 11.

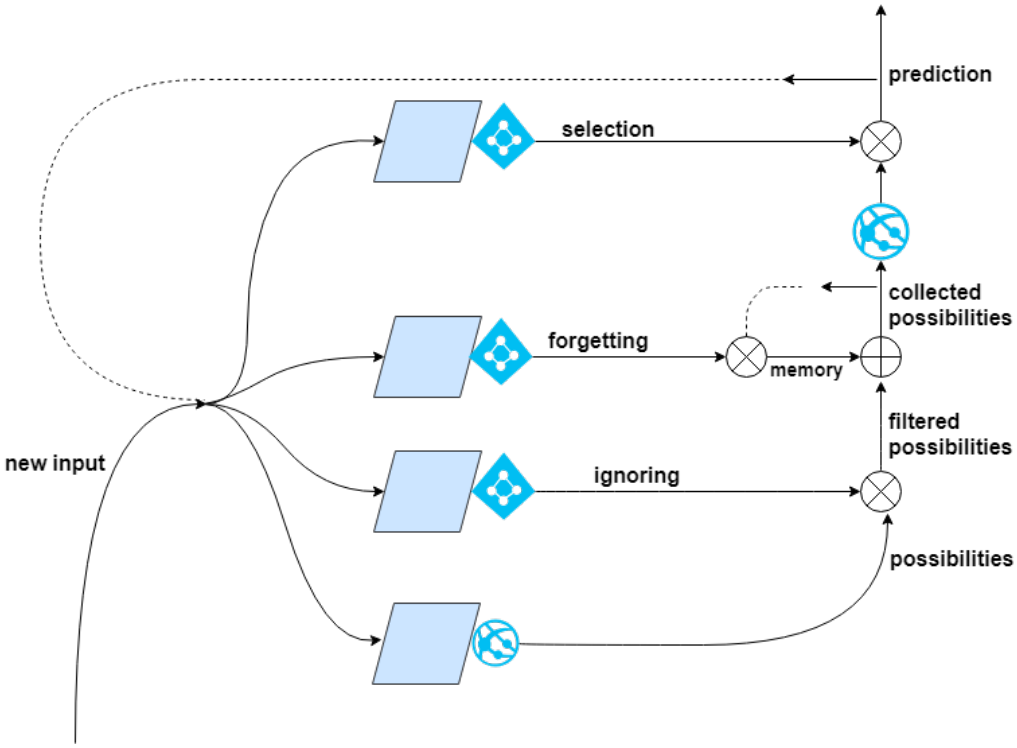

Figure 11: Visualisation of a LSTM Network

Here ⊗ represents $tanh$ layer and ◈ represents a $sigmoid$ layer. The sigmoid function is

$$sigmoid(z) = \frac{1}{1 - e^{-z}}$$

The dotted lines represent that it is being fed to network in next iteration. First the input is fed to normal $tanh$ layer with Neural Network to get all possibilities. The $sigmoid$ layer with Neural network get the features that are to be ignored and it point-wise multiplied with all posibilities to get filtered ones. Now possibilities from previous iteration along with learned information from forgetting network is point-wise added to current filtered possibilities to get collected possibilities. Now the selection network selects the most probable output based on result of all other networks and predicts a single prediction which is again fed to network in the next iteration.

8.5. Fully Connected Network

Fully Connected Networks are basic type of Deep Learning models. In a Fully Connected Network, each and every node of a particular layer is connected to every node of previous and leading layers. Although it is a basic neural network, it is a powerful network and has increased the accuracy of Machine Learning models drastically.A F.C.N contains input layer, hidden layers and output layer.

Forward Pass is a set of operations on a Fully Connected network which takes input and transform it into output space. Backward Pass is another set of operations on F.C.N that is used mainly to train the model based on Cost Function which tells the difference between the actual output and expected output.

8.6. Workflow

The main model that we have used is, first we converted input in text format to vectors of fixed size($hidden\_size$ in our case) through Embedding layer. then we fed this to a LSTM layer that outputs a tensor of length $hidden\_size$ that is first passed through a Dropout layer and fed into a Fully Connected network whose output layer is size one.

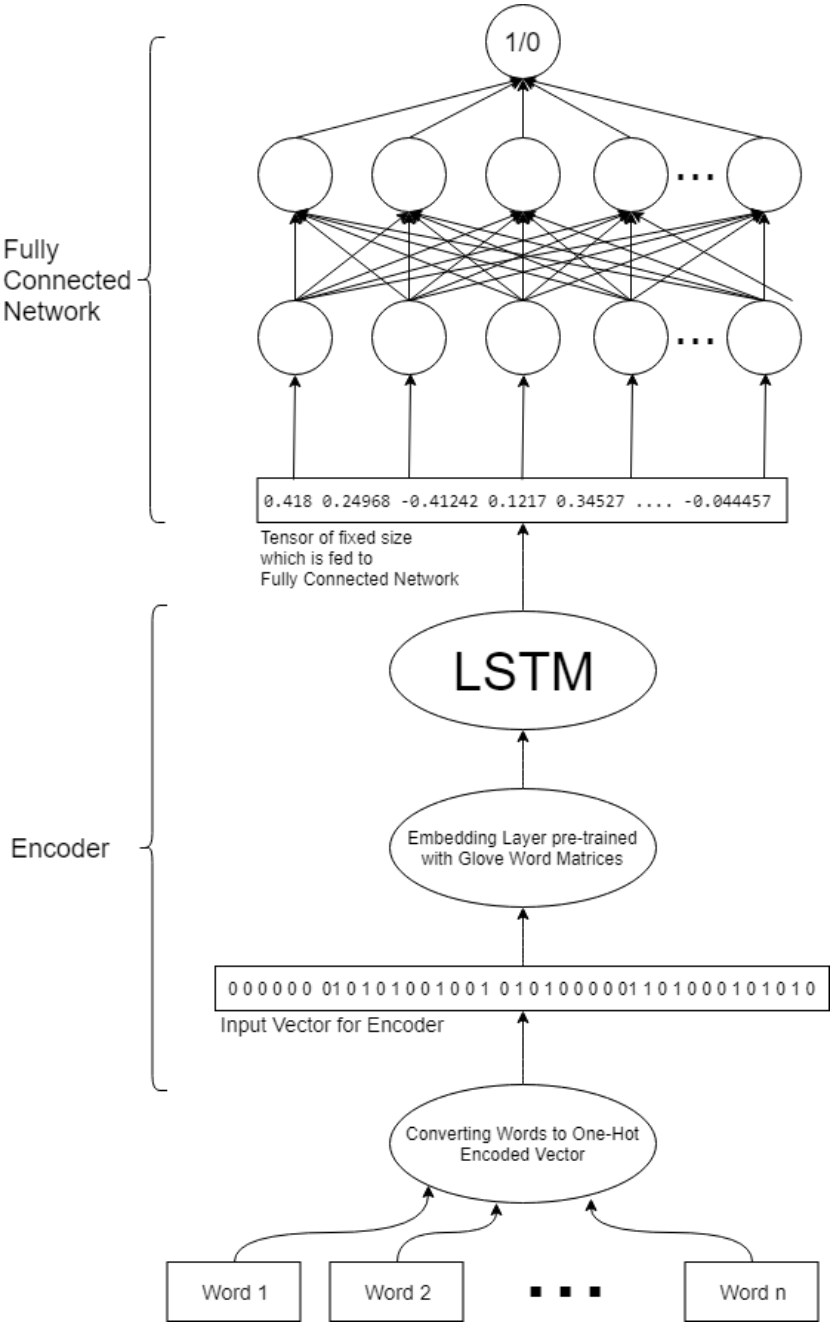

Figure 12: Deep Learning Model used for Inappropriate Content Filter

8.7. Implementation

We have implemented the above mentioned model in $python$. For implementing deep learning model we have used the Pytorch library for Python.

The dataset to train the model is obtained from Kaggle (https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge). This dataset contains 159570 rows of data. And it is classified into 6 categories viz., toxic, severe_toxic, obscene, threat,insult, identity_hate. We first combined all categories and tagged the data as either appropriate or not. Then we randomly selected 15,000 of appropriate examples and 15,000 of inappropriate examples. Then we used 60% of this data to train the model and 40% to test the model.

Then we imported glove matrix and made a dictionary from this which was used to create Embedding Layer with weights from the Glove matrix. The following is the implementation of the above model using Pytorch library for Python.

```
\\returns the weights_matrix required to create an
\\embedding layer with pretrained glove vectors
class EncoderRNN(nn.Module):
    def __init__(self, weights_matrix, input_size,hidden_size):
        super(EncoderRNN, self).__init__()
        self.hidden_size = hidden_size
        self.embedding, num_embeddings, embedding_dim=
            create_emb_layer(weights_matrix, True)
        self.gru = nn.GRU(embedding_dim, hidden_size)

    def forward(self, input, hidden):
        embedded = self.embedding(input).view(1, 1, -1)
        output = embedded
        output, hidden = self.gru(output, hidden)
        return output, hidden

    def initHidden(self):
        return torch.zeros(1, 1,self.hidden_size,device=device)
```

```
#FullyConnectedNetwork
class FullyConnectedNN(nn.Module):
    def __init__(self, input_size):
        super(FullyConnectedNN, self).__init__()
        self.dp1 = nn.Dropout()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.tanh1 = nn.Tanh()
        self.dp2 = nn.Dropout()
        self.fc2 = nn.Linear(hidden_size, hidden_size)
        self.tanh2 = nn.Tanh()
        self.dp3 = nn.Dropout()
        self.out = nn.Linear(hidden_size, 1)

    def forward(self, input_tensor):
        input_tensor = input_tensor.view(-1)
        o1 = self.dp1(input_tensor)
        a1 = self.fc1(o1)
        h1 = self.tanh1(a1)
        o2 = self.dp2(h1)
        a2 = self.fc2(o2)
        h2 = self.tanh2(a2)
        o3 = self.dp3(h2)
        a3 = self.out(o3)

        return a3
```

9. Dockerisation

9.1. What is a docker?

Docker is a container platform that performs operating-system-level virtualization which is also referred to as containerization.

Docker is aimed at running software packages or containers which are isolated from each other and use their own set of tools and libraries; they can communicate through well-defined channels. All containers use the same kernel and are therefore more lightweight than virtual machines. Containers are created from images which specify their precise contents.

Docker brings into use a high level API to provide lightweight containers which allows a single server to run several containers simultaneously. Building on top of facilities provided by the Linux kernel (primarily cgroups and namespaces), a Docker container, unlike a virtual machine, does not require or include a separate operating system. Instead, it relies on the kernel's functionality and uses resource isolation for CPU and memory,and separate namespaces to isolate the application's view of the operating system. Docker accesses the Linux kernel's virtualization features either directly using the libcontainer library.[11]

9.2. Advantages

1. Consistent development environments for the entire team. All developers use the same OS, same system libraries, same language runtime, independent of the host OS. The development environment is exactly the same as the production environment.
2. If you're having a hard time building / compiling the application code, then build it inside Docker. This primarily applies to developers using MacOS and Windows.
3. With Docker, you can use different versions of same programming language without having to resort to all the hack around for the language on your machine.

4. Deployment is easy. If it runs in your container, it will run on your server just the same. Just package up your code and deploy it on a server with the same image or push a new Docker image with your code in it and run that new image.

9.3. Docker Compose

Docker Compose is a tool for defining and running several docker containers simultaneously. It uses YAML files to configure the application's services and performs the creation and start-up process of all the containers with a single command. The docker-compose command-line interface utility allows users to run commands on multiple containers at once, for example, building images, and running containers. The docker-compose.yml file is used to define an application's services and includes various configuration options.[11]

The advantage is that it caches the configuration used to create a container. When a service is restarted that has not changed, Compose re-uses the existing containers. Re-using containers means that changes to your environment can be made very quickly.

Using Docker Compose is basically a three-step process:

1. Define your app's environment with a Dockerfile so it can be reproduced anywhere.
2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
3. Run docker-compose up and Compose starts and runs your entire project.

Figure 13: Services running through Docker-Compose.

9.4. Workflow

9.4.1. Initial Configuration of Collaboration System

In the initial system, there are two docker containers, 'db' - the MySQL database and 'web' - the Collaboration System, which is run using the docker-compose.yml file. Inside this file, the database configurations, port on which the 'web' runs, the MySQL image to be used for 'db', are specified. 'db' and 'web' run simultaneously when the start-up takes place. The Collaboration System has its Dockerfile used while building the 'web' image.

9.4.2. Implementation

In our implementation, 'h5p' image is built from the Dockerfile inside the H5P project directory and is specified in the docker-compose.yml of Collaboration System so that it runs simultaneously with 'web' and 'db'. Same is the case for the 'node' image built from Dockerfile in the EtherpadLite directory. 'web', 'h5p', 'db', 'node' run on ports 7000, 8000, 3306, 9001

respectively as shown in figure 13. What we aim to achieve that all these services run on separate containers in an isolated environment

10. Conclusion

Before the start of the project, the collaborative communities portal had Basic Article, Courses as their content creation types.The articles that were created did not support real-time collaborative editing, but was not of real-time collaboration which was a major drawback. Also, the description of the community in the community view page was not sufficient for explaining the vital and core elements of the community in an orderly fashion.

So, in this project we implemented PyEtherpad which allows real-time collaborative editing of articles; and we added creation of a new content type, interactive contents using HTML5 Package (H5P) which is impressive as the user experiences is bend towards interactive learning.

For proper description of a community we have integrated Django-wiki which has more readable and eye-catching markdown syntax for describing the community's objective in a structured manner.

Also, to avoid misuse of the Collaborative Communities platform, we implemented an inappropriate content filter which flags abusive or toxic content. Thus, preventing it from getting published or accessible to other users.

11. Future Work

11.1. Interactive Content

For Interactive content, H5PP (Python port for H5PP) has been integrated with Collaborative Communities portal.The source code is written in Python 2.7 and Django 1.8 whereas the Collaborative Communities portal is written in Python 3.6 and Django 1.11. Though both have no conflicts as both are being hosted on different servers, it would be better if the source code of H5PP is updated to python 3.6 and Django 1.11. As of now, the user is redirected to another server (H5P server) for creating and viewing Interactive contents. This can be solved either by using `iframe`, which makes an API call to the H5P server and displays the required contents for creating or viewing interactive contents; or by updating the code base of H5PP, both the code bases can be merged which will be more compact. Also, there is always scope for enhancing the User Interface (UI) for the H5P Server.

11.2. Wiki

In the collaborative communities portal, a community has a mandatory wiki page when it is created.The groups that are created can also have a wiki page under the community's wiki page as a sub-page. This is where the django-wiki will be more useful in structuring the wikis of the communities and groups.A wiki page for a group has not been implemented yet, but by following similar steps as followed in creation of communities wiki page, a wiki page for a group can also be created.

11.3. Real Time Collaborative Editor

- Currently the etherpad implementation supports insertion of images and formulae but the API function getHTML(), when called, inserts an asterisk(*) instead of the image in the HTML document. Efforts can be made to tackle this issue.

- While editing, the document is saved in the $store$ table (node server) when the user makes any changes to the document but the HTML of the same is autosaved in the $BasicArticle\_articles$ table in regular intervals through ajax requests. It is because events inside the iframe can't be detected due to the same origin policy. Sending requests at regular intervals can be a problem when the number of clients to serve is more.

11.4. Inappropriate Content Filter

- Build an architecture to automatically flag the article regarding its appropriateness.
- Notify the author regarding the same
- Scope for improving the accuracy by using a better data set and optimization techniques.
- Make the model dynamic i.e., it should take review from users by allowing them to mark content inappropriate and training itself from these reviews.

12. Selenium Test For Project Based Learning

12.1. Project based learning

Project-based learning (PBL) is a student-centred pedagogy that involves a dynamic classroom approach in which it is believed that students acquire a deeper knowledge through active exploration of real-world challenges and problems.Students learn about a subject by working for an extended period of time to investigate and respond to a complex question, challenge, or problem.It is a style of active learning and inquiry-based learning.

This project is one of Fundamental Research Group's project which is in its initial stages.The basic workflow of the Web service system for PBL has been finalized and few Client Side pages have been implemented.Our prime aim is to check the user interface created, by using selenium.

12.2. Tests and their results

S.no	Action	Expected result	Comments
1	Click on Create a Project button in the teacher's dashboard.	Create a Project page should open in the same tab	As expected
2	Click on the Cancel button.	Redirected to the teacher's dashboard.	As expected
3	Click on Create button without filling the details	An alert-box appears with following message "Please enter all information to continue".	Not as expected, it redirects to previous window rather then dashboard
4	Clicking on the save button after filling the title and the description should	Redirect to the Create a Module Page with the Project title in the breadcrumb.	As expected

Table 2: Create Project Page - Test cases

S.no	Action	Expected result	Comments
1	Click on the signup option at the bottom of the login page	Signup page should open in the same tab	As expected
2	Hover the mouse on "Create your account" button.	The button background should change from light blue colour to dark blue colour	As expected
3	Click on "Create your account" button	The following alert message should pop up, 'Enter All Information'	As expected
4	Fill in all the details, check 'Agree the terms and policy' checkbox and click on 'Create new account' button	When the 'Create new account' button is clicked, the page should redirect to the respective dashboard page	As expected, but it has a glitch. If a user clicks on the button before he has entered any info and then fills the details and clicks on the button again, the button doesn't work.
5	Hover mouse on "Sign in" button	When mouse is hovered on "Sign in" button, the button text should get underlined and text should change from light blue to dark blue colour.	As expected
6	Click on "Sign in" button	Login page should open in same tab	As expected

Table 3: Sign Up Page - Test cases

S.no	Action	Expected result	Comments
1	Fill the username and password. Select login as teacher and then submit using login button.	Teacher's dashboard should open.	As expected
2	Filling in random username and password who has not signed up and then submit using login button	When a new user tries to sign in without registering, an error message should be displayed that he is not a authenticated member.	Not as expected, neither error message is displayed nor the user is redirected to the sign up page.
3	Click on the "Sign in" button without filling any details.	An error message should be displayed below the header part of the webpage.	As expected
4	Hover mouse on "Sign up" button	When mouse is hovered on "Sign up" button, the button text should get underlined and text should change from light blue to dark blue colour.	As expected
5	Click on "Sign up" button	User should be redirected to the signup page.	As expected

Table 4: Login Page - Test cases

S.no	Action	Expected result	Comments
1	Click on Add New module button.	A new row with the fields title, description, dependencies and students should add up.	As expected
2	Click on Remove Last field.	The last added row should vanish provided that it is not the only row.	As expected
3	Click on Save button.	The following message should be displayed below the header part of the webpage (with a light pink color background and a dark red coloured bottom border): We couldn't create the project. Please add atleast one module in the project. All the mandatory fields(with *) should be highlighted in red colour.	Not as expected because no error message shown. Instead empty modules are submitted.
4	Click on the save button after filling the title and the description.	Alert box shows up with the message "Project successfully added"	As expected

Table 5: Create module Page - Test cases

13. Appendix

13.1. Installation

13.1.1. H5P Module for Collaborative Communities

In order to install this module, follow the steps given below: For development installation -

1. Clone this repository

```
git clone https://github.com/fresearchgroup/Community-Content-Tools.git
```

2. Create a virtual environment, with python 2.7 , django 1.8

```
virtualenv venv --python=python2.7
```

3. Activate the virtual environment

```
source venv/bin/activate
```

4. Inside the virtual environment, install the following:

```
pip install -r requirements.txt
```

5. Install the H5P plugin in the virtual environment

```
pip install H5PP-0.1.9.tar.gz
```

6. Install mysql and configure the database

```
sudo apt-get install mysql-server
sudo apt-get install libmysqlclient-dev
python sql_reset.py
```

7. Run the server by going to the project's main directory

```
cd H5P/myproject
python manage.py runserver
```

For Docker installation -

1. Install Docker and Docker-Compose from
Docker –

```
https://docs.docker.com/install/linux/docker-ce/ubuntu/
```

Docker Compose –

```
https://docs.docker.com/compose/install/
```

2. Clone the repositories

```
git clone https://github.com/Content-Tools-
Team/Collaboration-System.git
```

```
git clone https://github.com/Content-Tools-
Team/Community-Content-Tools.git
```

3. In the H5P (Community-Content-Tools) directory,

```
sudo docker build -t h5p_image .
```

4. In the Collaboration-System directory,

```
sudo docker-compose build

sudo docker-compose up db

sudo docker exec -i <db-container-image-name> mysql -
u<username> -p<password> django < collab-updated.sql

sudo docker-compose up
```

5. Go to https://h5p.org/sites/default/files/official-h5p-release-20170301.h5p and download the official h5p libraries.

6. Go to *http://yourdockerip:8000/h5p/libraries* and upload the downloaded libraries and select proceed.

13.1.2. Real-time Collaborative Text-editor

For development installation -

1. Install virtualenv

```
sudo pip3 install virtualenv
```

2. Clone the project from github

```
git clone https://github.com/fresearchgroup/Collaboration-
System.git
```

3. Create a virtual env

```
virtualenv collab -p python3
```

4. Activate the virtual environment

```
source collab/bin/activate
```

5. Install the requirements.txt

```
pip3 install -r Collaboration-System/requirements.txt
```

6. Install mysql server –

```
sudo apt-get update
sudo apt-get install mysql-server
sudo apt-get install libmysqlclient-dev
mysql -u root -p

Enter password=root

mysql> create database collaboration;
mysql> use collaboration;
mysql> source collab.sql
```

7. Create a .env inside CollaborationSystem and paste the following -

```
sudo nano .env
```

```
SECRET_KEY=myf0)*es+lr_3l0i5\$4^)^fb&4rcf(m28zven+oxkd6!
(6gr*6
DEBUG=True
DB_NAME=collaboration
DB_USER=root
DB_PASSWORD=root
DB_HOST=localhost
DB_PORT=3306
ALLOWED_HOSTS= localhost
GOOGLE_RECAPTCHA_SECRET_KEY=6Lfsk0MUAAAAAFdhF-dAY-
iTEpWaaCFWAc1tkqjK
EMAIL_HOST=localhost
EMAIL_HOST_USER=
EMAIL_HOST_PASSWORD=
```

```
EMAIL_PORT=25
EMAIL_USE_TLS=False
DEFAULT_FROM_EMAIL=collaboratingcommunity@cse.iitb.ac.in
SOCIAL_AUTH_GOOGLE_OAUTH2_KEY=735919351499-
ajre9us5dccvms36ilhrqb88ajv4ahl0.apps.googleusercontent.com
SOCIAL_AUTH_GOOGLE_OAUTH2_SECRET=I1v-
sHbsogVc0jAw9M9Xy1eM
APIKEY=
NODESERVERURL=Your IP address
NODESERVERPORT=9001
```

8. Clone the following directory:

```
git clone http://github.com/dhanushsr/etherpad-lite
cd etherpad-lite/
./bin/run.sh
```

9. Install PyEtherLite–

```
git clone http://github.com/dhanushsr/PyEtherpadLite
cd PyEtherpadLite
python setup.py install
cd ..
```

10. Paste the apikey from APIKEY.text from etherpad-lite folder in the .env file. Do all the migrations going back to django directory–

```
python3 manage.py migrate
```

11. Runserver –

```
python3 manage.py runserver
```

For docker installation -

1. Install Collaboration Communities from https://github.com/fresearchgroup/Collaboration-System/

2. Clone etherpad-lite from https://github.com/ether/etherpad-lite

```
git clone https://github.com/ether/etherpad-lite
```

3. Clone the current directory and place the contents of etherpad-lite folder in etherpad-lite root directory.

4. Install Node JS from https://nodejs.org/

5. Install Docker from https://docs.docker.com/install/linux/docker-ce/ubuntu/#set-up-the-repository

6. Install Docker-Compose form https://docs.docker.com/compose/install/

7. In the etherpad-lite folder run the following commands

```
sudo docker build -t etherpadlite .
```

8. Setup database from Collaboration Communities for Docker

9. Setup environment for Django app.

```
sudo docker run -p 9001:9001 etherpadlite
sudo docker ps
```

Using the container image from above

```
sudo docker exec -i <image-name> cat APIKEY.txt
```

10. Place the above string in the .env.docker in Collaboration-Communities folder for APIKEY variable.

11. Place the IP address of docker in the .env.docker in Collaboration-Communities folder for NODESERVERURL variable.

12. Continue with the setup of Collaboration Communities.

References

[1] H5P - Create and Share Rich HTML5 Content
https://h5p.org

[2] H5P – Wikipedia
https://en.wikipedia.org/wiki/H5P

[3] H5PP - HTML5 Package Python
https://github.com/DrClockwork/H5PP

[4] Examples and Downloads | H5P
https://h5p.org/content-types-and-applications

[5] Pytorch – Wikipedia
https://en.wikipedia.org/wiki/PyTorch

[6] pandas(Software) – Wikipedia
https://en.wikipedia.org/wiki/Pandas_(software)

[7] Blosc/bcolz – GitHub
https://github.com/Blosc/bcolz

[8] Easysync documentation
https://github.com/ether/etherpad-lite/raw/master/doc/easysync/easysync-full-description.pdf

[9] How etherpad-lite, a real time collaborative editor, works? – Geek Dirt,
http://geekdirt.com/blog/how-etherpad-works/

[10] Changeset Protocol
http://policypad.readthedocs.io/en/latest/changesets.html#attributes

[11] Docker – Wikipedia
https://en.wikipedia.org/wiki/Docker_(software)

[12] Understanding LSTM Networks
http://colah.github.io/posts/2015-08-Understanding-LSTMs/