Amogh Ravish | Elma Pinto | Elton D’Souza | Pradeep Nagendra
MAY 14, 2014
# Table of Contents

Overview .......................................................................................................................... 2  
Project Objectives ............................................................................................................. 2  
Targeted Users .................................................................................................................... 2  
Competitor Analysis .......................................................................................................... 3  
Business Case ..................................................................................................................... 3  
Novelty ................................................................................................................................. 4  
System Architecture .......................................................................................................... 4  
System Implementation .................................................................................................... 5  
  Course Data Collection ...................................................................................................... 5  
  Tweet Data Collection ....................................................................................................... 6  
  Twitter Analytics Data ...................................................................................................... 6  
  Server Hosting .................................................................................................................. 7  
Visualizations ..................................................................................................................... 8  
  HighCharts ....................................................................................................................... 8  
  Word Cloud ..................................................................................................................... 8  
  Most Positive/Negative Tweets ........................................................................................ 8  
  Global distribution ........................................................................................................... 8  
API’s Implemented ........................................................................................................... 8  
Twitter Sentiment Analysis ............................................................................................... 9  
Contribution of Team members ....................................................................................... 10  
Site Screenshots .............................................................................................................. 11  
Key Takeaways .................................................................................................................. 14  
References ......................................................................................................................... 14
Overview

Online education is increasingly gaining popularity among students and professionals in general who may not want to go back to school. To facilitate this learning while sitting anywhere and at any time, massive open online courses began to gain ground in the field of online education.

A **Massive Open Online Course** is an online course aimed at unlimited participation and open access via the web. In addition to the traditional course materials such as videos, readings, and problem sets, MOOCs provide interactive user forums that help build a community for students, professors, and teaching assistants and professionals. MOOCs are a recent development in distance education which began to emerge in 2012.

Although early MOOCs often emphasized open access features, such as connection and open licensing of content, structure, and learning goals, to promote the reuse and remixing of resources, some notable newer MOOCs use closed licenses for their course materials, while maintaining free access for students (Wikipedia).

We intend to use the power of MOOCs and create a stop destination for all seekers of online courses.

Project Objectives

- To leverage the power of existing MOOCs all over
- To allow users to obtain information about all courses at one location
- To empower users with the knowledge of what’s trending and which courses are most popular in which region
- To inform users about the usefulness of a course based on analytics done on feedback from previous course takers
- To enable users to read the tweets based on the most positive and negative comments about a particular course so that they can make informed decisions

Targeted Users

The main users of this system are students, professionals and industry experts in the information technology domain. However, the main target while building the system were students all over the world.
Competitor Analysis

Figure 1: Competitor Analysis

Business Case

The business model can be explained as follows:

Our website can obtain revenue from two major sources.

1. **MOOC Providers**: Each time the user clicks on the MOOC Provider’s course and user takes the course offered by that provider, we get a cut off from it. This process occurs for every course listed on our website that is charged by the provider.

2. **Recruiters**: We intend to provide the information we get from our analytics to the recruiters so that they can derive meaning information from it and use it. Information such as which course is being taken by which kind of students, who has taken the majority of courses in this field, which courses are trending in which university of region etc. This way the recruiters will know where students can be picked from quickly. It can possibly save them some time in the recruiting process as this will provide them with important and meaningful information.

**Financial Projections**: Considering we have 40% courses obtained through MOOC providers which are paid. Currently, we have roughly about 1000 courses. Let’s consider 400 of them are paid. Initially, we could take 3% of the course fee as our profit and increase the number after the MOOCs get more users through our website. The profit would thus increase in that case as shown in the figures below.

We tried to create dummy values of our projection in excel and created a visualization of how the profit would look with a rise from about 3% to 5 to 6 to 8% each month for the courses of same prices.
The graph depicts a promising trend.

Novelty

User feedback in the form of analytics:

Our website provides the user with an opportunity to make informed decisions about courses depending upon the analytics we provide. Our competitors so far have not provided twitter sentiment analysis to users so that they can see what’ trending the most in a region, what is the overall review about a course? Is it positive or negative? What is the most positive comment spoken about a topic or a course or a MOOC provider? What is the most negative comment? We answer all these questions for our user.

System Architecture

Our website MOOC Central is a mash up of API’s, some of which were dynamically integrated and some which had to be scraped. The scraped data was mainly MOOC courses which were loaded into our MySQL database. Our system was used Amazon EC2 Web server for hosting and our site is currently accessible at http://54.186.253.203/new/. You can also use http://mooccentral.tk/new/. The Website is also mobile friendly.
System Implementation

Course Data Collection
We began by gathering data from various MOOC providers about the courses offered. We collected data by two mechanisms:

- **REST APIs**: Khan Academy, Coursera and MIT OCW offered REST APIs that returned course information in JSON format. We parsed the JSON and stored the data in a MySQL database.
- **Web Scrapers and Crawlers**: We also scraped and crawled course information from edX, Udemy and Academic Earth by using the simple HTML DOM parser and PHP.

The structure of the table is as follows:

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Datatype</th>
<th>Length/Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CID</td>
<td>INT</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>SOURCE</td>
<td>VARCHAR</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>SUBJECT</td>
<td>VARCHAR</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>COURSE_NAME</td>
<td>VARCHAR</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>COURSE_DESC</td>
<td>TEXT</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SESSION_NO</td>
<td>BIGINT</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>COURSE_PHOTO</td>
<td>VARCHAR</td>
<td>100</td>
</tr>
<tr>
<td>8</td>
<td>COURSE_LINK</td>
<td>VARCHAR</td>
<td>200</td>
</tr>
<tr>
<td>9</td>
<td>COURSE_LESSON</td>
<td>VARCHAR</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>COURSE_INSTRUCTOR</td>
<td>VARCHAR</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>COURSE_START_DATE</td>
<td>VARCHAR</td>
<td>100</td>
</tr>
<tr>
<td>12</td>
<td>COURSE_END_DATE</td>
<td>VARCHAR</td>
<td>100</td>
</tr>
<tr>
<td>13</td>
<td>COURSE_RATING</td>
<td>DECIMAL</td>
<td>2.2</td>
</tr>
</tbody>
</table>

The search page of the site queries this table to retrieve results.
Tweet Data Collection

We collected Tweets using the Twitter Streaming API. The tweets were collected based on a set of hashtags and usernames related to courses. These were stored in a database table containing 64 attributes related to each tweet, including timestamp data, tweet contents, hashtags, user data, geolocation data, etc.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Datatype</th>
<th>Length/Set</th>
<th>Unsigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>tweet_id</td>
<td>INT</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>created_at</td>
<td>VARCHAR</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>id</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>id_str</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>text</td>
<td>VARCHAR</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>source</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>truncated</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>in_reply_to_status_id</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>in_reply_to_status_id_str</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>in_reply_to_user_id</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>in_reply_to_user_id_str</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>in_reply_to_screen_name</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>user_id</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>user_id_str</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>user_name</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>user_screen_name</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>user_location</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>user_url</td>
<td>VARCHAR</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>user_description</td>
<td>VARCHAR</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>user_protected</td>
<td>VARCHAR</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>user_followers_count</td>
<td>INT</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>user_friends_count</td>
<td>INT</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>user_listed_count</td>
<td>INT</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5: A snippet of the Twitter data table*

Twitter Analytics Data

To assess the sentiments of tweets, we created various views of queries to obtain counts of sentiment scores.
Figure 6: View to calculate sentiments of MOOCs

```
1 |
2 select case when hashtag_uppercase in ("MOOC", "MOOCs") THEN "MOOC" END "Hashtag",
3   WHEN INSTR(hashtag, ' ') THEN TRIM(LEADING ' ' FROM hashtag)
4   WHEN INSTR(hashtag, ' - ') THEN TRIM(LEADING ' - ' FROM hashtag)
5   WHEN INSTR(hashtag, ' ') THEN TRIM(LEADING ' ' FROM hashtag)
6   WHEN INSTR(hashtag, ' - ') THEN TRIM(LEADING ' - ' FROM hashtag)
7 GROUP BY hashtag, hashtag, hashtag, hashtag
8 ORDER BY 1
9 )
```

Figure 7: Views for analysis

Server Hosting

The website is hosted on an Amazon EC2 server running Windows Server 2008. The server is running an Apache HTTP server and a MySQL server. The tweet collection script was constantly running to collect real-time tweets. The PHP scripts to collect data via REST APIs and scraping are run periodically to keep course data up to date.

Figure 8: Tweet collection script running on server
Visualizations
The tweets were classified into positive, negative and neutral sentiments along with scores to measure the extent of positivity and negativity of sentiments. These results were stored in the MySQL database.

HighCharts
We used HighCharts API to visualize the data. We retrieved data from the views created to obtain analysis scores and percentage distributions. This data was then passed to the HighCharts API which generated the various visualizations shown on the website.

Word Cloud
We created two views to retrieve the top trending hashtags and users on Twitter, and the most popular words used to describe MOOCs. These words with their frequencies were then fed into a PHP function which generated a word cloud.

Most Positive/Negative Tweets
We calculated the overall score of a tweet by subtracting the negative score from the positive score. We then sorted the tweets to arrive at the top positive and negative tweets.

We used the username and profile picture information to display the tweets with the user details.

Global distribution
We obtained the users’ location from the Twitter streaming API. This usually contained the city name of the user. In order to plot this information on a map, we used the CloudMate geocoding API to obtain the latitude and longitude of the corresponding location.

Using these coordinates, we plotted the locations on a map using Tableau Public.

API's Implemented

<table>
<thead>
<tr>
<th>MOOC API's</th>
<th>Social Media API's</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khan Academy</td>
<td>YouTube</td>
</tr>
<tr>
<td>Open Course Ware</td>
<td>Facebook</td>
</tr>
<tr>
<td>Coursera</td>
<td>Twitter Post</td>
</tr>
<tr>
<td>Udemy</td>
<td>Twitter Timeline</td>
</tr>
<tr>
<td>edX</td>
<td>Twitter Streaming</td>
</tr>
</tbody>
</table>

| HighCharts                  | GeoCoding                               |

Figure 9: Implemented APIs
Twitter Sentiment Analysis

We implemented the Twitter Streaming API and extracted about 200,000 tweets over the span of two weeks. One set of tweets was related to various MOOC providers such as Coursera, MIT-OCW, Khan Academy, edX and Udemy, and another set to various computer science topics. We carried out sentiment classification on the two sets of data using a Python script which in turn uses SentiWordNet as the lexical dictionary. The script is based on the research paper ‘Reviews Classification Using SentiWordNet Lexicon’ (Alaa Hamouda).

The various phases of our Sentiment Analysis are shown in the figure below.

Tokenization and Speech Tagging: This process divides the tweets text into tokens like numbers, punctuation and words, and creates a tag as an annotation based on the type of word in the tweet (noun, verb, etc.)

WordNet Word Sense Disambiguation: In this case, for every word in a tweet, a synset in WordNet that best matches the word is selected for the tweet. A simple WSD algorithm has been implemented for this purpose using Natural Language Toolkit (NLTK).

SentiWordNet Interpretation: A search for sentiment score corresponding to every synset is done in SentiWordNet.
Sentiment Orientation: A positive score ($s^+$) and a negative score ($s^-$) is calculated by adding the positive and negative scores of each word in a tweet. In the formula, ‘$i$’ represents each word in a tweet ‘$t$’.

Tweet Classification: The sentiment of a tweet ‘$st$’ is calculated by the formula:

$$s_t = \begin{cases} 
\text{positive if } s^+_i > s^-_i \\
\text{negative if } s^+_i \leq s^-_i 
\end{cases}$$

The scores of tweets were stored in the MySQL database and used in the High Charts API to create visualizations as explained in the Novelty section of this document.

Contribution of Team members

Elton D’Souza
- Front end development - PHP, HTML5, jQuery
- Twitter data collection
- Visualizations - HighCharts
- REST APIs - Khan Academy, MIT OCW, Coursera, YouTube

Pradeep S Nagendra
- Database Design - Creation of Tables and Views
- Analytics using Tableau
- Geocoding API
- EC2 setup

Amogh Ravish
- Web Scraping/Crawling - Udemy, edX, PHP
- Database Administration
- APIs - Twitter Timeline, YouTube
- Sentiment Analysis - SentiWordNet

Elma Pinto
- Data Scraping
- APIs - Facebook, Google+, Twitter, Coursera
- UI design
- EC2 setup
- Business Case Development
- Presentation Preparation
Site Screenshots

Figure 11: Home page

Figure 12: Search page with course and video results
Figure 13: Description on clicking a course name

Figure 14: Analytics page - 1
Figure 15: Analytics page - 2

Figure 16: Analytics page - 3
Key Takeaways

Our team learnt a great deal during this project. Some important learnings are:

- Development of a business case
- Pitching the business case to investors
- Sentiment analysis
- API integration
- Working with REST APIs
- JSON parsing
- Web scraping
- Time and resource management

References


Wikipedia. Massive open online course - Wikipedia, the free encyclopedia. n.d. 11 May 2014. 
<http://en.wikipedia.org/wiki/Massive_open_online_course>.