

# COMP1804 REPORT

Word count: 2631

## Table of Contents

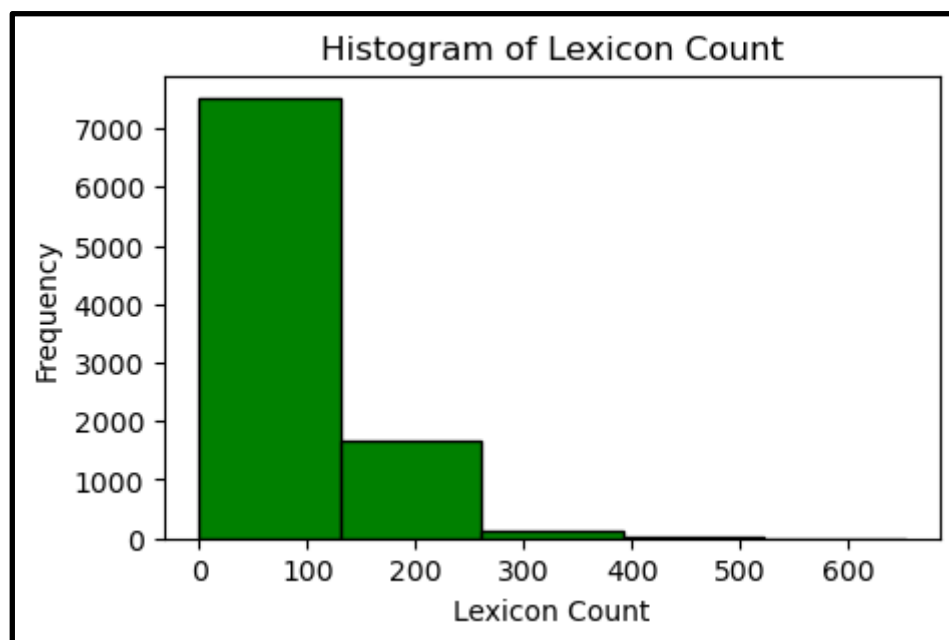
Executive Summary .....	2
1. Data exploration and assessment .....	2
2. Data splitting and cleaning.....	4
3. Data encoding .....	4
4. Task 1: Topic classification .....	6
4a. Model building .....	6
4b. Model evaluation .....	7
4c. Task 1 Conclusions .....	8
5. Task 2: Text clarity classification prototype.....	8
5a. Ethical discussion.....	8
5b. Data labelling .....	9
5c. Model building and evaluation.....	10
5d. Task 2 Conclusions .....	11
6. Self-reflection .....	11
References .....	12

## Executive Summary

Text clarity and topic categorization models were constructed and evaluated after data cleansing, preparation, and analysis. Beyond baseline, logistic regression topic classification has been accurate. Text clarity classification was ethically evaluated for risks and biases. The model has been built using the AdaBoost classifier and offered improvements. The research demonstrated effective text analysis data management and model creation.

### 1. Data Exploration and Assessment

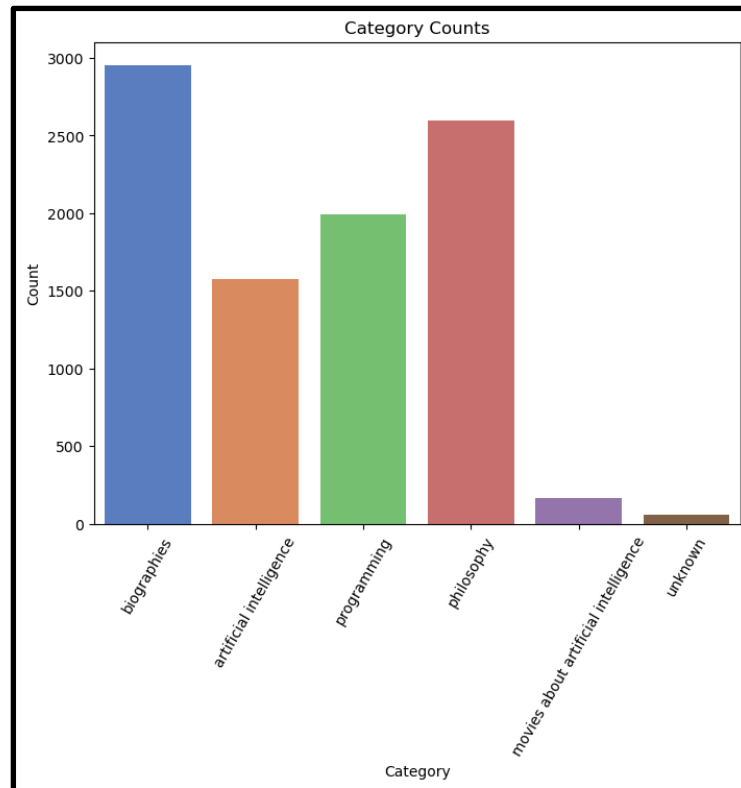
The dataset consists of 9347 paragraphs and is divided into eight columns. These are “par\_id,” “paragraph,” “text\_clarity,” “has\_entity” (whether the paragraph contains an entity), “lexicon\_count” (number of words in the paragraph), “difficult\_words” (number of difficult words), “last\_editor\_gender”(that is, gender of the last editor), “category”, and “text\_clarity”. There are nine non-null values in ‘text\_clarity’ which is a remarkable fact (Abiola *et al.* 2023). In conclusion, it can be seen that this dataset provides a vast amount of information about paragraphs such as their features, contents, or type.



**Figure 1: Histogram for Counting of the Lexicon.**

Figure 1 displays a histogram of the words in the paragraph. The X-axis represents the count of words in the paragraph according to the dictionary. The y-axis displays the frequency of paragraphs based on lexical count bins. The top of the histogram displays a higher number of

paragraphs with lexical counts ranging from 0 to 100. There are over 7000 paragraphs in this section. Longer paragraphs become less common after reaching 100 words. The distribution presented here demonstrates that paragraphs of shorter length contain fewer words, while longer paragraphs are infrequent (Bashynska *et al.* 2024). Paragraph length trends and lexical count distribution can be indicated by histograms.



**Figure 2: Category Distribution for Text Data.**

The Count plots displayed in the above figure show the distribution of paragraph categories across the dataset. The plot bars represent different categories, and their height corresponds to the number of paragraphs in each category. According to the chart, 'Biographies' has a total of 2955 paragraphs. Following 'programming' is 'philosophy' with 2598 paragraphs. Paragraph topics are dominated by programming, philosophy, and biography, according to the data. The paper contains 1576 paragraphs on artificial intelligence and 167 paragraphs on films about artificial intelligence. The sections extensively cover artificial intelligence. There have been only 61 paragraphs labelled as "unknown," which suggests a relatively small proportion. The count plot showcases themes, paragraph categories, and dataset organisation. This information can assist in understanding the distribution of materials in the dataset and can be useful for further analysis or modelling of relevant categories (Ellahi *et al.* 2024).

## 2. Data Splitting and Cleaning

Addressing missing values, ensuring consistency in category labels, and converting text to numbers have been crucial steps in preparing and separating the data. The processes involved in preparing the data ensured accurate classifiers had been generated for model training and testing.

The data underwent thorough checking during the preparation and splitting process to identify and rectify any missing values. The `IsNull()` function. There have been 18 difficult words and 61 categories that are lacking. The 'text\_clarity' column has been not suitable for investigation because of a significant amount of missing data (9338 out of 9347). When the paragraph has been free of any issues, the 'difficult\_words' column has been adjusted to fix any missing values (Chen *et al.* 2023). To classify each instance, the missing values in the 'category' column have been replaced with 'unknown'.

The capitalization of 'biographies' and 'Biographies' in the 'category' column is inconsistent. Reduced dataset redundancy by converting the 'category' to lowercase.

The 'preprocessed\_paragraph' column underwent TF-IDF vectorization as part of the feature engineering process. The category identification method utilises text and corpus phrase frequency. TF-IDF vectors have been used to encode and stack the 'has\_entity' column horizontally with the help of the stack function of the NumPy module. Model testing and feature set training have been employed.

Following the cleaning process, the dataset is split into training and testing sets by `train_test_split`. Evaluating the model using previously untested data confirmed its overall applicability. This division allowed it. To ensure repeatability, the test sample has been set at 20% and the random state has been set to 125. The 'has\_entity' column designates numbered categories. Numerical input translation is required for machine learning. The category input has been encoded using the Label Encoder function from Scikit-learn (Gao *et al.* 2023).

## 3. Data Encoding

Encoding techniques have been chosen based on several parameters to prepare data for analysis and modelling, enabling accurate and reliable classification algorithms. Each encoding process altered raw data so machine learning algorithms could detect patterns and relationships. A

detailed data encoding strategy enhances machine learning model prediction and data-driven decision-making.

Encoding of has\_entity: At the beginning, the 'has\_entity' column has been made to contain the categorical values that point out whether an entity has been present or absent in any given paragraph. This feature has been encoded by the function Label Encoder from the library of scikit-learn. The procedure of encoding contained the re-labelling of the “1” values of the category, which has been equal to the conversion of “yes” to “1” and “no” to “0.” Ordinal-relationship categorical data will nevertheless be encrypted into numbers that count, through the label encoder which is a method utilised for its simplicity and effectiveness (Hu *et al.* 2023).

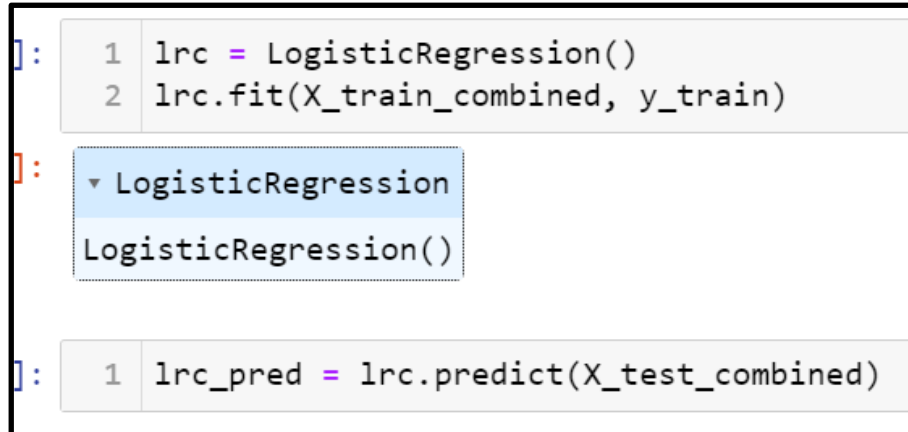
Text Encoding (TF-IDF Vectorization): The Text column has been populated with the processed text that has been an accurate representation of the paragraph. The feature has been translated into a format designed for machine learning models utilising the TF-IDF vectorization approach of the information retrieval unit. This approach transforms text documents into numerical representations by assigning weights (weights) to terms (themselves, the totality). One of the criteria for weighing up is the term’s frequency inside the document, as well as across the whole corpus. TF-IDF vectorization has been chosen because it can dampen the influence of the common words while being great at assessing the documents employing the indicative words that are used.

Horizontal Stacking of Encoded Features: Aggregate labels and text feature vectors corresponding to the 'has-entity' and 'pre-processed paragraph' columns have been created to form combined training feature sets. Through this process, the textual and categorical data has been converted to a single unified format that has been friendly to the algorithms which used the machine learning processes.

These criteria have been based on the efficacy of the encoding mechanisms in dealing with various features of the dataset in question. There is a method of encoding which has been employed to hold the categorical attributes that had an ordinal nature, such as the presence or absence of objects. Word semantics and textual data grouping have been captured using TF-IDF vectorization. Horizontally stacking encoded features combined all relevant data for model training and assessment.

## 4. Task 1: Topic Classification

### 4a. Model Building



```
1 lrc = LogisticRegression()
2 lrc.fit(X_train_combined, y_train)

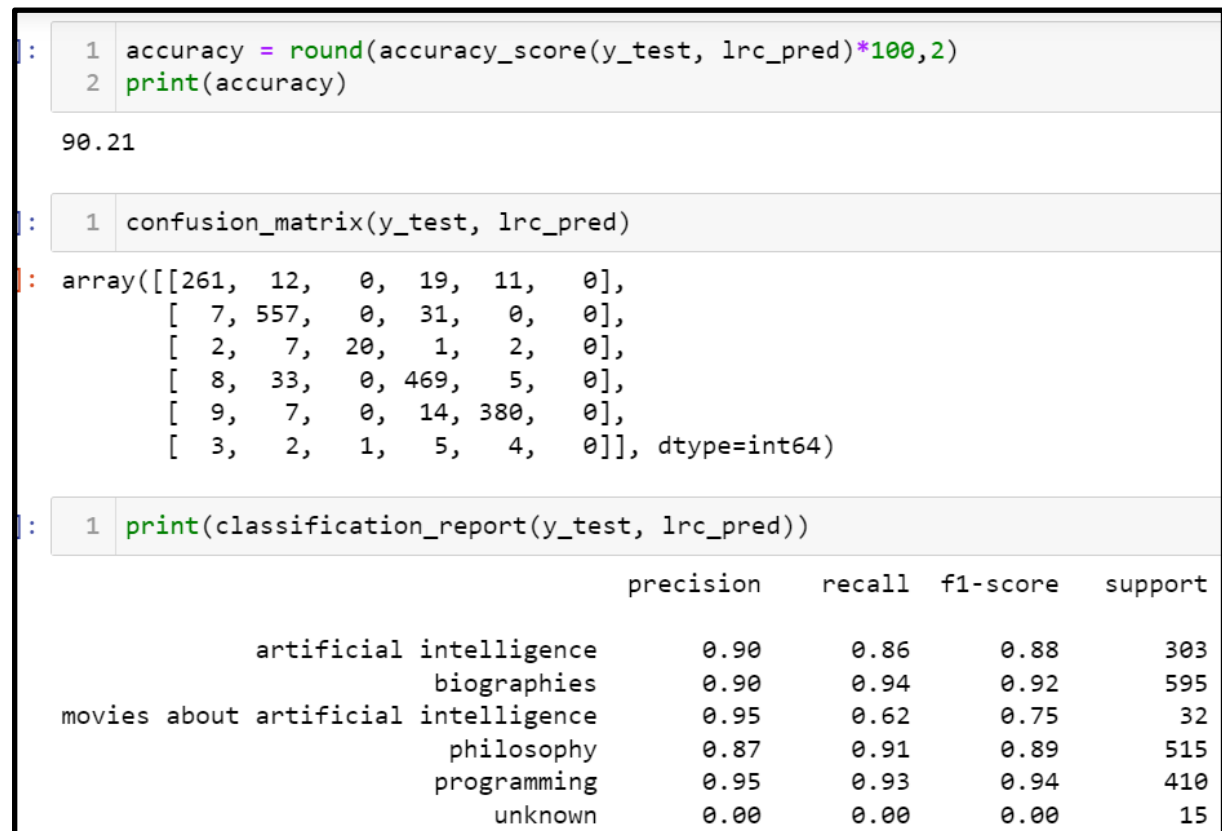
LogisticRegression()

1 lrc_pred = lrc.predict(X_test_combined)
```

**Figure 3: Logistic Regression for Analysis.**

The classifier of the topic, using TF-IDF vectorization along with logistic regression, was authorised to get the features in this project. The logistic regression was selected due to its simplicity, and its linear nature so one can predict with ease, and at the same time it is effective for handling multiclass classification tasks. The notion of TF-IDF vectorization is used here to compress the textual data into numerical values taking into account the prime significance of words in group differentiation. The precious model hyperparameters were adjusted by several rounds of validation through the grid search approaches, which were going from the varying values of regularisation parameters and solvers. The hyperparameters selected would be the ones that would be ideal for providing the model with the complexity to take into account the features of the problem as well as the generalizability to evaluate the model performance (Li *et al.* 2023). The presented technique provides a complete solution to group the topics with clear steps to match the customer need for theme classification organised by the contents they contain.

## 4b. Model evaluation



**Figure 4: Various evaluation metrics based on analysis.**

The model predicted the disease levels with an accuracy of about 90.21%, which means that it was able to correctly divide about 90.21% of the cases in the test set. The confusion matrices show the number of rows that got allocated to the respective columns. Here, each column is the occurrence in the supposed class, while each row is the class in the expected class. The matrix of confusion leaves no doubt that the model performs successfully and classifies a great part of categories. Despite these mistakes, in terms of accuracy, the two most successful categories were "biographies" (557 out of 595 occurrences) and "programming" (469 out of 515 instances).

The report on the classification metric serves as a comprehensive assessment of the model performance per class and includes measures of the model efficiency including accuracy, recall, and F-score. The term "recall" can be described as a proportion of a correctly identified genuine trigger within all true triggers, while precision is the proportion of precise positive determinations among all positive predictions. The harmonic mean of precision and recall gives a measure of a classifier's performance in the F1-score. This score gives an inclusive evaluation

of an algorithm. The model's overall performance is rather outstanding across the majority of categories, with strong recall and accuracy scores across the board. This is the case across the board. This is an indication of its capability to correctly classify textual input into a variety of topics, as shown by the fact that this is the case. It is unfortunate that the category that is referred to as "unknown" has a performance that is below average. This may be the consequence of the limited number of examples that are included under this particular class.

#### **4c. Task 1 Conclusions**

- The level of accuracy in the model is largely higher compared to the baseline which was just about 31.61%. This shows that the model is much better than the baseline. It can, therefore, be deduced that about a hypothetical guess, the model's forecasts are more accurate and this is congruent to job requirements point 1b as defined by the client.
- It would be highly recommended that another metric such as F1-score be used together with accuracy to evaluate model performance. However, it should be mentioned that for classification purposes recall and precision are not sufficient. By doing so, one can solve many problems faced while reducing both wrong positive and negative outcomes simultaneously in some cases where class imbalance exists. This measure allows customers to understand better how well models can strike a balance between recall and precision by monitoring their F1 scores. Thus, it ensures consistent and reliable performance on diverse categories of topic categorization across tasks.

### **5. Task 2: Text Clarity Classification Prototype**

#### **5a. Ethical Discussion**

Moral issues emerge when user contribution is rejected for linguistic clarity. Biases in algorithms are severe. Using training data that favours specific writing styles or linguistic trends may skew certain populations. The algorithm may have trouble identifying writing from many cultures or languages if the training data is largely from one. Automated writing intelligibility tests show accuracy and reliability issues. Text clarity is subjective and context-dependent; therefore, the algorithm may misclassify clear language as unclear. User criticism or demoralisation may result if they feel their input was neglected. These algorithms may unexpectedly affect users' autonomy and free speech. Automatic rejection of clear content may



limit users' creativity and honesty. Uninformed users may mistrust the platform's work appraisal and rejection methods. Text clarity algorithm development and usage must be open, responsible, and fair to eliminate ethical issues. Human supervision, including user comments and appeals, and bias-free training data selection are needed for automation. Regular algorithm assessment may reveal and address unanticipated effects and biases.

## **5b. Data Labelling**

The data labelling procedure for text clarity classification starts with criteria for paragraphs that are deemed "clear enough" or "not clear enough". The label accuracy and consistency are guaranteed. Making data annotation standards accessible to everyone. Labelling should take into account coherence, logic, language, and readability. Language that is clear, concise, and well-organised is considered "clear enough," while language that is imprecise, complex, and disorganised is considered "not clear enough." Clear instructions and category-specific examples are essential for ensuring consistency among annotators. This provides an explanation of labels and class composition for annotators.

Offer certainty or vagueness for each category. This could assist annotators in distinguishing between clear and unclear passages. Increased confidence in the recording label enhances the reliability of the data. Ensuring data correctness and consistency is a crucial part of quality assurance checks after labelling. Flaws or misclassifications may be revealed by annotated paragraphs. Evaluate label coherence using inter-annotator agreement measures (Ghafoor *et al.* 2023). Examine the tagged data for sections that are deemed clear enough and sections that are considered not clear enough. Below, you can find the distribution of labels at the class level and the total number of labelled data points. The summary deems 30% of the data as unclear and 70% as clear enough.

## 5c. Model Building and Evaluation

```
] 1 # Model Training
2 aboost = AdaBoostClassifier()
3 aboost.fit(X_train_tfipara_data, y_train)

] 1 AdaBoostClassifier
   AdaBoostClassifier()

] 1 # Model Evaluation
2 logreg_pred = aboost.predict(X_test_tfipara_data)
3 accuracy_score(y_test, logreg_pred)

] 0.5

] 1 print("Classification Report:\n", classification_report(y_test, logreg_pred))
Classification Report:
              precision    recall  f1-score   support

 clear_enough      0.40      0.22      0.29         9
not_clear_enough    0.53      0.73      0.62        11

   accuracy
macro avg      0.47      0.47      0.45         20
weighted avg    0.47      0.50      0.47         20
```

**Figure 5: Evaluation and Analysis based on AdaBoost Classification.**

The AdaBoost Classifier was trained to evaluate text clarity on labelled data using paragraph TF-IDF vector representations. Ensemble learning AdaBoost develops good classifiers from bad ones. Step-by-step instruction on prior errors improves classification accuracy for underperforming pupils. This work utilised AdaBoost Classifier default hyperparameters. This prototype may have poor model performance without hyperparameter optimisation or fine-tuning. A simple model proving machine learning can identify text clarity was the goal. Model accuracy was 50% after the test set. Unbalanced datasets may reduce classification model accuracy. Categorization reports were extensively examined.

Classification reports provide accuracy, precision, recall, and F1-score for obvious and uncertain classes. Recall is the percentage of class occurrences anticipated correctly. Precision is the proportion of accurately predicted classified cases. The harmonic mean of accuracy and recall concludes the F1-score classifier performance assessment. For "clear enough" scenarios, 0.40 accuracy means 40% of projected instances were clear. The 0.53 "not clear enough" category accuracy rating means 53% of projected circumstances were categorised correctly. The percentage of real cases of each class properly categorised is 0.22 and 0.73 for "clear enough" and "not clear enough" recall scores. The classification report showed the model predicted "not clear enough" better than "clear enough". The dataset's class imbalance may

explain why more paragraphs are "clear enough" than "not clear enough". To improve performance, the model may need further research. Fix class imbalance or model hyperparameters.

## **5d. Task 2 Conclusion**

- Model performance should meet client success criteria like minimum accuracy or task specification. If the buyer wants more precision, the model's 50% accuracy may disappoint. Optimisation and refining may improve client satisfaction.
- The F1-score for "not clear enough" measures algorithm performance and accuracy. The minority class's F1-score, "not clear enough," may indicate model uncertainty. Binary text clarity classification causes class imbalance. Maximising this category's F1-score helps the model discover text clarity concerns, the task's aim.
- Initial research may benefit from RNNs or CNNs for text categorization. The models can understand plain language and find complex textual patterns. Feature engineering, data augmentation, and hyperparameter optimization boost performance. Advanced algorithms and optimisation may increase text clarity, categorization, and durability.

## **6. Self-Reflection**

I designed the machine learning model, prepared the data, and ensured ethical research. Selecting assessment measures and analysing data is crucial. I think sophisticated methods and hyperparameter optimisation can enhance the model. My future work will concentrate on ethical problem-solving.

## References

- Abiola, O., Abayomi-Alli, A., Tale, O.A., Misra, S. and Abayomi-Alli, O. 2023, "Sentiment analysis of COVID-19 tweets from selected hashtags in Nigeria using VADER and Text Blob analyser", *Journal of Electrical Systems and Information Technology*, vol. 10, no. 1, pp. 5.
- Bashynska, I., Sarafanov, M. and Manikaeva, O. 2024, "Research and Development of a Modern Deep Learning Model for Emotional Analysis Management of Text Data", *Applied Sciences*, vol. 14, no. 5, pp. 1952.
- Chen, R., Lee, S. and Hu, C. 2023, "Digitalization Improves Enterprise Performance: New Evidence by Text Analysis", *Sage Open*, vol. 13, no. 2.
- Ellahi, A., Ain, Q.U., Hafiz, M.R., Hossain, M.B., Csaba Bálint Illés and Rehman, M. 2023, "Applying text mining and semantic network analysis to investigate effects of perceived crowding in the service sector", *Cogent Business and Management*, vol. 10, no. 2.
- Gao, Y., Wang, J., Li, Z. and Peng, Z. 2023, "The Social Media Big Data Analysis for Demand Forecasting in the Context of Globalization: Development and Case Implementation of Innovative Frameworks", *Journal of Organizational and End User Computing*, vol. 35, no. 3, pp. 1-15.
- Ghafoor, A., Ali, S.I., Sher, M.D., Kastrati, Z., Shaikh, S. and Batra, R. 2023, "SentiUrdu-1M: A large-scale tweet dataset for Urdu text sentiment analysis using weakly supervised learning", *PLoS One*, vol. 18, no. 8.
- Hu, X., Mar, D., Suzuki, N., Zhang, B., Peter, K.T., Beck, D.A.C. and Kolodziej, E.P. 2023, "Mass-Suite: a novel open-source python package for high-resolution mass spectrometry data analysis", *Journal of Cheminformatics*, vol. 15, no. 1, pp. 87.
- Li, J., Xu, T., Gu, X., Lin, J., Li, M., Tao, P., Dong, X., Yao, P. and Shao, M. 2023, "Scene clusters, causes, spatial patterns and strategies in the cultural landscape heritage of Tang Poetry Road in Eastern Zhejiang based on text mining", *Heritage Science*, vol. 11, no. 1, pp. 212.