The University of Hong Kong  
Department of Computer Science  

COMP4801 Final Year Project  
CAES9524 Technical English for Computer Science  

A Web Application for CS course selection  

Final Report  

Supervisor  
Dr. Yu, Tao  

Member  
To Wing Kwong  
3035692072  

Date: 4 April 2023
Abstract

Many students will check and evaluate courses before enrollment. There are many resources available online regarding the comments of courses. However, existing resources are not reliable and comprehensive. This project aims to build a web application that provides reliable, concise, comprehensive and up-to-date information for students to select courses. The web application will be based on Python Django framework. API will be provided for existing official course website or trusted third-party websites to retrieve student feedback, summary and course instructor responds. Aspect Based Sentiment Analysis (ABSA) is deployed on the web application. The ABSA can effectively extract student comments on different components of a course and predict the polarity of the comments. A summary table can be generated after analyzing all student feedback. It is believed that the summary table provides concise, comprehensive information for students to evaluate the course.
Acknowledgment

I would like to thank my supervisor, Dr. Yu Tao, for the great support and helpful advice. In particular, the idea of using Sentiment Analysis is very worth investigating.
Contents

Abstract ......................................................................................................................... I
Acknowledgment ......................................................................................................... II
List of Figures ............................................................................................................... IV
List of Tables ................................................................................................................ V
Abbreviation ................................................................................................................ VI

1. Introduction .............................................................................................................. 1
   1.1 Background and Motivation ................................................................. 1
   1.2 Objective ............................................................................................... 2
   1.3 Report Outline ....................................................................................... 3

2. Methodology ............................................................................................................. 4
   2.1 Web Application Overview ................................................................. 4
   2.2 Language and Framework ..................................................................... 5
   2.3 Aspects Based Sentiment Analysis ................................................... 6
   2.4 Database .............................................................................................. 7
   2.5 Authentication ..................................................................................... 7
   2.6 Management Site .................................................................................. 9
       2.6.1 Common User Interface (UI) ...................................................... 9
       2.6.2 Home Page View ................................................................. 10
       2.6.3 Course Page View .............................................................. 11
       2.6.4 Class Page View ................................................................. 12
       2.6.5 Feedback View ....................................................................... 13
   2.7 Course Forum ......................................................................................... 15
       2.7.1 Home Page View ................................................................. 15
       2.7.2 Course Page View .............................................................. 15
   2.8 API .......................................................................................................... 16
       2.8.1 API Authentication .................................................................. 17
       2.8.2 API and Query String .......................................................... 18

3. Results ..................................................................................................................... 19
   3.1 Deployment .............................................................................................. 19
   3.2 Limitations ............................................................................................... 20
       3.2.1 Lack of mobile support .......................................................... 20
       3.2.2 ABSA table ............................................................................ 20
       3.2.3 Secure HTTP .......................................................................... 20

4. Conclusion ............................................................................................................... 21
References ...................................................................................................................... 22
Appendices .................................................................................................................... 23
List of Figures

Figure 1.1  Course Effective Profile in HKU Portal  2
Figure 2.1  Structure of CS course selection system  4
Figure 2.2  Example of ABSA on course feedback  6
Figure 2.3  Database schema  7
Figure 2.4a  Login page for Course Forum  8
Figure 2.4b  Redirect page for Google Oauth  8
Figure 2.4c  Google Oauth page and example of rejected access  9
Figure 2.5  Common layout of Management Site  9
Figure 2.6a  Home page of Management Site  10
Figure 2.6b  Home page of Management Site (with searching)  11
Figure 2.7  Course page of Management Site  11
Figure 2.8  Class page of Management Site  12
Figure 2.9a  Student Feedback view of Management Site  13
Figure 2.9b  ABSA view of Management Site  14
Figure 2.10  Home Page of Course Forum  15
Figure 2.11a  Course Page of Course Forum (staff or teachers)  16
Figure 2.11b  Course Page of Course Forum (students)  16
Figure 2.12  Django admin page for manage token  17
Figure 2.13a  Example of HTTP request to API  18
Figure 2.13b  Example of HTTP response from API  18
Figure 3.1  Client-server pattern  19
List of Tables

Table 2.1  Example of ABSA summary table  6
Table 2.2  API Endpoints  18
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect Based Sentiment Analysis</td>
<td>ABSA</td>
</tr>
<tr>
<td>Application Programming Interface</td>
<td>API</td>
</tr>
<tr>
<td>Natural Language Processing</td>
<td>NLP</td>
</tr>
<tr>
<td>User Interface</td>
<td>UI</td>
</tr>
</tbody>
</table>
1. Introduction

Many students will check and evaluate the course before enrollment. Although there are online resources that helps students to do so, those resources are not comprehensive enough. This project will introduce a new web application that provides comprehensive information for students selecting courses.

1.1 Background and Motivation

Computer Science (CS) department offers various free electives in each year, this provides wide range of choices for students. Meanwhile, students’ time and credit are limited. Information is needed for students to utilize their time and credit wisely.

Past student feedback is useful when evaluating the course. Indeed, some online forums provide this function, for example, goop [1] (a mobile application), richku.com [2]. However, those comments are not verified. It cannot guarantee that the student has studied the course before. The reliability is low. Additionally, those comments may be outdated, improvement might have been made. There is no effective channel to show the specific improvement of a course made by course instructor.

HKU gathers student feedback through SFTL. Unfortunately, the detail feedback is only visible to teaching staff. Students can only access the Course Effective Profile (see figure 1.1). The Course Effective Profile provides a Mean Course Effectiveness score for each course. Although the score is real statistics from past students, the score is not comprehensive. It cannot show the performance of different aspects of a course.
One way to evaluate different aspects of a course is to read through all past student feedback. However, it is time consuming for students to apply this to every course. If an auto generated summary can be provided, then students can scan summaries and read interested courses past feedback in details. Notably, Natural Language Processing (NLP) can process natural texts. Some NLP techniques, for example, Sentiment Analysis and Text Summarization can analyze or summarize texts. With the help of NLP techniques, it is possible to generate summaries automatically. A platform that provides concise, comprehensive, up to date information will become a strong support for students selecting courses.

1.2 Objective

This project aims to build a web application that existing official course website or trusted third-party websites can retrieve student feedback, summary and course instructor respond via Application Programming Interface (API). The project also aims to build a simple course forum that students can view collected data and instructor can respond to student feedback.

---

1 Screenshot from HKU Portal (login required)
1.3 Report Outline

The remainder of this report proceeds as follows. Chapter 2 discusses on methodology of the project. In particular, the choice of libraries, web frameworks and the details implementation of the web application. Chapter 3 states the result of the project and limitations of the application. Chapter 4 comments on the findings and makes recommendations for further studies.
2. Methodology

This chapter discusses the libraries and tools for ABSA, the framework for the web application and the details implementation of the web application.

2.1 Web Application Overview

Figure 2.1 shows the structure of the web application. Student feedback from SFTL will be uploaded to the application by admin after obtaining student consent. Student feedback will be extracted. In addition, NLP techniques, in particular, Aspect Based Sentiment Analysis (ABSA) will be applied to student feedback to obtain effectiveness of different aspects of a course. A summary will be generated after the system analyzed student feedback. Instructors can respond to student feedback via course forum. All above data will be stored in the database and can be retrieved by using API.

![Figure 2.1 Structure of CS course selection system](image-url)
2.2 Language and Framework

ABSA is a computation intensive job and only be used at the end of each semester. It is better to separate the ABSA and the user faced web application. However, due to budget concern and complexity, ABSA is integrated into the web application. Python is preferred for the backend development language. This maintains the consistency between ABSA module and the web application. The ABSA module can simply be kept in the web server and can be called by HTTP request directly.

Regarding the web framework FastAPI and Django are popular web framework in Python. FastAPI has the highest performance among three, but the framework is relatively new and contain less packages. While Django provides many commonly used functions out of the box and have a huge community. For example, it is relatively easy to implement user authentication on Django. The built-in admin panel also provides simple and intuitive user interface (UI) for non-IT specialist to manage the web application. Moreover, Django can handle frontend matters for us, we do not need to use a frontend framework. Therefore Django is preferred.
2.3 Aspects Based Sentiment Analysis

ABSA is one of the key features of this project. ABSA involves two NLP tasks, aspect term extraction and sentiment analysis. Aspect term extraction aims to identify the key aspects of sentences. While sentiment analysis aims to determine the polarity regarding the identified aspects. As shown in figure 2.2, ABSA will identify key terms (course, TA, assignment) and classify each key term as Neutral comment, Positive comment, or Negative comment.

![Example of ABSA on course feedback](image)

A summary table (see table 2.1) can be generated if ABSA is applied on every student feedback. This provides an overview for students of the rating of different course components, in particular, the count of positive, neutral and negative feedback.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TA</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Assignment</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Table 2.1 Example of ABSA summary table

As for the implementation, due to time concern, existing library will be used. In particular, PyABSA [3], an open-source framework will be used. This framework encapsulates famous, widely used deep learning framework, for example, pyTorch and Transformer. This framework provides pre-trained models that trained by
different datasets. Moreover, it is easy to deploy to this project with simple code.

2.4 Database

Django adopts Object Relational Mapping (ORM) design. This helps reduce the chances of SQL injection [4]. Also, this makes switching databases easier. Therefore, SQLite is preferred as development used as it is light and easy to deploy. Database will switch to PostgreSQL as it is more suitable for actual deployment.

Figure 2.3 shows the database schema of the application (Not include User Model). The application will process student feedback in Batch, each student feedback (Comment) and Batch and the response form instructor are related to a Class. A Class teaches one Course.

![Figure 2.3 Database schema](image)

2.5 Authentication

The web application consists of 3 parts, Management Site, Course Forum and API. Regarding Management Site and Course Forum, user account will be used for authentication. In particular, there are 2 permission classes, staff and normal user.
Management Site requires staff permission while Forum requires normal user permission only.

The web application does not provide any account registration service. Staff account should be created via Django built-in admin site manually. As for normal account, since all HKU members have a “@connect.hku.hk” email account. The Forum will employ OAuth authentication (see figure 2.4). OAuth authentication will reject email account other than “@connect.hku.hk” email account (see figure 2.Xc). With the help of OAuth, the Forum can verify HKU member identity without managing accounts for every HKU member.

![Login page for Course Forum](image)

**Figure 2.4a Login page for Course Forum**

**Sign In Via Google**

You are about to sign in using a third party account from Google.

![Continue](image)

**Figure 2.4b Redirect page for Google OAuth**
As for the authentication of API, details can be found in chapter 2.8.

2.6 Management Site

The web application will provide a management site for admin to upload, view and manage student feedback from different classes.

2.6.1 Common User Interface (UI)

Figure 2.5 shows the common layout of web pages of Management Site. The layout consists of 3 parts, navigation bar on top, data table on left and HTML form on right.
The navigation bar provides some frequently used functions, shortcut to home page, shortcut to Django built-in admin site, shortcut to Forum, logout function and the search box that will be discussed in next subchapter. Moreover, a breadcrumb navigation is provided. This shows the current location of the user and provides shortcut for user back to previous page.

The data table will display the key data of particular webpage. Pagination is used in server side.

The HTML form is for upload an instance of the key data of particular webpage.

2.6.2 Home Page View

Figure 2.6a shows the home page of Management Site. The key data of this page is “Course”. User can register a new course via the HTML form. User can access “Class” of specific “Course” by clicking details button. Apart from searching course page by page, the search box on the navigation bar can search for courses by the course code. An example is shown on figure 2.6b.

Figure 2.6a Home page of Management Site
2.6.3 Course Page View

The course page will show all “Class” that teach that “Course” (see figure 2.7). The key data of this page is “Class”. User can register a new class via the HTML form. User can access “Batch” of specific “Class” by clicking details button.
2.6.4 Class Page View

The class page will show all “Batch” that belongs to that “Class” (see figure 2.8). The key data of this page is “Batch”. User can upload student feedback via the HTML form as a “Batch”. User can view the raw feedback by clicking “view” button and submit a batch of feedback by clicking “submit” button. If the ABSA result is ready, user can view the ABSA result by clicking “result” button.

![Class page of Management Site](image)

Figure 2.8 Class page of Management Site
2.6.5 Feedback View

Figure 2.9a shows the raw view of student feedback of selected “Batch”. While figure 2.9b shows the ABSA result student feedback. Identified aspects are bounded by color box. Intuitively, positive aspects are green, negative aspects are red and neutral aspects are yellow.

Figure 2.9a Student Feedback view of Management Site
Figure 2.9b ABSA view of Management Site
2.7 Course Forum

The web application provides a forum for students to view the past feedback of courses. In addition, Instructors can respond to student feedback.

2.7.1 Home Page View

Figure 2.10a shows the home page of Course Forum. User can access “Course” by clicking details button. Apart from searching course page by page, the search box on the navigation bar can search for course by the course code.

![Figure 2.10a Home Page of Course Forum](image)

2.7.2 Course Page View

Figure 2.11 shows a example course page of Forum. Left hand side consists of instructors’ response and past student feedback. Right hand side consists of ABSA table (discussed in chapter 2.3) and the description of the course. All responses and feedback are sorted by latest. Figure 2.11a is the view of staff or teachers of the course, while figure 2.11b is the view of student. Only staff or teachers of the course can submit “Response”.

![Figure 2.11 Course Page View](image)
This project will provide API using Django Rest Framework. Django Rest Framework provides packages to handle authentication and permissions with simple code. API provides a convenient way for existing official course website or trusted third-party website to retrieve students’ feedback, ABSA summary and course instructor responds.
2.8.1 API Authentication

As those data should be used by official course website or trusted third-party website only. Authentication should be implemented to protect the data. Django Rest Framework provides different authentication methods, for instance, Token Authentication and Session Authentication.

Token Authentication is a simple solution to client-server model [5]. Official course website or trusted third-party website should include a token when they retrieve data from our API, otherwise, it will receive HTTP 401 Unauthorized. In particular, the token should place in the Authorization HTTP header as following:

```
Authorization: Token 9944b09199c62bcf9418ad846dd0e4bbdfc6ee4b
```

To avoid man-in-the-middle attack, all parties should access the API through HTTPS only.

Session Authentication makes use of Django's default session backend, it is similar to authentication of user account. It is suitable for clients that are running in the same session as our website. However, our API is provided for third-party which is not using our website directly like a normal user. Token Authentication is preferred.

Token can be created and managed through Django admin page (see Figure 2.X).

![Django admin page for manage token](image)

Figure 2.12 Django admin page for manage token
2.8.2 API and Query String

The API endpoints are shown in table 2.2. Data can be retrieved by HTTP GET request. The HTTP query string will take course, start year and end year as input. Student feedback or ABSA table of selected course and within selected time will be returned in JSON format.

<table>
<thead>
<tr>
<th>Data</th>
<th>HTTP</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student feedback</td>
<td>GET</td>
<td>/api/comments?course={course_code}&amp;from={year}&amp;to={year}</td>
</tr>
<tr>
<td>ABSA table</td>
<td>GET</td>
<td>/api/absa-table?course={course_code}&amp;from={year}&amp;to={year}</td>
</tr>
<tr>
<td>Instructor responds</td>
<td>GET</td>
<td>/api/responses</td>
</tr>
</tbody>
</table>

Table 2.2 API endpoints

Figure 2.13a shows an example of retrieve student feedback of “Course TEST1001” using PowerShell Invoke-RestMethod. Figure 2.Xb shows the retrieved data.

Figure 2.13a An example of HTTP request to API

![Figure 2.13a An example of HTTP request to API](image)

Figure 2.13b An example of HTTP response from API

```
$headers = @{
    'Authorization' = 'Token a4d28c1b5442a22a7f782aff8883871799ce90c8'
}
Invoke-RestMethod -Uri $uri -Method Get -Headers $headers
```
3. Results

This chapter will state the result of the project, in particular, the deployment of the project and the limitation of the project.

3.1 Deployment

To demonstrate this project, I decided to set up a web server on cloud virtual machine. In particular, Amazon EC2 is chosen. EC2 is priced based on usage. EC2 provides relatively low baseline performance and relatively low pricing compared to other cloud products. Although the baseline performance of EC2 is low, its unlimited mode can provide burstable performance when CPU loading is high [6]. This feature is very suitable for our application which requires high CPU power when running machine learning module but low utilization in normal operation. The machine learning model consumes about 2.5GB ram. Therefore a t2.medium (2 vCPU, 4GB Ram) instance is selected for this project.

Nginx is used for web server. However, Nginx cannot execute Django application directly, gunicorn is used as the middle WSGI server (see figure 3.1).

![Client-server pattern](image)

Figure 3.1 Client-server pattern

Address and the demo account of the web application can be found in Appendix I.
3.2 Limitations

3.2.1 Mobile support

The web views of this project are designed for desktop view only. The web pages are very small on a mobile device which is not user-friendly. Considering the trend that many people surfing the web using mobile phones. A mobile friendly design should be implemented in the future.

3.2.2 ABSA table

ABSA table is generated using data in Aspect model. There are some limitations. First, the ABSA process will identify many aspects if ABSA is applied on real comments. This violates the objective that provides concise summary for students. To keep the ABSA table concise, ABSA table will only consider frequently appear aspects. Second, the ABSA model is not specifically for course review. The model is trained by review in different context.

3.2.3 Secure HTTP

The deployment of the web application is limited to HTTP (port 80) only. To protect user information, HTTPS (port 443) connection should be used in real production. However, setting up a HTTPS connection requires a real domain, which incur extra cost. This project will stick to HTTP as HTTP is enough for demonstration purpose.
4. Conclusion

Information is needed for students to utilize their time and credit wisely. However, existing resources are not reliable and comprehensive. This project proposes a web application that aims to provide useful information for student enrollment. This web application will gather, analysis and summarize past student feedback from SFTL. In addition, the web application will allow course instructors to update the course information. It is believed that these concise, comprehensive, up to date information can help students to select the most suitable course.

The preliminary version of the web application is completed. The basic function of the web application has completed and tested. However, the web application still not perfect for real production. Many small details may be needed for real production environment.

Due to time concern, we are not planning to further improve or enhance the ABSA model. We noticed that the performance can be enhanced if a more suitable and larger dataset can be used for training the model. However, producing ABSA dataset involves subjective judgements. An effective way to produce ABSA dataset or unsupervised machine learning version of ABSA is worth investigating. This will benefit all applications that analyze reviews or feedbacks using NLP.
References


Appendices

I. Web application demo

(The web application demo will close on 7 May)

Management Site Address

http://ec2-13-230-203-0.ap-northeast-1.compute.amazonaws.com/adminboard/

Demo account

<table>
<thead>
<tr>
<th>Username</th>
<th>admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>admin</td>
</tr>
</tbody>
</table>

Course Forum Address

http://ec2-13-230-203-0.ap-northeast-1.compute.amazonaws.com/forum/

Student feedback (API)

http://ec2-13-230-203-0.ap-northeast-1.compute.amazonaws.com/api/comments

ABSA (API)


Instructor responds (API)

http://ec2-13-230-203-0.ap-northeast-1.compute.amazonaws.com/api/responses