Why is This Sensitive? Visualizing Important Sensitivity Classification Features

Gangxin Li

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

10 December 2021

Abstract

This core goal of this project is to visualize important sensitivity classification features. First, it obtains the content of the article by parsing the HTML file. It is divided into two categories by various classifiers to detect whether an article is sensitive or insensitive, and finally the analysis results are presented to the user through visual means.

As an auxiliary technology for sensitivity review, it must have a certain basis to prove that it is sensitive or non-sensitive. Therefore, the focus of the project is to visualize those sensitive words and their corresponding weights, so that reviewers can quickly see through this article. Through the realization of a large number of visualization functions and charts to corroborate the results of the article. Reviewers can index corresponding articles through certain conditions and view the internal data of the article.

At the same time, the application also provides a statistical analysis of the entire data set and a model selection process. Finally, user study is used to verify the feasibility of the project and the corresponding evaluation. According to the research results of 12 participants, 76% of users expressed their willingness to accept this app as an auxiliary review. At the same time, the project uses pre-training technology, and there is almost no lag in project operation. User experiments have proved the feasibility and effectiveness of the system, and the design has been implemented as required.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

<Please note that you are under no obligation to sign this declaration, but doing so would help future students.>

Name: Signature:

Acknowledgements

I want to thank my supervisor, Dr, Graham McDonald who not only provided me with the much-needed groundwork to kick start the project but also in the process of project construction, he always encourages me not to give up or help me analyze the realization of the function together. At the same time, it also pointed out the direction of my project and guided me on how to realize it all the time. Without his help, I think my project will be very difficult.

Secondly, I want to thank my father. Without his funding, I might not have the opportunity to study in the UK at all. This is of great significance to my life.

Contents

Chapter 1 Introduction

There had more than 100 countries [1] around the world devote at freedom of information laws and acts. Through the enactment of freedom of Information laws, it has become a mainstream form to ensure that the government can provide the public with the relevant basis for decision-making information openly and transparently, e.g. United Kingdom's (UK) Freedom of Information Act (FOIA) [1]. Hence, sensitivity analysis and inspection are still required for public documents or information to ensure that the published documents do not contain sensitive information.

However, With the development of the information age and lots of meetings, a large number of digital files are produced, so there will be a shortage of manpower, so the auxiliary review has been widely recognized. For technical reviewers, it is a challenging task to be able to visually see the sensitivity level and sensitive words of the article before the article is published.

The aim of the project is providing a visual solution to make it possible to highlight sensitive words. It could provide reference for reviewers by visualizing sensitive words and highlighting the weight of sensitive words. The main goal of this project is to enable users to use various features of the search document, such as important entities, time, author, etc., and to allow users to have a quick analysis and understanding of their sensitivity through a visual method. In this case, the user can make a judgment on the corresponding document by searching for different articles or keywords, and decide whether the article can be submitted for archiving.

1.1 Outline

The chapters are organized as follows: Chapter 2 offers the background and related work in the areas of sensitive and visualizations and the aiming of the project. Chapter 3 provides the architecture diagram and the discussion of high-level architecture design decisions. Chapter 4 and 5 provide detailed design and implementation of the project, including concrete implementation methods and examples of documented code. Chapter 6 discuss the results of the model and the effect of the model. Chapter 7 and 8 show some additional analysis and the main findings or conclusions.

Chapter 2 Survey

The focus of this project has two parts. The first is to conduct sensitivity analysis, which can extract effective keywords from the article and infer whether the article is sensitive. The second is the visual content, which includes providing users with an interactive interface and the results of graphical data, including highlighting keywords and graphically related data operations. At the same time, a certain explanation of the visualized content is required. Finally, the usability of the overall software is tested by means of personnel testing, etc.

2.1 Sensitive analysis

As for the sensitive analysis, McDonald et al [2,3] already did lots of works on the Digital Sensitive Review in order to specific the importance and necessity for efficient sorting of documents. They use SVMLight [4] with a linear kernel which be measured by several classification measures named: Precision, Recall, F-measure, as well as Balanced Accuracy (BAC). And considering the SVM's sensitivities to the imbalanced training data, they up-sample the training sets in each fold until the number of sensitive and non-sensitive dataset balance. They found that interest exchanges and possible conflicts between countries are the essence of the relationship between the two countries [2]. Furthermore, they promote an evaluation of the effectiveness of semantic word embedding features, and with term and grammatical features in order to make progress in sensitivity classification [3]. And they also using knowledge graph such as WikiData, DBpedia etc., which by using the various entities and relationships to gain insights and enrich the data.

In the meantime, Shota Okumura et al. promote a new sensitivity analysis framework [6], it could add or remove some of instances without any problems. And it could refresh the framework frequently which solve those problems with large data overhead and small incremental data. In some of sensitive analysis tasks, the framework shows great benefits, where only a small number of instances are updated.

2.2 Visualization

As for the second part of the project, Visualization is an important display direction of this project. It presents by create a visual platform to show whether an article is sensitive. Marco et al [5] propose a new technology named LIME, which one could explain the predications of any classifier by learning an interpretable model locally. It can choose between different models, make trust assessments, improve those models that are not trustworthy, and show the source of the predicted results. Compared with the direct development of new packages, this greatly improves the development efficiency and also has better stability. And T.J.Jankun-Kelly and Kwan-Liu Ma [10] also promote a spreadsheet interface for visualization exploration. It could let spreadsheet more interactive and beautiful. In the meantime, Rob Lintern et al [11] also gives me the inspiration about how to do the suitable plugging-in visualization. They conduct open collaboration and visual development on the Eclipse platform, and they add their own visualization tool, SHriMP Views, to this platform.

As the field evolves, information visualization has reached the field of linguistics, Harri Siirtola et al. explored text visualization, and it is popular both as an object of research direction and among the consumers of the visualizations [13]. It can abstract text into flowcharts or text-centric images, make it easier for people to understand the relationship or story outline in the article. Furthermore, Yedendra et al. present a new information visualization framework that supports the analytical reasoning process [14]. They use extensive visual development support, and link the analysis artifacts to the visualizations in order to validate the findings. Also, Heli Väätäjä et al. found that the selected heuristics were useful with good coverage in practical heuristic evaluation [15]. This research provides a theoretical basis for discovering usability problems in information visualization systems. And based on the advances in graphics hardware, Bown et al. have a chance to highlight interoperation among disciplines at this arts-science-social science interface [16]. Definitely, it was the best of times and it was the age of wisdom [17], it seems everything needed us to explore. A large number of fields urgently need the support and expansion of information visualization, and large amounts of data can be colored. Even more, certain color hues could improve short-term memory in suitable information visualization [18]. Therefore, both academia and education should pay attention to the importance of information visualization, which not only improves the quality of life, but also lays the foundation for future development.

2.3 Evaluation

Michael Aupetit [12] discussed the pitfall of the scatterplot spatialization of data similarities. And let the reader to witness the result is not suitable.

After finish the visualization, Tanja Blascheck et al [9] promote an evaluation interactive visualization system based on the eye tracking, it allows a more comprehensive evaluation of visualization based on the results of eye tracking, not just time and user reviews.

Chapter 3 Requirements Analysis

This chapter first enumerates the required requirements of the application, including the front-end and back-end as well as the visualization part. Then there is the requirement analysis, which tells how the project was carried out, and some more detailed process. And how to use original data to create more visual content.

3.1 Requirements

The main task is visualizing the important features, key idea is designing an application could provide a way to sensitive reviews to check the articles whether is sensitive or not. And through a variety of chart analysis, to provide reviewers with a judgment support. For example, when reviewers want to view specific sensitive words in an acritical, the program will highlight which are sensitive words and which are non-sensitive words. At the same time, the reviewer may want to know how many articles were judged to be sensitive in this year or in this batch of data, and what are the core sensitive vocabulary.

Because the content to be realized and displayed is diverse, these functions and tables describe and analyze the content of government documents one by one, and provide a reference for reviewers. Hence, after learning the different existing tools and discussing with my supervisors, I came up with these requirements (most requirements are inspired by my adviser Graham McDonald). The required requirements are discussed below, and the order of requirements is sorted according to the Moscow method [9].

Must have:

- **Develop a user interface.** Build a visual framework to ensure that different pages can be updated according to different data.
- **Process and build every visual content.** Visualize the data to facilitate the visualized loading and construction.
- **Parse HTML code.** Decompress the source file and extract the content of the article in a structured manner.
- **Categorize the data.** Use different models to fit the data and generate a pipeline to process the content of the article.
- **Link the processed data with the interface.** Link the page and the back-end data so that the entire project can run normally.
- **Indexing the articles.** Provide users with an index method that enables users to switch between different articles when using the application.

Should have:

- **Verify that the HTML parsing process is correct.** Verify whether the content extracted from the article is correct, and whether the extracted quantity and tags are correct.
- **Model Selection.** Skilearn provides a large number of classifiers, through model selection to find the best model and parameters, can effectively improve the performance of classification.

- **Structured storage data.** The results of the data are stored in a structured manner to facilitate page indexing.
- **Webpage navigation.** Provide users with a page navigation and page jump function to facilitate users to view different functions.

Could have:

- **Application could support entity query search.** The user could enter the key point such as nuclear, missile, contract and so on, then the interface will return the articles with highlight sensitive words. Key point is extract entity word and summaries the sensitive and non-sensitive words and then combine it.
- **Optimized Platform for smooth usability.** This software should respond quickly to meet the needs of users, while being able to interact well with users.
- **Logging of user operations.** In the background of the software, it should be able to accept user operations, as well as retrieved content and time spent, and record these log files for future update and optimization.

Won't have:

- **Support many different programming languages.** The backend of this project is all built using Python.
- **Replace special characters to meet some segmentation requirements.**
- **Users can build or train models by themselves.**
- **Display the content of multiple articles at the same time.**

3.2 Requirement Analysis

This section is divided into two parts, one is the analysis of the original article, which contains information such as the specific structure and quantity of the article. The other is the composition of the overall project and the analysis of the functions that may be required.

3.2.1 Raw Data:

The raw data for the application is HTML documents form the government records. There are 3,801 official UK Government articles received from my advisors in order to complete this project. Each html file is compressed and all the contents are construct by the following:

- **Reference Id**: It is the identifier of a target article.
- **Created**: Contains date and time when create the article.
- **Release**: Contains date and time when release the article.
- **Classification**: The classification category of the article.
- **Origin**: Comes from the UK Embassy or Consulate.

Based on the above features, there are also have some significant features such as:

- **TAGS**: Words that have strong relevance to the article.
- **Subject**: The title of the article.
- **REFS**: Contains the target article's ID.
- **Author**: In the last line of the article.

In addition to the above tagged data, there is also a file named **full.collection.cables.path.gold** which contains the ground truth of the articles with name sorted.

3.2.2 Components

Based on user requirements and requirement analysis, the project is divided into two main parts, back-end and front-end. The back-end component should be able to process and extract the main features and relationships between documents, and at the same time can efficiently save the processed content for easy indexing. For the front end, users can find the analysis results of the article through various buttons or search.

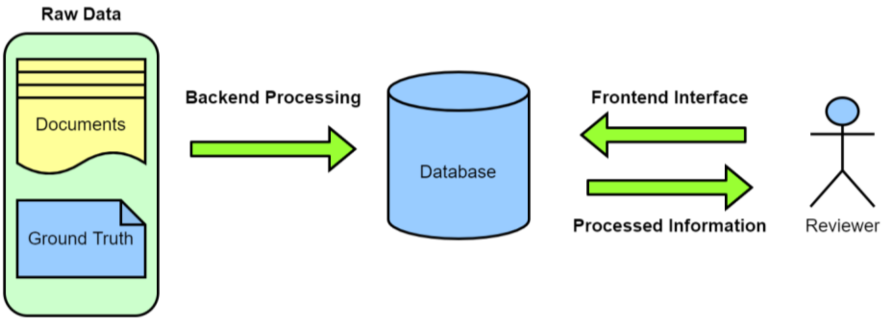


Figure 1: Components of the Application

R1: Provide an identification list for article search

Analysis: According to the year and month of the data, they are classified and stored in a structured manner. For the backend, the content of the index is fed back. For the frontend, when the user selects a specific article, the processed content is fed back to the page.

R2: Unambiguous and clear representation of information retrieved.

Analysis: The application program interface should present the results of all analyses in a structured and organized form. There should be clear instructions and reminders based on the displayed icons. The visualized image should be clear enough so that the user can freely zoom in and out to meet viewing needs.

R3: Optimize the platform in order to search efficiency

Analysis: Minimize the opportunities for online training and analyze all the articles directly. This can greatly reduce the query time and the user waiting time

R4: Facilitate migration and deployment

Analysis: In order to ensure the portability and universality of the application, the application should be made in the form of a website so that it can be deployed on the local network and is convenient for reviewers to use.

R5: Save user log files

Analysis: By saving the user's operation records and query records, it is convenient for further development and improvement of this application.

Chapter 4 Architecture

This chapter mainly introduces the main framework of this project, which includes the framework of the entire system, as well as the front-end and back-end framework details.

4.1 System Architecture

As mentioned in the Requirement Chapter, the Application was incremental gradually, for this project, I chose to use agile development methods [8], because the project has great flexibility and uncertainty, so by using agile programming methods to improve the efficiency of software development.

First, read the data structured, and then parse the corresponding article. The next step is to use natural language processing to identify entities and construct data bags. These data provide a basis for subsequent article display. Then use different models to fit the data and compare the performance of each model. Finally, choose the best model to process all the articles. Structured storage of processed articles is convenient for retrieval and use.

After finishing these back-end work, it is to design the user interface. The interface will be expressed through the use of web pages. A search method is provided on the user interface to select the corresponding article. The content displayed is usually an article. The content includes highlighting the sensitive and non-sensitive words in an article, the proportion of sensitive and non-sensitive words in the judgment of this article, and the emotional tendency of this article.

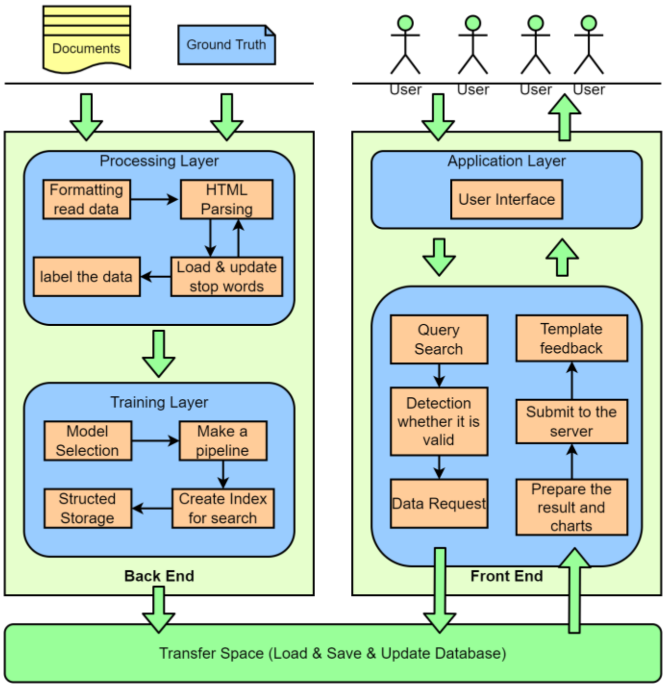


Figure 2: System Architecture

The figure 2 shows the while process, at first, the backend loads the documents and ground truth from the local disk, then using the processing layer to formatting the data and parse the HTML, base on the features and entities, they could update the stop words in order the accuracy in the model selection. Then label the date prepared for the training. In the training layer, when done the model selection, choose a best model as pipeline, then train the dataset and save the result structured. Finally, it could load from the disk.

After the backend, in the frontend, the use could use the interface to interact with the dataset. The user using indexing to get target article, when pair the target article, it will generate a new HTML for the user, then it will show on the interface. And user could get the result.

For the backend, all data is static, including results. Therefore, the back-end operation is offline and only needs to be processed once. For the front-end page, it is a process of dynamically calling back-end data. The front-end includes all executable logic and interactions. The overall data transmission method is to use the back-end to analyze the data, and timely feedback to the user through the front-end.

4.2 Backend

The backend data processing operation has a set of independent processes, and the entire process is described in a formatted manner below.

Backend-1. Document Parsing: Firstly, uncompressing the files and save it structured. Then parsing each HTML which contains Reference ID, Created, Released, Classification, Origin, DECL, Tags, Subject, REF, main content and author. Read data in blocks for easy data analysis. Here is the relationship between (Only shows necessary).

Backend-2. Statistic analysis: Perform statistical analysis on the time, keywords, author, etc. in the article, and draw the corresponding distribution diagram and cloud diagram

Backend-3. Label/Split the data: Based on the ground truth, label the data to each file, and separate the sensitive training/test dataset and non-sensitive training/test dataset.

Backend-4 TF-IDF & update stop words: Here we set the stop words in order to get more accuracy in the future, and using TF-IDF vectorizer to transform the dataset.

Backend-5 Model Selection: Choose a set of estimators as variable, and try Extra Trees Classifier, K-Neighbors Classifier, AdaBoost Classifier, Random Forest Classifier, Gradient Boosting Classifier, Bagging Classifier, Decision Tree Classifier to fit the model, and choose the best one as application model. And then make a pipeline to fit the dataset.

Backend-6 structed result: Test all test data and store the results in a structured hard drive for easy indexing later.

4.3 Frontend

The front end includes specific components and interfaces, and its components are as follows.

Frontend-1 Visualization Interface: This interface is the main interface of the front-end. It includes the query interface of the page. It allows users to select filters to filter conditions. It also provides a direct search function based on file names. At the same time, it also highlights sensitive and non-sensitive information, as well as corresponding icons. Figure 3 shows the interface.

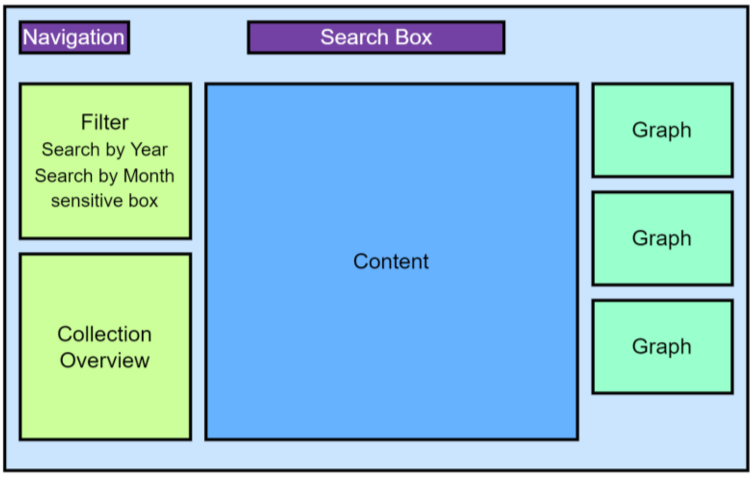


Figure 3: The interface of main page

Frontend-2 Visualization Implement: By calling the ready-made data file at the backend and formatting it, the information that needs to be expressed is extracted to the user.

First, the user enters a specific document name or uses filtering to index, then the server will search the index from the saved data, and then format and extract the HTML content to re-typeset to the user, and finally feedback the organized page to user. Figure 4 shows the whole process.

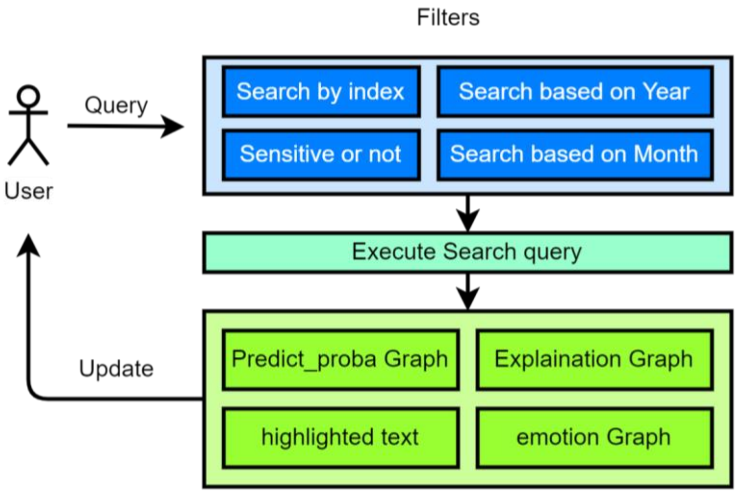


Figure 4: The query process

Frontend-3 Interface for Article Content: The content of the article is obtained by parsing HTML, and the data is transmitted to the page by obtaining the content of lime text_div. For the content of the article, orange is used to indicate sensitive words, and blue is used to indicate non-sensitive words. At the same time the shades of colors also indicate their degree of relevance. At the same time, for those highlighted vocabulary, a certain search feedback is given to the user, that is, when the user clicks on the vocabulary, it will automatically jump to the wiki page.

Frontend-4 Statistical analysis page: This page provides users with detailed statistical analysis, including basic file information, time distribution, author's cloud map, etc. It also provides training effects of various models, and shows one by one why this model was chosen.

Frontend-5 User portability operation: The application provides a certain welcome interface and page jump function, which is convenient for users to use and experience better.

Chapter 5 Implementation

In the content of this chapter, it introduces how to implement the project from two aspects. They are the front-end and the back-end respectively. For the back-end, it first introduces how to select the required tools and some of the algorithms used and the corresponding results. For the front-end, it also first explains what kind of technology is used, and how it is implemented for each page. At the same time, some more important page results are presented.

5.1 Backend

This section is divided into two parts. One is the details of the technical implementation and explains why such a technology is used. For the second section, it introduces a more important back-end code and some important results, including Model selection sections made during the training process, and how to process and generate tagged articles.

5.1.1 Technical specifications

In order to realize the back end of the project, there are a large number of tools available, but the following tools are selected as support for the project. For python anaconda distribution, it is a very good package management tool, it can perform version control very well, while also using some other tools, their advantages and disadvantages and the reasons for how to choose are explained in justification.

Tools and technology used are as below:

No.	Tool	Justification
1	Python Anaconda Distribution	Python is a good development tool, but here, anaconda distribution is more suitable because it has more convenient package management tools, and it also provides a large number of data analysis tools for use.
2	BeautifulSoup	Compared with extracting data directly through regular methods, BeautifulSoup can parse HTML files better, and simplifies the parsing process by directly reading HTML table, div and other fields.
3	Scikit-learn	Scikit-learn is a simple and efficient tool for predictive data analysis, it could accessible to everybody, and reusable in various contexts. Especially when it comes to model selection, it provides a large number of classifiers.
4	Seaborn	Seaborn is a Python data visualization library based on matplotlib, it could provide a high-level interface for drawing attractive and informative statistical graphics.
5	Lime	Lime could explain the sklearn model predictions, as for this application, it could show the weight between sensitive and non-sensitive, and highlight them.

6	Numpy/Pandas	Numpy and Pandas is prepared for data analysis, both of them could load or save data effective.

Table 1: Backend Technology

5.1.2 Backend Implementation Details

The following will explain in detail how each component is used, and the difficulties or frustrations encountered in the process.

HTML parse with BeautifulSoup: First, the read HTML file is parsed and identified by using a specific table. The article ID, time and other characteristics can be directly read. At the same time, the content must be regularized to process each piece of content separately. The relevant core code is as follows.

```
For each file in the whole html:
    if file path is not a legal path or not exist the file:
        continue
    try:
        Load the html file, using BeautifulSoup to parser, extra the article ID, time and so on. Then using regular expression to get content separately. The regular expression is “<[^<]+?> ”, definitely, need to cut the original html language.
    except:
        give the programmer a warning.
Loading the ground truth and label the data.
```

Algorithm 1: Extracting Information (pseudocode)

Statistical analysis: Using Pandas to get the article created time and released time, then using matplotlib to plot the graph. As for the Origins (author), by the wordcloud tool support, Figure 5 could show the main authors and occupy the more space in the graph.

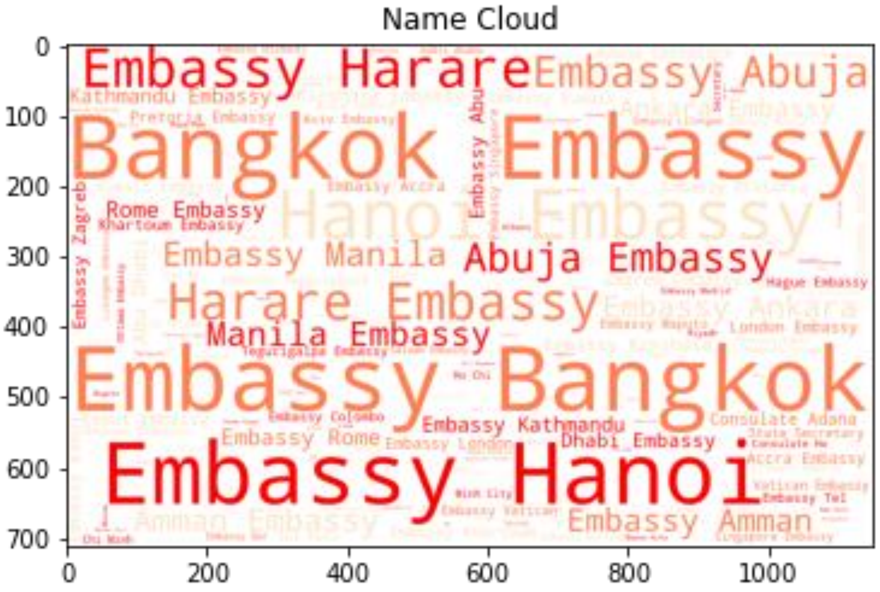
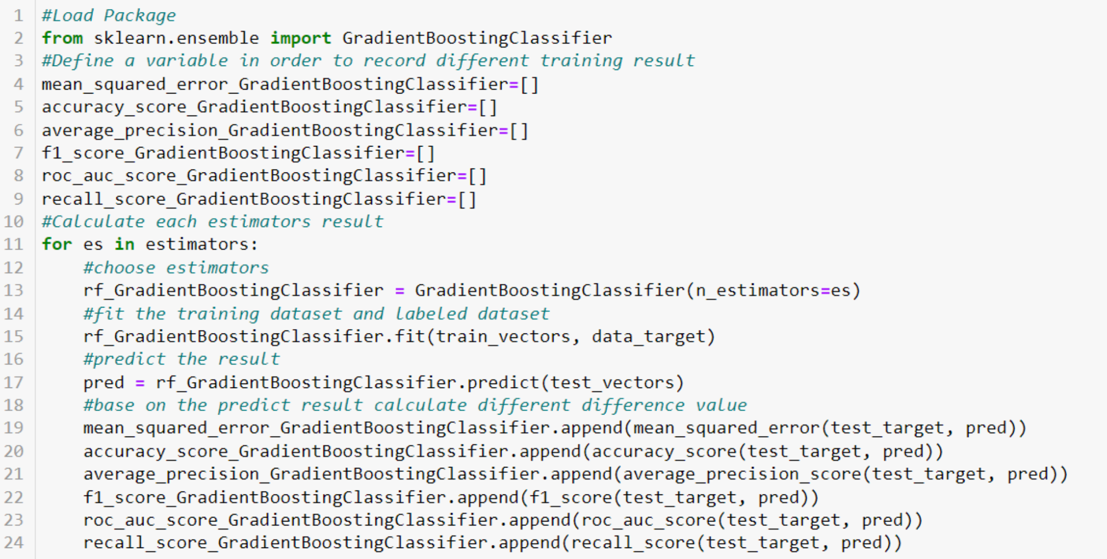


Figure 5: Name Cloud

Model Selection: First, a public stop_word[1] library is used. On this basis, sensitive words and invalid [1]words are artificially distinguished. For such articles, the accuracy of classification can be significantly improved and extreme deviations can be reduced. And use the sklearn's TF-IDF to transform the dataset.



```python
#Load Package
from sklearn.ensemble import GradientBoostingClassifier
#Define a variable in order to record different training result
mean_squared_error_GradientBoostingClassifier=[]
accuracy_score_GradientBoostingClassifier=[]
average_precision_GradientBoostingClassifier=[]
f1_score_GradientBoostingClassifier=[]
roc_auc_score_GradientBoostingClassifier=[]
recall_score_GradientBoostingClassifier=[]
#Calculate each estimators result
for es in estimators:
    #choose estimators
    rf_GradientBoostingClassifier = GradientBoostingClassifier(n_estimators=es)
    #fit the training dataset and labeled dataset
    rf_GradientBoostingClassifier.fit(train_vectors, data_target)
    #predict the result
    pred = rf_GradientBoostingClassifier.predict(test_vectors)
    #base on the predict result calculate different difference value
    mean_squared_error_GradientBoostingClassifier.append(mean_squared_error(test_target, pred))
    accuracy_score_GradientBoostingClassifier.append(accuracy_score(test_target, pred))
    average_precision_GradientBoostingClassifier.append(average_precision_score(test_target, pred))
    f1_score_GradientBoostingClassifier.append(f1_score(test_target, pred))
    roc_auc_score_GradientBoostingClassifier.append(roc_auc_score(test_target, pred))
    recall_score_GradientBoostingClassifier.append(recall_score(test_target, pred))
```

Algorithm 2: Gradient Boosting Classifier model

Here are the four metrics[2] selected for explanation.

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

Mean F1 score:

$$\text{F1 score} = \frac{2}{|C|}\sum_{i=1}^{|C|}\frac{TPR_i \times PPV_i}{TPR_i + PPV_i}$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$

AUC score:

$$\text{AUC} = \frac{\sum_{positive} rank_i - \frac{M(1+M)}{2}}{M * N}$$

[1] https://github.com/Alir3z4/stop-words/blob/master/english.txt

[2] TP - True Positives, FN - False Negatives, TN - True Negatives, FP - False Negatives. C is the number of classes

Finally, the model selection mainly bases on the sklearn's support, originally, the baseline is Random Forest Classifier with 200 estimators, the most accuracy is 71%, then trying the following methods to make a progress.

Method	Best estimators	Accuracy	F1 score	Recall	AUC score
RandomForestClassifier	200	0.71	0.81	**1.00**	0.76
ExtraTreesClassifier	150	0.73	0.79	0.98	0.73
KNeighborsClassifier	100	0.67	0.74	0.96	0.67
AdaBoostClassifier	200	0.72	0.78	0.98	0.72
GradientBoostingClassifier	200	**0.79**	**0.83**	**1.00**	**0.79**
BaggingClassifier	100	0.77	0.81	**1.00**	0.77
DecisionTreeClassifier	1000	0.74	0.79	0.98	0.74

Table 2: Model Selection Parameters and Performance

The above table shows as for the accuracy, F1 score and AUC score, Gradient Boosting Classifier has the highest score, which are **0.79**, **0.83** and **0.79** respectively. This is an average increase of 4% compared to Random Forest Classifier. For recall, Random Forest Classifier, Gradient Boosting Classifier and Bagging Classifier all have the highest Recall score, all of which are 1. Therefore, based on the results of the chart, the Gradient Boosting Classifier is the most suitable model.

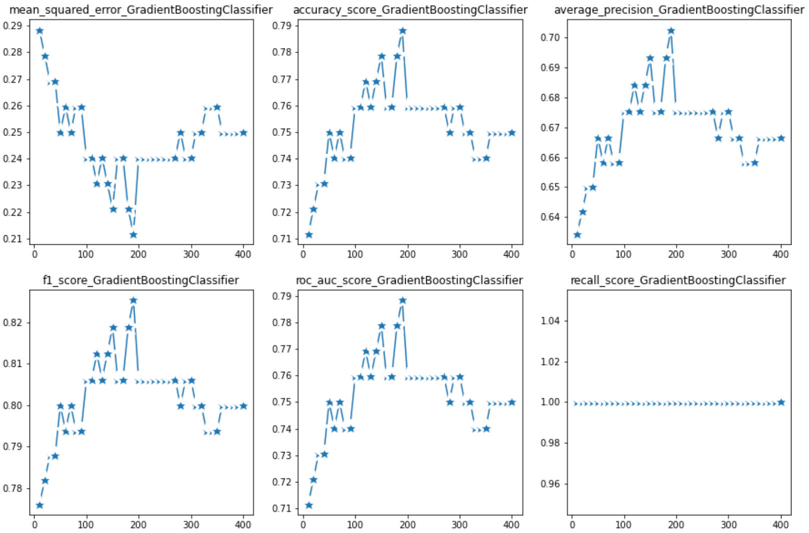


Figure 6: Gradient Boosting Classifier Performance

Figure 6 shows that the Gradient Boosting Classifier has the most accuracy score at 200 estimators, and it also get highest f1 score at 0.83 and AUC accuracy score at 0.79 separately.

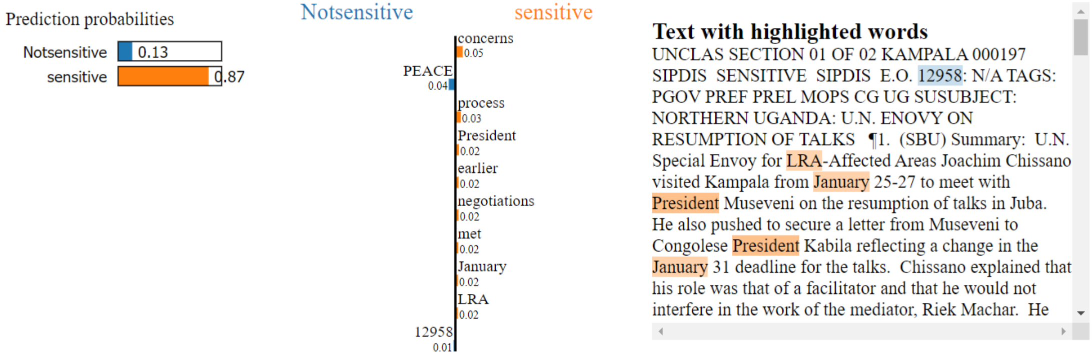
Generate the result: Based on the best model, generate each test file, and save the results in a structured way to facilitate storage and utilization. The result contains the sensitivity indicators, sensitive words and their proportions, as well as the analysis of the entire article content, take measures to highlight sensitive words, and use blue to emphasize non-sensitive words. Here, only the first 20 features are used as references. If too many references are selected, a large number of non-sensitive words may cover the proportion of sensitive words. HTML is still used for the saved files, which facilitates the front-end extraction and construction of new pages.

Here is an example of how to generate a result

```python
# Choose the best parameters and model as pipeline
rf = GradientBoostingClassifier(n_estimators=180)
# fit the training dataset and their label
rf.fit(train_vectors, data_target)
# make a pipeline
c = make_pipeline(vectorizer, rf)
# Load the result in order to label the result
class_names = ['Notsensitive', 'sensitive']
# build a explain model
explainer = LimeTextExplainer(class_names=class_names)
# choose a file to show up
idx=40
# get the explain result
exp = explainer.explain_instance(test_data[idx], c.predict_proba, num_features=20)
# show the explain result within html format
exp.show_in_notebook(text=True)
```

Algorithm 3: Generate a html result

A sample is shown below.



Text with highlighted words

UNCLAS SECTION 01 OF 02 KAMPALA 000197 SIPDIS SENSITIVE SIPDIS E.O. 12958: N/A TAGS: PGOV PREF PREL MOPS CG UG SUSUBJECT: NORTHERN UGANDA: U.N. ENOVY ON RESUMPTION OF TALKS ¶1. (SBU) Summary: U.N. Special Envoy for LRA-Affected Areas Joachim Chissano visited Kampala from January 25-27 to meet with President Museveni on the resumption of talks in Juba. He also pushed to secure a letter from Museveni to Congolese President Kabila reflecting a change in the January 31 deadline for the talks. Chissano explained that his role was that of a facilitator and that he would not interfere in the work of the mediator, Riek Machar. He

Figure 7: A template result

On the left of the graph, it is the prediction of the probabilities, in the middle of the picture, it is the key words among non-sensitive and sensitive, the left one is the content of the article with highlight words.

Structured storage: In order to make this project have a better user experience, all the files are trained here to get the corresponding results. In this way, the front-end call does not need to reload the model, which saves a lot of time.

5.2 Frontend

The front-end content is divided into two sections, one section introduces the tools that need to be used, and the second section shows the detailed page and interface.

5.2.1 Technical specification

For the front end, the flask framework is mainly used as the basis for development. It provides a good development environment for website construction and is also a relatively lightweight development framework, which can quickly develop and achieve the target effect in a short time.

Tools and technology used are as below:

No.	Tool	Justification

1	Flask	For this project, both Flask and Django can complete website construction. Relatively speaking, flask is a more lightweight framework, so flask is more suitable for this project.
2	Mpld3	Mpld3 is brings together Matplotlib, it is a popular JavaScript library for creating interactive data visualizations for the web. So, it is suitable for this application.
3	HTML, JavaScript, CSS	These are the fundamental components for any web site, HTML provide the basic web structure, JavaScript could implement some interactive events, and CSS is more concentrate on the color, structure and template.

Table 3: Frontend Technology

5.2.2 Frontend impalements details

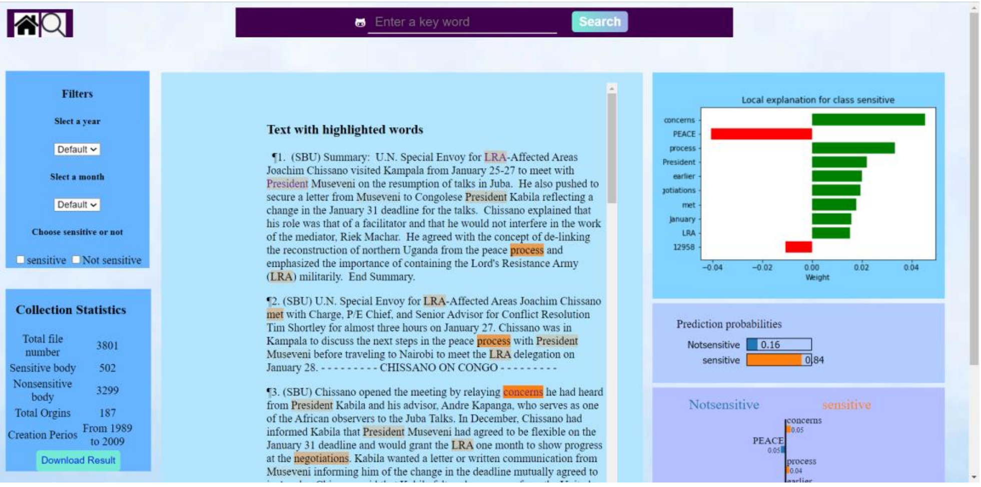
The following will explain in detail how each component is used, and the difficulties or frustrations encountered in the process.

Welcome Page: It contains a button which guide the user to Home Page.

Home Page: It is the main page of the application, it contains the filters, search box, visualization of the data, collection statistics. There are two parts of the home page, the first one is the filters and the other one is refreshing the homepage.

For the filter, first use html and css to typeset it, and then use the request method of the form form to implement the front-end data transmission function, index the data according to the transmitted conditions, and then save the indexed data in Variable.

Then use BeautifulSoup to extract the HTML table, classify the different tables, and send the corresponding feedback to the front-end page.



Collection Statistics

Total file number	3801
Sensitive body	502
Nonsensitive body	3299
Total Orgins	187
Creation Perios	From 1989 to 2009

Text with highlighted words

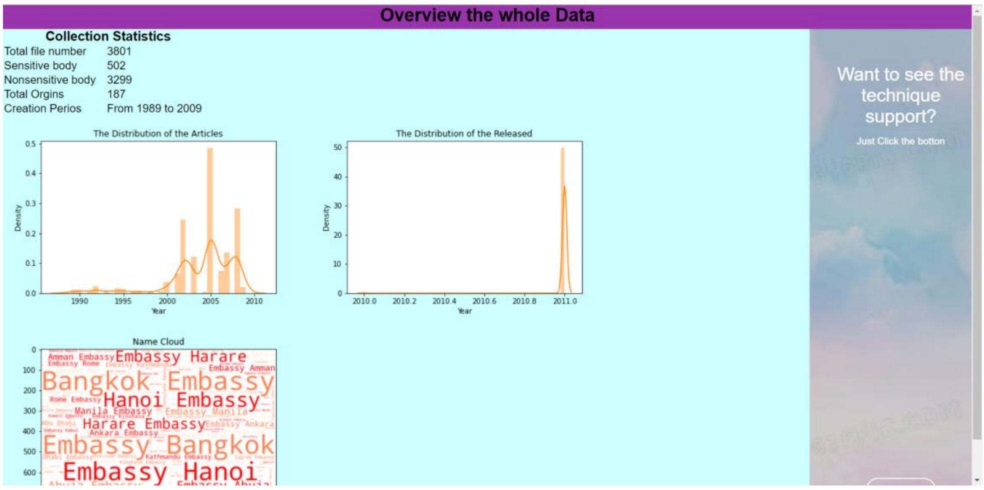
¶1. (SBU) Summary: U.N. Special Envoy for LRA-Affected Areas Joachim Chissano visited Kampala from January 25-27 to meet with President Museveni on the resumption of talks in Juba. He also pushed to secure a letter from Museveni to Congolese President Kabila reflecting a change in the January 31 deadline for the talks. Chissano explained that his role was that of a facilitator and that he would not interfere in the work of the mediator, Riek Machar. He agreed with the concept of de-linking the reconstruction of northern Uganda from the peace process and emphasized the importance of containing the Lord's Resistance Army (LRA) militarily. End Summary.

¶2. (SBU) U.N. Special Envoy for LRA-Affected Areas Joachim Chissano met with Charge, P/E Chief, and Senior Advisor for Conflict Resolution Tim Shortley for almost three hours on January 27. Chissano was in Kampala to discuss the next steps in the peace process with President Museveni before traveling to Nairobi to meet the LRA delegation on January 28. - - - - - - - - - CHISSANO ON CONGO - - - - - - - - -

¶3. (SBU) Chissano opened the meeting by relaying concerns he had heard from President Kabila and his advisor, Andre Kapanga, who serves as one of the African observers to the Juba Talks. In December, Chissano had informed Kabila that President Museveni had agreed to be flexible on the January 31 deadline and would grant the LRA one month to show progress at the negotiations. Kabila wanted a letter or written communication from Museveni informing him of the change in the deadline mutually agreed to

Figure 8: Screenshot the home interface

Statistical Page: This page provides theoretical support for the results of Home Page. It contains two parts, one is the statistical analysis of the article, and the other is the details of Model Selection.



Overview the whole Data

Collection Statistics

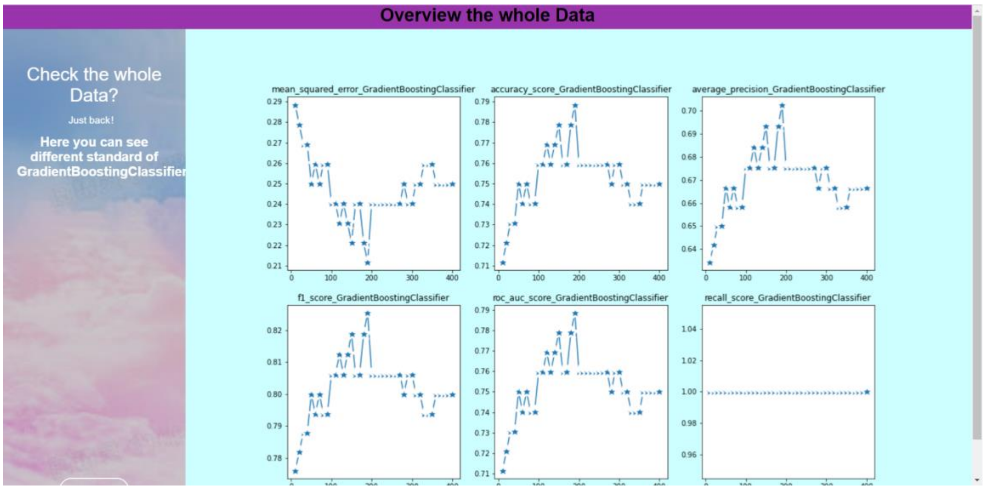
Total file number	3801
Sensitive body	502
Nonsensitive body	3299
Total Orgins	187
Creation Perios	From 1989 to 2009

Want to see the technique support?

Just Click the botton

Figure 9: collection statistics page

For the statistical content of the article, it includes the basic information of the article, the time distribution of creation and publication. For model selection, users can choose the methods shown in Table 1 to view their training process one by one.



Overview the whole Data

Check the whole Data?

Just back!

Here you can see different standard of GradientBoostingClassifier

mean_squared_error_GradientBoostingClassifier
0.21
0.22
0.23
0.24
0.25
0.26
0.27
0.28
0.29
0
100
200
300
400
accuracy_score_GradientBoostingClassifier
0.71
0.72
0.73
0.74
0.75
0.76
0.77
0.78
0.79
average_precision_GradientBoostingClassifier
0.64
0.65
0.66
0.67
0.68
0.69
0.70
f1_score_GradientBoostingClassifier
0.78
0.79
0.80
0.81
0.82
roc_auc_score_GradientBoostingClassifier
0.71
0.72
0.73
0.74
0.75
0.76
0.77
0.78
0.79
recall_score_GradientBoostingClassifier
0.96
0.98
1.00
1.02
1.04

Figure 10: Gradient Classifier Result

Chapter 6 Evaluation

This article mainly discusses testing and user study for this application. In addition, an evaluation experiment was carried out to test the user's time to complete the task.

The experiment is conducted as follows:

- Overall 12 participants were having valuable questionnaires sheet.
- In order to ensure the normal process of the task, two experimenters were selected as experiments and a set of standard processes were determined.
- Participants are asked to do two sections of content, one is quantitative operation, the other is qualitative comprehensive analysis.
- Participants are required to evaluate their own experiments, including whether they are pleasant or not, and their subjective feelings when using the software.
- Given some fixed tasks for them, let them index and view the corresponding pages.
- After doing the experiment, ask them how they feel about the page, as well as relevant suggestions.
- Ask them whether their understanding of the chart meets the expectations of the chart.
- The operating environment of the experiment is as follows:
  Processor: Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz 2.21 GHz RAM: 16.0 GB (15.8 GB usable) System type: 64-bit operating system, x64-based processor Web-Server: Flask development server, Browser: Microsoft Edge 96.0

Appendix-B includes the User Tasks in this experiment.

6.1 Testing Strategy

As for the backend component, the test methods are including:

Test-Backend-1: Check whether the data is complete: Check whether the extracted features or content meet the specifications and whether there are any deficiencies, etc. And verify that the data is correctly classified and stored.

Test-Backend-2: Check the legitimacy of the data results: The sensitive words and non-sensitive words are distinguished, and whether there are label data or mixed data as features.

Test-Backend-3: Test the effect of data classification: Manually check extreme classifications, and find words that have a greater impact on weights but have no practical meaning.

As for the backend component, the test methods are including:

Test-Frontend-1: Check whether the page buttons, options, and page navigation can operate normally: check whether all the controls can operate normally, and whether the content entered or selected by the user in the search box meets the standard.

Test-Frontend-2: Verify the results of the search: verify whether the article inquired is the target article.

Test-Frontend-3: Verify that keywords on the page can automatically jump to the wiki: For some keywords, the wiki page is set to automatically jump, that is, the wiki is used to query the current keywords.

Test-Frontend-4: Verify text data: Compare whether the content displayed on the static web page matches the actual content.

Test-Frontend-5: Verify model selection: Check whether each model correctly displays its training process and corresponding parameters.

Test-Frontend-6: Verify page interaction: check whether the page can interact normally, whether it can be returned or selected.

Test-Frontend-7: Verification log function: Check the back-end log record of the page, as well as the user's selection and feedback content.

In order to verify the contents of the front and back ends, the following tasks are set up for detection.

Requirement	Design Section	Test Plan
R1-Interface for Index	F-1, F-2, F-3	TF-1, TF-2, TF-4
R2-Navigation	F-5	TF-6
R3-Collection Statistics	F-4	TF-2, TF-3, TF-4
R4-Operation log	F-5	TF-7

Table 4: The Requirement Operation

6.2 Evaluation Strategy

When designing user tests, it is mainly carried out around the following issues:

1. Can the user learn to use it in a short time and find the target article?

For this application software, it needs to provide services for different users. How to let users get started quickly is also an assessment content. At the same time, whether the target article can be indexed in a short time is also a criterion for consideration.

2. Can the user understand the given form and picture, and whether it needs further explanation.

We have given a lot of pictures and explanations in this application. Not all reviewers have the corresponding professional knowledge and background, so you need to provide them with popular explanations when necessary.

3. Does the user approve the analysis of the application?

A large number of explanations and explanations are given in the application as to why it is a sensitive article or a non-sensitive article. Are all the words listed as keywords? To a certain extent, some keywords are not judged as sensitive or insensitive words. Therefore, for this possible situation, whether the user approves of this application.

Experimental Design:

In order to be able to test the function of the application, the software is divided into two types of tests. One is the use test of the software, which includes the efficiency test of the application, the overall response of the participants and the time required to complete the target task. The other is the understanding test of the icon, which checks whether the user can understand the content shown in the picture. By capturing the start and end time, and the tasks completed by the user as measurement indicators, the following is the measurement standard.

Criteria	Measure	Type	Description
Efficiency	Learn to Use	Quantitative	Learning time
	Index the article	Quantitative	Index time
Effectiveness	Savvy	Qualitative	Comprehension ability
	Task Completion Time	Quantitative	Log file to record
	Comfort level	Qualitative	Visualization effeteness
Feed Back	Record	Qualitative	User's Feedback

Table 5: Different part of Experiment

6.3 Evaluation Results

The evaluation results are mainly carried out from two aspects. One is a quantifiable result, which includes the user's use time and the user's subjective evaluation of the application's difficulty. The second is a more subjective evaluation, which includes the user's direct evaluation and subjective likes.

6.3.1 Quantitative result

The test time of the entire project is the time the user spends from opening the webpage to exiting the webpage.

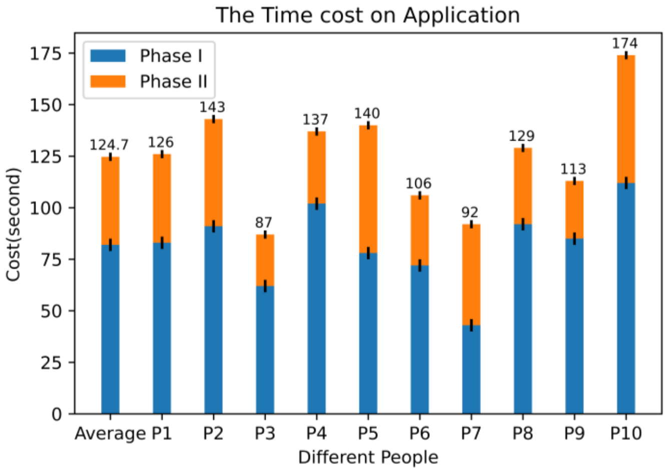


Figure 11: Time cost on the Application

The average time is 124.7 seconds to finish the task (not include typing or commutate time). From the perspective of time distribution, the time is controlled between 90 seconds and 180 seconds, which is acceptable for a reviewer, because a certain amount of learning time is also included here. For a person who uses the application, this time can be greatly reduced.

Next is the evaluation of the difficulty of the task, including some evaluations of the difficulty of getting started, the difficulty of cognition and so on. This difficulty assessment is judged by using a scoring mechanism. Its judgment interval is 0 to 5 points.

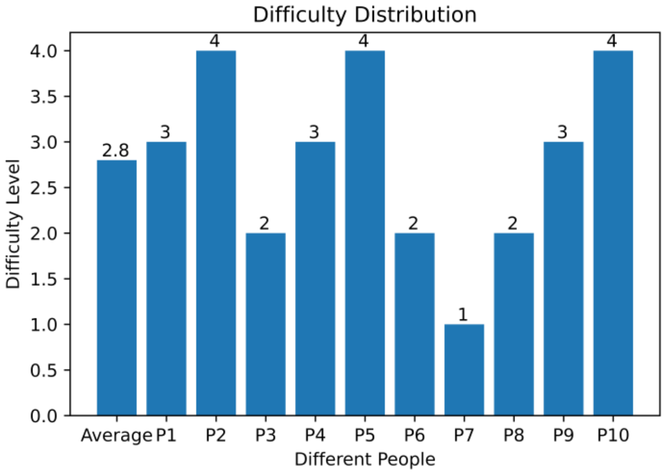


Figure 12: Difficulty Distribution

The average difficulty is 2.8 points, this is an extremely subjective evaluation, but it still has a certain reference significance. If everyone's score is too high, it means that the project is difficult to get started, so more interactions need to be added to guide users to learn and use. And from the distribution between the cost time and difficulty distribution, there seems have strong relative between them. The people who cost less time on it, he/she also think it is easy to use.

6.3.2 Qualitative Result

Here are lots of subjective evaluations with personal emotions, it is difficult to judge with specific numbers or vocabulary, so some valuable suggestions or feedback will be listed below.

People	Suggestions/Feedback
P1	The navigation page of the project can be optimized to make it better integrate into the page.
P1&P3	Hope there are more debugging buttons or explanations
P5	Harmonious page color matching
P6&P8&P9	Model selection content is clear
P7	Easy and fast to use
P7&P4	Can clearly see sensitive and non-sensitive words
P10	Wikipedia automatically jumps to the page and is easy to use

Table 6: User's Feedback

Another evaluation is the degree of recognition or preference for the system.

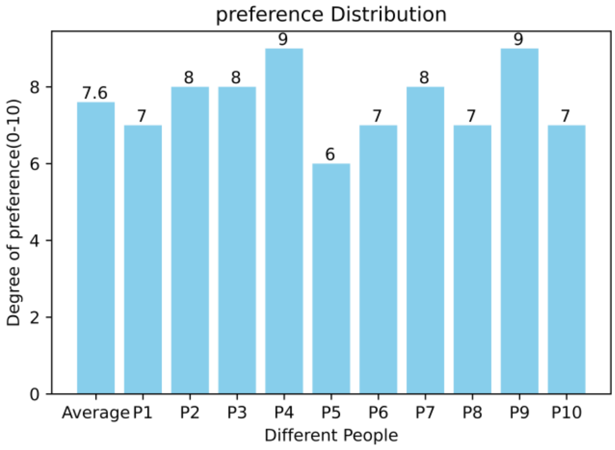


Figure 13: The User's Preference Distribution

From the graph, the average preference is 7.6 score, among them, some people like its page, and some people like the functions it provides. Of course, some people don't like this format very much. Perhaps because the experimenters are all Chinese, the scores are generally high. This may be due to China's inherent culture and will not embarrass others.

6.4 Discussion

After a series of development, testing, communication with teachers, user study, improvement, now summarize all the situation.

This application is a convenient operation program for an introductory user. It provides suggested operations and certain explanations, but it does not give sufficient and reasonable explanations for some professional knowledge, even though it provides certain the explanations include Wikipedia jumps, etc., but there are still certain difficulties for a non-computer professional user.

For a person in the computer industry, this software is very easy to use, and at the same time it provides sufficient explanations and sufficient explanations on how to choose models. The requirements for visualization or aesthetics are not so high, because as long as the image is clearly explained, there is no need to do so many modifications.

Generally speaking, this project is an offline project, and all the results have been trained in advance, so basically no time is spent on indexing, which provides great convenience for reviewers and also ensures that the results are traceable. Of course, there are also some areas that can be improved, which will be explained in future work.

Chapter 7 Conclusion

The project aims to design a practical review tool to help reviewers conduct sensitivity reviews. It mainly used the method of visualizing sensitive words and non-sensitive words to highlight the basis of judgment, and at the same time provided their weights to better understand and analyze the article.

It has completed its due tasks and functions, and can provide reviewers with an efficient review method. Suggested ways to improve are discussed in the next section, which can be used as the direction of project improvement.

In summary, the application implements the correct classification and provides a reasonable visualization page and has the function of multi-angle observation.

7.1 Future Work

The project still has a lot of room for improvement, but due to the constraints of time and the author's personal ability, some content can be extended well. First, in the process of back-end processing, operations such as entity linking can be performed on the entities that have been extracted, the internal connections between articles can be seen through the method of graph database, and the extracted entities can also be used to build keyword queries.

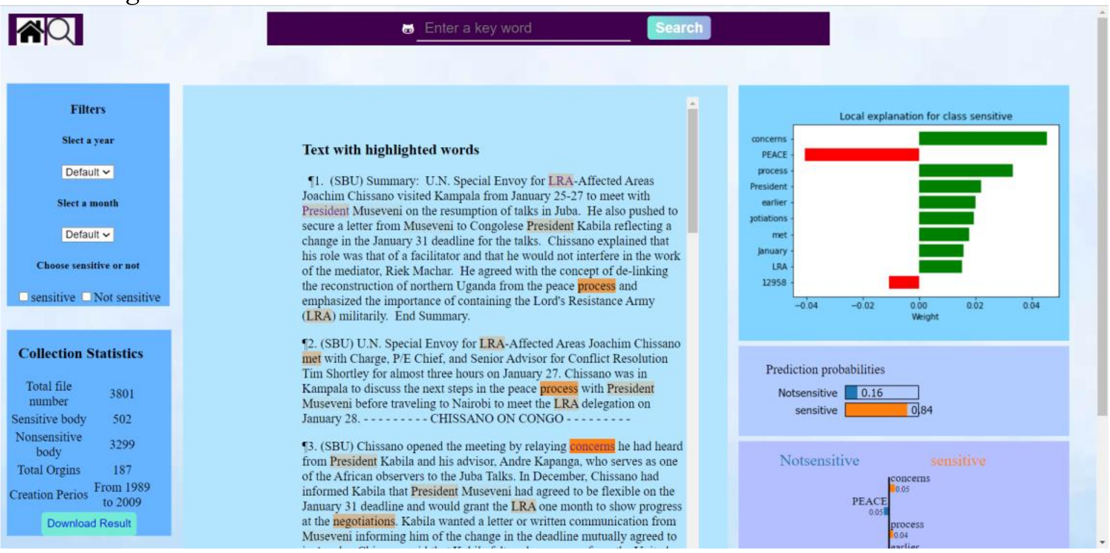
In addition, you can also launch online training modules. For this content, most of the content has been completed, and you can select data for training and output, but it is temporarily shelved because it does not match the general direction of the project. If it is implemented Greatly expand the effectiveness of the program, and provide reviewers with the ability to update the model in real time.

Furthermore, you can also build article similarity to compare content, and find articles with higher similarity to the current article, which can speed up the review of content in the same field by reviewers, thereby greatly increasing the review rate.

Appendix A Website Screenshots and Manual

Welcome Page

Home Page



Collection Statistics

Total file number	3801
Sensitive body	502
Nonsensitive body	3299
Total Orgins	187
Creation Perios	From 1989 to 2009

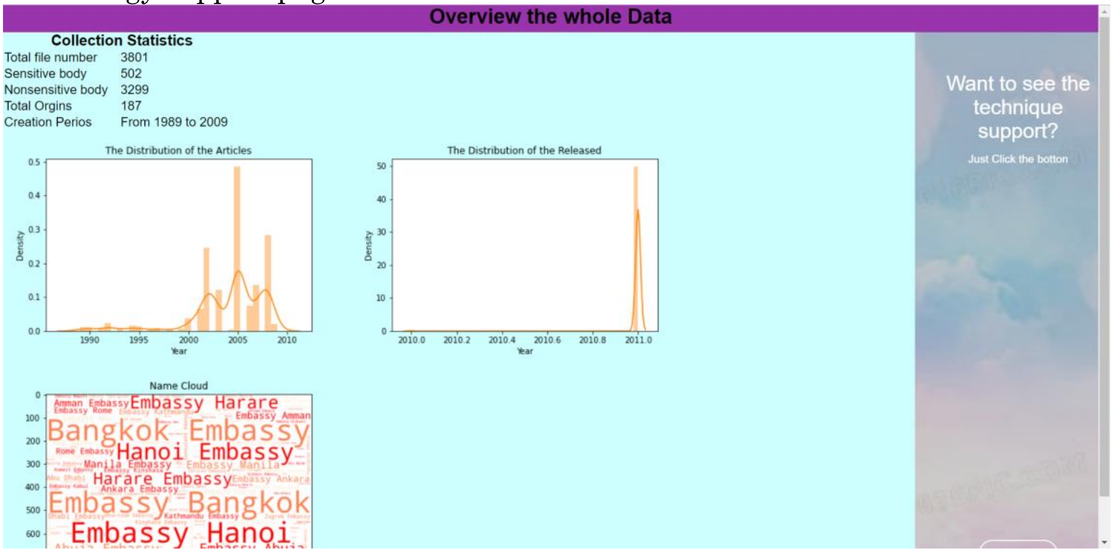
Text with highlighted words

¶1. (SBU) Summary: U.N. Special Envoy for LRA-Affected Areas Joachim Chissano visited Kampala from January 25-27 to meet with President Museveni on the resumption of talks in Juba. He also pushed to secure a letter from Museveni to Congolese President Kabila reflecting a change in the January 31 deadline for the talks. Chissano explained that his role was that of a facilitator and that he would not interfere in the work of the mediator, Riek Machar. He agreed with the concept of de-linking the reconstruction of northern Uganda from the peace process and emphasized the importance of containing the Lord's Resistance Army (LRA) militarily. End Summary.

¶2. (SBU) U.N. Special Envoy for LRA-Affected Areas Joachim Chissano met with Charge, P/E Chief, and Senior Advisor for Conflict Resolution Tim Shortley for almost three hours on January 27. Chissano was in Kampala to discuss the next steps in the peace process with President Museveni before traveling to Nairobi to meet the LRA delegation on January 28. - - - - - - - - - CHISSANO ON CONGO - - - - - - - - -

¶3. (SBU) Chissano opened the meeting by relaying concerns he had heard from President Kabila and his advisor, Andre Kapanga, who serves as one of the African observers to the Juba Talks. In December, Chissano had informed Kabila that President Museveni had agreed to be flexible on the January 31 deadline and would grant the LRA one month to show progress at the negotiations. Kabila wanted a letter or written communication from Museveni informing him of the change in the deadline mutually agreed to

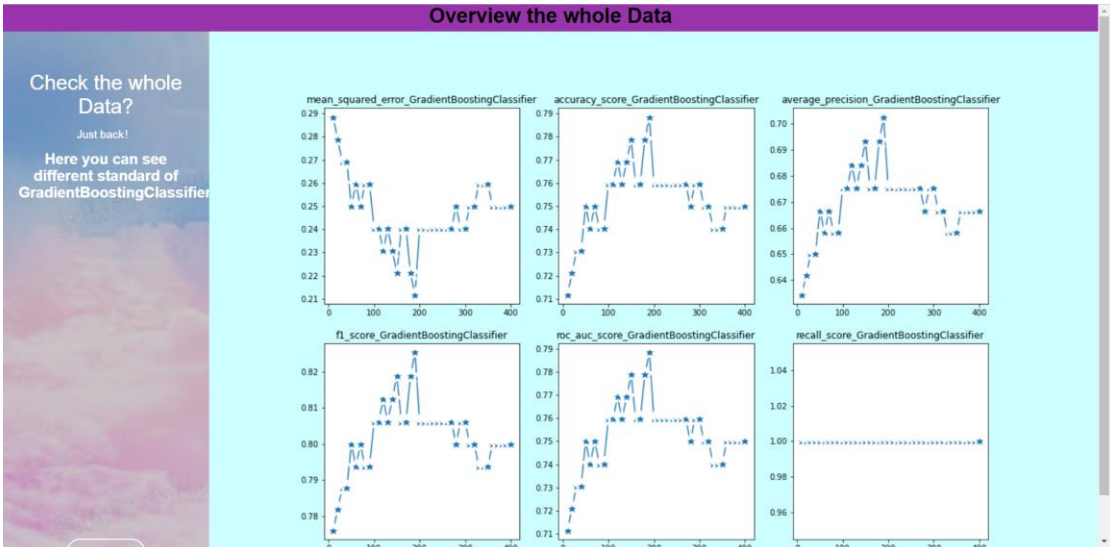
Technology support page



Collection Statistics

Total file number	3801
Sensitive body	502
Nonsensitive body	3299
Total Orgins	187
Creation Perios	From 1989 to 2009

Model Selection page



Overview the whole Data

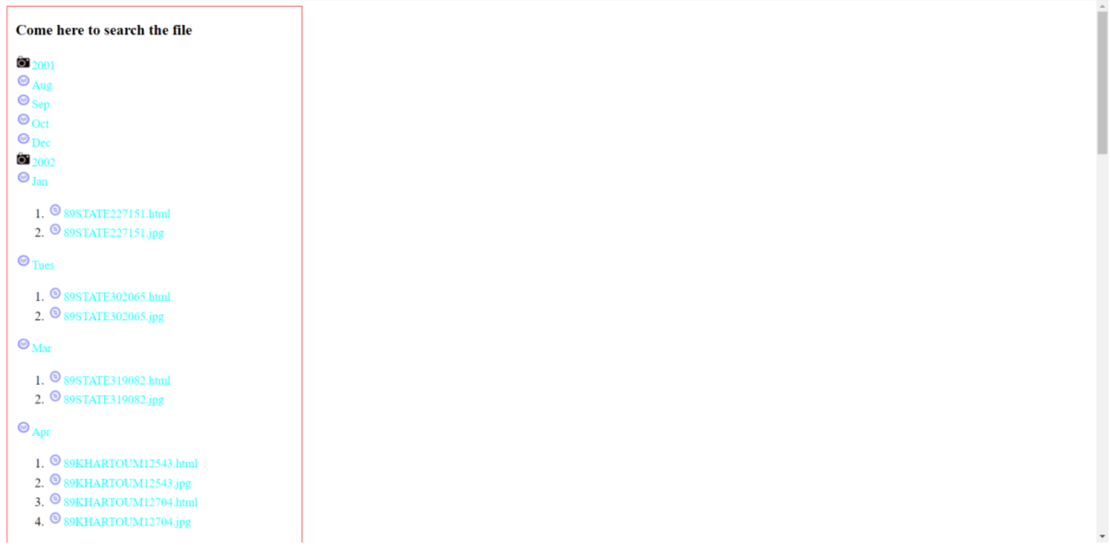
Check the whole Data?

Just back!

Here you can see different standard of GradientBoostingClassifier

mean_squared_error_GradientBoostingClassifier
accuracy_score_GradientBoostingClassifier
average_precision_GradientBoostingClassifier
f1_score_GradientBoostingClassifier
roc_auc_score_GradientBoostingClassifier
recall_score_GradientBoostingClassifier

All result file page (not link now)



Come here to search the file

- 2001
- Aug
- Sep
- Oct
- Dec
- 2002
- Jan

1. 89STATE227151.html
2. 89STATE227151.jpg

- Tues

1. 89STATE302065.html
2. 89STATE302065.jpg

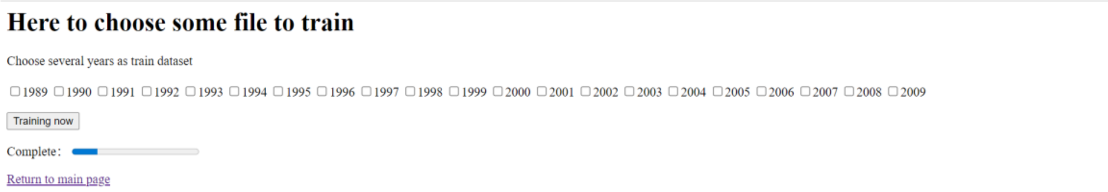
- Mar

1. 89STATE319082.html
2. 89STATE319082.jpg

- Apr

1. 89KHARTOUM12543.html
2. 89KHARTOUM12543.jpg
3. 89KHARTOUM12704.html
4. 89KHARTOUM12704.jpg

0nline Training Page



Here to choose some file to train

Choose several years as train dataset

□1989 □1990 □1991 □1992 □1993 □1994 □1995 □1996 □1997 □1998 □1999 □2000 □2001 □2002 □2003 □2004 □2005 □2006 □2007 □2008 □2009

Complete:

Return to main page

Appendix B User Task Sheet

Phase 1: Efficiency

Task1:

Now open your application and use the navigation interface to browse the framework of the entire project.

Task2:

Select an article named 92ZAGREB1002, and see the result.

Task3:

Select an article named 92ZAGREB557 that occurred in April 1992 and view the results.

Phase 2: Effectiveness

Task4:

In your opinion, talk about what you get from the Task 2 result.

Task5:

Talk about what you get from the technology support page.

Task6:

Do you have any suggestion or feedback?

Bibliography

[1] “Freedom of Information Act 2000”. [Online] <http://www.legislation.gov.uk/ukpga/2000/36/contents>

[2] McDonald, G., Macdonald, C., Ounis, I. and Gollins, T., “Towards a classifier for digital sensitivity review”. In European Conference on Information Retrieval (pp. 500-506), 2014. Springer, Cham.

[3] McDonald, G., Macdonald, C. and Ounis, I., “Enhancing sensitivity classification with semantic features using word embeddings”. In European Conference on Information Retrieval (pp. 450-463), 2017. Springer, Cham.

[4] Roberto Basili. 2003. Book review: learning to classify text using support vector machines: methods, theory, and algorithms by thorsten joachims cornell university dordrecht: kluwer academic publishers, 2002, xvii+205 pp; hardbound, isbn 0-7923-7679-x. Comput. Linguist. 29, 4 (December 2003), 655–661. Marco Tulio Ribeiro,

[5] Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 1135–1144. DOI:https://doi.org/10.1145/2939672.2939778

[6] Shota Okumura, Yoshiki Suzuki, and Ichiro Takeuchi. 2015. Quick Sensitivity Analysis for Incremental Data Modification and Its Application to Leave-one-out CV in Linear Classification Problems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). Association for Computing Machinery, New York, NY, USA, 885–894. DOI:https://doi.org/10.1145/2783258.2783347

[7] Likoebe Mohau Maruping. 2006. Essays on agility in software development teams: process and governance perspectives. Ph.D. Dissertation. University of Maryland at College Park, USA. Advisor(s) Ritu Agarwal and Viswanath Venkatesh. Order Number: AAI3236745.

[8] Clegg, Dai; Barker, Richard (1994). Case Method Fast-Track: A RAD Approach. Addison-Wesley. ISBN 978-0-201-62432-8.

[9] Tanja Blascheck and Thomas Ertl. 2014. Towards analyzing eye tracking data for evaluating interactive visualization systems. In Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV '14). Association for Computing Machinery, New York, NY, USA, 70–77. DOI:https://doi.org/10.1145/2669557.2669569

[10]T. J. Jankun-Kelly and Kwan-Liu Ma. 2000. A spreadsheet interface for visualization exploration. In Proceedings of the conference on Visualization '00 (VIS '00). IEEE Computer Society Press, Washington, DC, USA, 69–76.

[11]Rob Lintern, Jeff Michaud, Margaret-Anne Storey, and Xiaomin Wu. 2003. Plugging-in visualization: experiences integrating a visualization tool with Eclipse. In Proceedings of the 2003 ACM symposium on Software visualization (SoftVis '03). Association for Computing Machinery, New York, NY, USA, 47–ff. DOI:https://doi.org/10.1145/774833.774840

[12]Michaël Aupetit. 2014. Sanity check for class-coloring-based evaluation of dimension reduction techniques. In Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV '14). Association for Computing Machinery, New York, NY, USA, 134–141. DOI:https://doi.org/10.1145/2669557.2669578

[13]Siirtola H, Räihä KJ, Säily T, Nevalainen T. Information visualization for corpus linguistics: Towards interactive tools. InProceedings of the first

international workshop on Intelligent visual interfaces for text analysis 2010 Feb 7 (pp. 33-36).

[14]Shrinivasan YB, van Wijk JJ. Supporting the analytical reasoning process in information visualization. InProceedings of the SIGCHI conference on human factors in computing systems 2008 Apr 6 (pp. 1237-1246).

[15]Väätäjä H, Varsaluoma J, Heimonen T, Tiitinen K, Hakulinen J, Turunen M, Nieminen H, Ihantola P. Information visualization heuristics in practical expert evaluation. InProceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization 2016 Oct 24 (pp. 36-43).

[16]Bown J, Fee K, Sampson A, Shovman M, Falconer R, Goltsov A, Issacs J, Robertson P, Scott-Brown K, Szymkowiak A. Information visualization and the arts-science-social science interface. InProceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia 2010 Dec 27 (pp. 9-17).

[17]Dickens C. A tale of two cities [1859]. Gawthorn; 1949.

[18]Kang X. The effect of color on short-term memory in information visualization. Inproceedings of the 9th International Symposium on Visual Information Communication and Interaction 2016 Sep 24 (pp. 144-145).