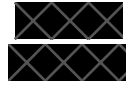


Final Report



Abstract

The project AWSW (Academic Writing Suggestion Website) is aimed at building an academic writing collection/suggestion website for users who need academic writing suggestions. The website system uses the off-the-shelf NLP technique to collect sentences and expressions from published proceeding or journal papers and theses as a continuously updated database, and a state-of-the-art sentiment analysis machine learning model to give users suggestions. It is a platform that provides an interface to users and collects data from users.

AWSW will simplify their research process by reducing the workload on checking the technical diction and expressions.

The team has achieved most of the basic targets, listed as follows:

- The machine-learning algorithm has reached a high accuracy.
- The establishing of a corpus-based on published academic papers.
- Analysing input academic sentences and giving suggestions accordingly.
- Association function.
- Customized library function.

1. Introduction

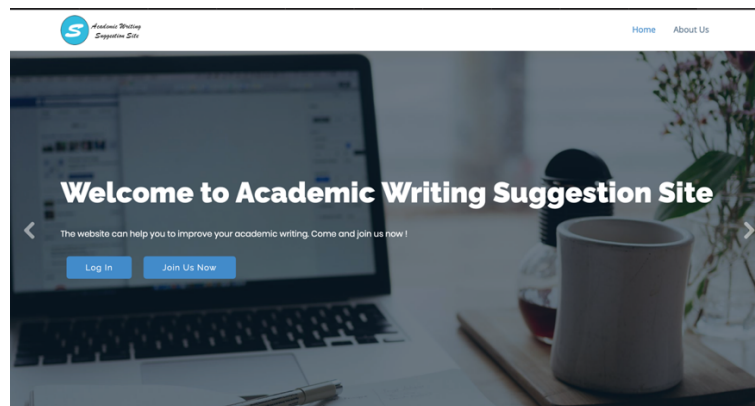
In general, AWSW is to build a website for users to correct their academic writing mistakes, based on professional suggestions from our database.

There is a niche market and an increasing demand in the academic writing suggestion and correction area, because writing is of critical importance to the academic world. It is not only the main method for scientists to communicate, but also a significant means for researchers to show their discoveries to the world (Ventola & Mauranten 1996). And unlike oral or other kinds of written languages, academic writing is a formal style of writing that has its own pattern. Thus, for the junior researchers and those who just involve in the academic area, it can be a huge challenge to start academic writing, even for native speakers. Writing academic papers, dissertation and research papers in a foreign language is even more challenging, yet there are tens of thousands of people facing the increasing pressure to write and publish in English (Tang 2012).

Comparing to the existing writing suggestion systems such as the most widely used one - Grammarly, ASWS has huge advantages. When Grammarly can only provide grammar suggestions, ASWS offers more specific suggestions in academic writing. Also, ASWS advises on whole sentences instead of only word corrections. And ASWS offers multi-suggestions while Grammarly only gives single solution. Thus, ASWS is a product more convenient for academic staffs. Have a look at the interface of ASWS website (as shown in Figure 1), you will be attracted by the simple and clear design.

ASWS is aimed at establishing an academic writing suggestion website for users such as students, researchers, and other academic staffs struggling in academic writing.

Figure 1. the interface of ASWS website

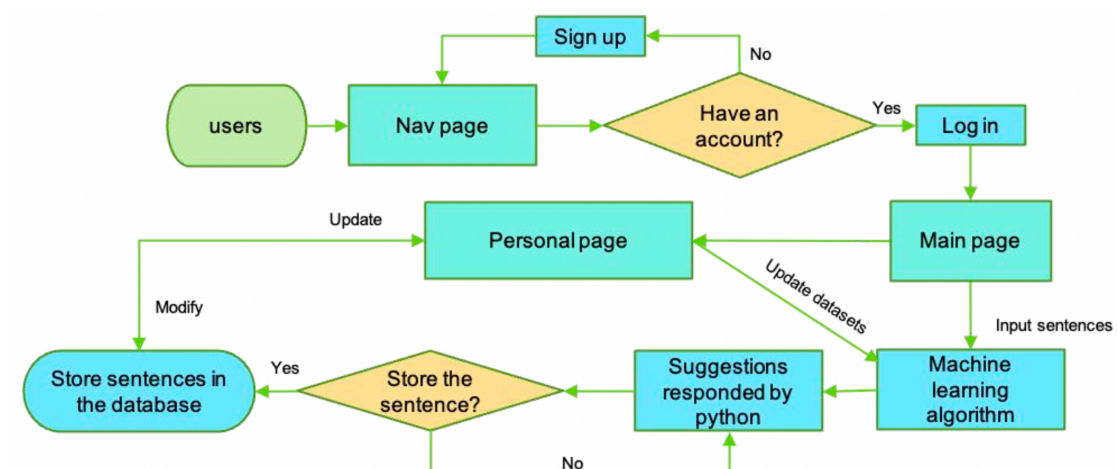


In Section 2, I will describe what our project aimed to achieve. Section 3 will be details about the developing process and the approaches I used. In Section 4, I will discuss the achievement and the skills I have gained. Section 5 will be a summary of the project and the future work that could be done.

2. Project Aims

Figure 2 shows the main aiming functions of ASWS. Our program starts with a navigation page (Figure 1). If users already have an account, they can log into the main page. If not, users need to sign up first.

Figure 2. Flowchart of ASWS



Then, users can input a query sentence on the main page. The python interface will give five most popular suggestions to users. Users can save the results in their customized database, copy them to the clipboard or ask the page to show more samples from the same classification if not satisfied.

Figure 3. The result of the machine learning algorithm

input your sentences

Hello
Home
Log Out
Collection

Run python python_interface.py. Install all the necessary package by using pip until it can run.

Submit

#	classification	Sentences	Actions
1			+ 📄 🗑️
2			+ 📄 🗑️
3			+ 📄 🗑️
4			+ 📄 🗑️
5			+ 📄 🗑️

Users can check their saved results on their personal pages, also they can upload a dataset to enlarge the database of the machine learning algorithm.

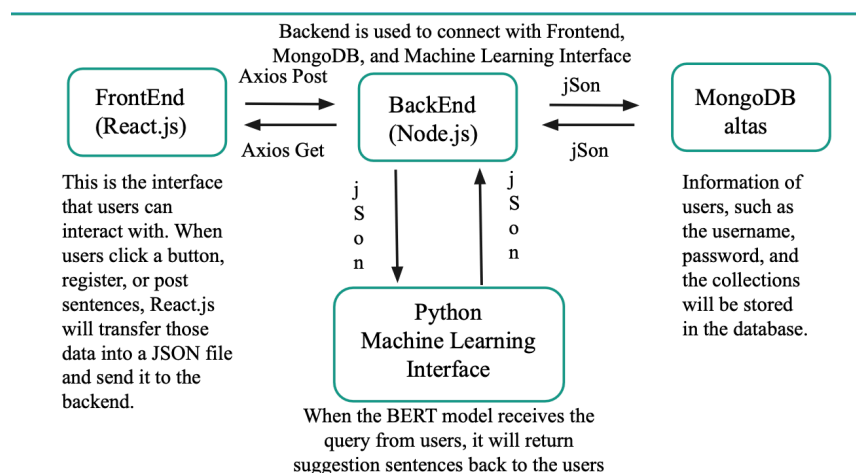
3. Approach

There are three main parts to the project. The software used in each part are shown in Table 1, and the system structure is shown in Figure 4.

Frontend	Backend	Machine Learning	Others
React.js, HTML, CSS, Javascript	express framework, Node.js, mongoose, MongoDB atlas	Python, Pytorch, Transformers	GitHub API

Table 1. Used tool & software

Figure 4. system structure



Front-end

For the Milestone one, we only want to achieve the basic functions and want our website to be simple, so we use HTML, CSS and JavaScript to create our website in the first place. With the HTML we can create the framework of our website. With the CSS we can style our website and with JavaScript we can achieve some functions and give the data from frontend to the backend. These three tools are the fundamental tools of developing website.

For the Milestone two, after achieving the basic functions, our customer suggested us to beautify our website and achieve some more complicated functions. So, we decided to achieve this by using Bootstrap and React. Bootstrap is a free and open-source CSS Framework, and with the Bootstrap we can use many beautiful sample structures to style our website instead of

just using CSS, which can make our work much easier. React is a JavaScript library which allows us to create reusable UI components and makes our coding fast, scalable, and simple.

Back-end

After creating our frontend, we need to build server and connect it to the database. To achieve this, we used Node.js and Express.js as our developing tools. By using these tools, we can set up the backend environment and connect the frontend to the backend.

For the database, we use MongoDB atlas, which is a database implemented on an online server. With the use of MongoDB, we can connect the database to the local backend easily.

Machine Learning Algorithm

Machine Learning Algorithm is the core algorithm in our project. It can analyse the user's sentence and return suggestion sentences based on the similarity between this sentence and all types of sentences in the database. In this project we use Bert Library as our Machine Learning model. With Bert we can train our algorithm with high accuracy.

4. Results

As listed in the abstract, we have achieved most of our initial goals. I will discuss the results from the following three aspects.

Group achievement

Our project has completed most of the requirements that proposed by the client. we have completed the following results as a team: 1) The machine learning algorithm can reach 92% of accuracy after training in 20 epochs. 2) The machine learning model can analyze the user's sentence and return five different types of sentences based on the similarity between this sentence and all types of sentence in the database. 3) Users can customize their library by adding or deleting their database and provide feedback to improve the algorithm. 4) Users can check more sentences from one specific class easily by just clicking one button.

Individual achievement

We have allocated our work into three parts, which are Front-end developer, Back-end developer and Website tester. Front-end developer is responsible for the interface of the website. Back-end developer is responsible for writing the web services and APIs used by front-end developers. Website tester is responsible for reviewing and testing the website.

In this project, I am mainly in charge of frontend and test part.

For the frontend my individual achievements are as follows:

- Design and develop the main function page.
- Design and develop the users' sign-in and log-in function.
- Using bootstrap table to beautify our website interface.
- Add an About page to introduce our website.
- Help connect the backend to the front end.

Specific case:

The design and develop the main page.

At the beginning I am a total stranger in web design, so I watched a lot of videos on how to create a website using HTML, CSS and JavaScript. After discussing the main structure of our website with my group members, I began my first webpage design.

At first, I use HTML and CSS trying to build the main page. The main page has three parts: input part, suggestion part and menu part. It is not hard to put all these parts on the page. The

hard part is layout. For example, when I narrowed down the page using the mouse the layout changed a lot and was not as I expected. To solve this problem, I searched stack overflow to get the answer. The solution is to change a parameter value which can make the layout fixed. The first version of my main page design is not very beautiful. In order to improve my website's interface. I watched some videos to learn how to use Bootstrap. By using the Bootstrap, my page looked more attractive.

Test part

For the test part my achievements are as follows:

- Finish the frontend tests and fix the problems.
- Finish Integration test of frontend and backend and fix problems.
- Finish the system tests.

During the test part, I came across a lot of troubles.

For example, During the integration test, I faced an issue of asynchronous problems between frontend and backend. When I sign in with new username and password, the user's data should show in the backend. But it didn't show in the backend side. It turned out the port was used by another process, which lead to this problem. After changing the port, the problem was solved. Another problem is that after we create a new account in the sign-in page, I can't log in directly because the page stayed in sign-in page. To solve this problem, I redirect the page to the login page as long as I sign in.

5. Conclusion

In conclusion, our project has achieved most of the basic requirements. Users can get professional suggestions from machine learning algorithms and they can also collect sentences and improve the database by uploading useful datasets. Details have been demonstrated in both Section 2 and 4.

Limited by resources, we do not have the sufficient test users to try our website. So, we still need more users to test and verify our program. Also, we have no conclusion on the current capacity. If there are a huge number of users using our website at the same time, we may need to improve our capacity.

However, our project is not finished yet. The following tips can be the extended path in the future work.

- Migrate the website from localhost to a cloud server.
- Improve the efficiency of the machine learning algorithm and shorten the response time. To reduce the response time of algorithm we may need to load the python model in advance and deploy the model on a cloud server.
- Add new examples and classifications to the database to improve the accuracy of the machine learning algorithm.
- Ensure users can upload PDF files to update the database and export their favourite sentences as PDF format.

References

Tang, R 2012, *Academic writing in a second or foreign language issues and challenges facing ESL/EFL academic writers in higher education contexts*, Continuum International Pub, London New York, NY.

Ventola, E & Mauranen, A 1996, *Academic Writing: Intercultural and textual issues*, John Benjamins Publishing Company, Amsterdam.