**A blue and white sign with white text

Description automatically generated with medium confidenceCapstone project phase A**

**Cross-Sentiment Analysis of Literature Sources using BERT**

**23-2-R6**

Advisors:

Dr. Renata Arvos and Prof. Zeev Volkovich

Authors:

Omer Asus 205923717

Alon Modin 316567585

Table of Contents

[1. Abstract: 3](#_Toc137476288)

[2. Introduction. 4](#_Toc137476289)

[3. Background and related works: 5](#_Toc137476290)

[3.1 Machine Learning 5](#_Toc137476291)

[3.2 Preprocessing 6](#_Toc137476292)

[3.3 Training 6](#_Toc137476293)

[3.4 Transformers 6](#_Toc137476294)

[3.5 Encoders 7](#_Toc137476295)

[8](#_Toc137476296)

[3.6 Bert 8](#_Toc137476297)

[9](#_Toc137476298)

[3.7 Transfer Learning 9](#_Toc137476299)

[3.8 CNN 10](#_Toc137476300)

[4. Research Process 11](#_Toc137476301)

[4.1 PROBLEM DEFINITION 11](#_Toc137476302)

[4.2 Flow chart 12](#_Toc137476303)

[4.3 Data Collection 12](#_Toc137476304)

[4.4 Algorithm Development 12](#_Toc137476305)

[4.5 Experimental Setup 13](#_Toc137476306)

[4.6 Evaluation Metrics 13](#_Toc137476307)

[4.7 Result Analysis and Interpretation 13](#_Toc137476308)

[4.8 Limitations and Future Directions 13](#_Toc137476309)

[4.9 UML class diagram 14](#_Toc137476310)

[5. Graphical User Interface 14](#_Toc137476311)

[5.1 First Starting Page 14](#_Toc137476312)

[5.2 Second screen for selecting the original author documents and selecting the tested document 15](#_Toc137476313)

[5.3 Third screen to select the type of BERT you would like to work with 15](#_Toc137476314)

[5.4 A screen to select the size of the chunk you would like to work with 15](#_Toc137476315)

[5.5 A screen that appears when the tested document belongs to the same author 16](#_Toc137476316)

[16](#_Toc137476317)

[5.6 A screen that appears when the test was successful and you choose to see the graphs 16](#_Toc137476318)

[16](#_Toc137476319)

[5.7 A screen that appears when the tested document does not belong to the same author 16](#_Toc137476320)

[6. Evaluation/Verification Plan 17](#_Toc137476321)

[6.1 GUI test plan 17](#_Toc137476322)

[7. Work plan for part B 18](#_Toc137476323)

[8. References 19](#_Toc137476324)

## 1. Abstract:

This project presents a novel approach for aspect-level sentiment analysis in cross-domain text using BERT, a state-of-the-art deep learning model for natural language processing. The objective of aspect-level sentiment analysis is to ascertain the sentiment or opinion expressed towards specific aspects or entities within a given text. Cross-domain analysis involves applying the developed algorithm trained on one domain to another domain, enabling the model to generalize its understanding across different domains or topics. The proposed algorithm leverages the powerful language representation capabilities of BERT to capture fine-grained sentiment information. By utilizing BERT's contextualized word embeddings and attention mechanism, the algorithm achieves enhanced performance in aspect-level sentiment analysis compared to traditional methods. Experimental results on benchmark datasets demonstrate the effectiveness of the proposed approach in accurately identifying and categorizing sentiment towards various aspects across diverse domains. The findings highlight the potential of BERT-based aspect-level sentiment analysis algorithms for addressing challenges in sentiment analysis tasks and enabling domain adaptation in real-world applications.

## 2. Introduction.

Sentiment analysis, a prominent research area within natural language processing, aims to automatically identify and understand the sentiment or opinion expressed in textual data. Traditional sentiment analysis techniques often need help to capture nuanced sentiment towards specific aspects or entities mentioned in the text. However, recent advancements in deep learning models have shown promising results in addressing this challenge. One such model is BERT (Bidirectional Encoder Representations from Transformers), which has exhibited exceptional language representation capabilities.

This project focuses on applying a BERT-based aspect-level sentiment analysis algorithm for cross-domain text. The primary objective is to develop an algorithm to effectively discern and categorize sentiment towards specific aspects or entities mentioned in diverse textual data, independent of the domain or topic. By leveraging BERT's contextualized word embeddings and attention mechanism, the proposed algorithm seeks to enhance the performance of aspect-level sentiment analysis compared to conventional methodologies.

Crucial to this project is the incorporation of cross-domain analysis. Cross-domain analysis entails training the algorithm on one domain and applying it to another, enabling the model to generalize its understanding across dissimilar domains or topics. This approach holds significant potential for improved generalization and adaptation to diverse real-world scenarios that require sentiment analysis across various domains.

The primary motivation behind this project lies in addressing the limitations inherent in existing sentiment analysis techniques and exploring the efficacy of BERT-based models in aspect-level sentiment analysis. By harnessing the capabilities of BERT, this project endeavors to provide a more accurate and comprehensive understanding of sentiment towards different aspects within textual data, regardless of the domain or topic. The findings of this project bear relevance to numerous applications, including social media analysis, customer reviews, and market research, where capturing nuanced sentiment information is critical for informed decision-making processes.

In the subsequent sections, we will delve into the methodology, experimental setup, and evaluation of the proposed BERT-based aspect-level sentiment analysis algorithm. The obtained results from benchmark datasets will be thoroughly discussed, showcasing the effectiveness and potential applications of the algorithm. Finally, we will conclude with a comprehensive discussion of the implications and future avenues of research in the field of sentiment analysis and natural language processing.

## 3. Background and related works:

Sentiment analysis, or opinion mining, has emerged as a crucial task in natural language processing. It involves determining the sentiment or opinion expressed in textual data, allowing organizations and researchers to gain insights into public perception, customer feedback, and market trends. Traditional sentiment analysis techniques typically classify the overall sentiment of a document or sentence as positive, negative, or neutral. However, these approaches often need to capture the sentiment associated with specific aspects or entities mentioned in the text. Sentiment analysis

Sentiment analysis, also known as opinion mining, is a computational approach used to analyze and extract subjective information from text, with the aim of determining the sentiment or emotional tone conveyed by the author. It involves techniques that automatically identify and classify the sentiment expressed in textual data as positive, negative, or neutral. Sentiment analysis methods typically rely on natural language processing and machine learning algorithms to process and interpret textual content, considering linguistic patterns, contextual cues, and semantic information. By analyzing sentiment, sentiment analysis provides valuable insights for applications such as brand monitoring, market research, social media analysis, and customer feedback analysis, contributing to better understanding public opinion and sentiment trends.

Aspect-level sentiment analysis addresses this limitation by analyzing sentiment towards individual aspects or entities within a text. By providing a more fine-grained understanding of sentiment, aspect-level sentiment analysis enables more nuanced insights into customer preferences, product features, and public sentiment toward specific entities. This level of granularity is especially valuable in domains such as product reviews, social media discussions, and customer feedback analysis.

### 3.1 Machine Learning

Machine learning is a branch of artificial intelligence that involves the development of algorithms and models capable of automatically learning from data without explicit programming. By analyzing large datasets, these algorithms identify patterns and relationships, enabling them to make predictions, classify objects, and gain insights from complex and unstructured data. Machine learning has widespread applications in areas such as image recognition, natural language processing, and predictive analytics, driving advancements in technology and scientific research.

~~A picture containing screenshot, circle, diagram, line

Description automatically generated~~

### 3.2 Preprocessing

Preprocessing, in the context of data analysis and machine learning, refers to the set of techniques and operations applied to raw data before it is used for further analysis or model training. It involves cleaning, transforming, and organizing the data to improve its quality and compatibility with subsequent tasks. Preprocessing steps may include removing duplicate or irrelevant records, handling missing values, normalizing, or standardizing numerical features, encoding categorical variables, and reducing dimensionality. The goal of preprocessing is to enhance the data's suitability for analysis, mitigate potential biases or inconsistencies, and improve the performance and interpretability of machine learning models. By addressing data quality issues and optimizing the representation of the data, preprocessing plays a critical role in ensuring reliable and meaningful results in data-driven research and decision-making processes.

### 3.3 Training

Training, in the context of machine learning, refers to the process of optimizing a model's parameters or weights using labeled training data. It involves presenting the model with a set of input examples along with their corresponding target outputs and iteratively adjusting the model's parameters to minimize the difference between predicted outputs and the true outputs. Through an optimization algorithm, such as gradient descent, the model learns from the training data and updates its internal representations to improve its performance on the given task. Training involves iteratively adjusting the model's parameters by computing gradients and updating them based on the chosen optimization algorithm. The objective of training is to enable the model to generalize well to unseen data, effectively capturing the underlying patterns and relationships within the training examples.

### 3.4 Transformers

A transformer is a deep learning architecture designed explicitly for sequence-to-sequence tasks, such as machine translation and natural language processing. It utilizes self-attention and multi-head attention mechanisms to capture relationships and dependencies between elements within a sequence. Unlike recurrent neural networks (RNNs) that process sequences sequentially, transformers operate in parallel, making them more efficient for long-range dependencies. Transformers are constructed within the encoder-decoder framework, where the encoder learns representations of the input sequence, and the decoder generates the output sequence based on those representations. The transformer architecture has shown significant advancements in various natural language processing tasks, achieving state-of-the-art results and enabling efficient modeling of long-range dependencies in sequence data.

### 3.5 Encoders

In the context of deep learning and neural networks, encoders refer to the components responsible for transforming input data into a compressed or abstract representation. Encoders process the input data through a series of layers, typically consisting of non-linear activation functions, to extract and encode relevant features or patterns. This transformation reduces the dimensionality of the input while retaining important information. Encoders play a crucial role in dimensionality reduction, feature extraction, and representation learning tasks. By capturing and encoding salient information from the input data, encoders facilitate subsequent stages of processing, such as classification, clustering, or generating output sequences. Their effectiveness lies in their ability to learn hierarchical and abstract representations that capture meaningful features from the input, enabling improved performance and generalization in complex machine learning tasks.

To understand how encoders work within the framework of deep learning and neural networks, let's delve into the steps involved:

1. Input Data: The encoder receives raw input data, which can be in different formats such as text, images, or audio. The data is typically preprocessed to ensure it is in a suitable format for further analysis.

2. Feature Extraction: The first step in the encoder involves extracting relevant features from the input data. This is achieved by applying transformations, such as convolutions or recurrent operations, which capture low-level and high-level patterns in the data. For example, in computer vision, convolutional neural networks (CNNs) are commonly used for feature extraction.

3. Nonlinear Mapping: After feature extraction, the encoder applies nonlinear mapping functions to capture complex relationships between the extracted features. These functions introduce nonlinearity into the model, allowing it to learn more sophisticated representations of the data.

4. Dimensionality Reduction: Encoders aim to compress the extracted features into a lower-dimensional representation. This step reduces the complexity of the data while retaining its important characteristics. Dimensionality reduction techniques, such as autoencoders or pooling operations, are commonly employed for this purpose.

5. Encoding: The encoder further processes the reduced-dimensional representation to generate an encoded vector or embedding. This vector is a condensed representation of the input data that captures its salient information. The encoding process can involve various operations, such as fully connected layers, recurrent layers, or self-attention mechanisms in transformer-based models.

6. Training: The encoder is trained using a large dataset through a process called backpropagation. During training, the encoder adjusts its parameters iteratively to minimize a predefined loss function. This optimization process ensures that the encoder learns to generate meaningful encodings that are useful for the intended task.

7. Transfer Learning: Encoders often benefit from transfer learning, where pre-trained encoders from large-scale datasets are fine-tuned on specific tasks or domains. By leveraging knowledge

### A picture containing screenshot, rectangle, line, design Description automatically generated

### 3.6 Bert

BERT (Bidirectional Encoder Representations from Transformers) is an influential natural language processing (NLP) algorithm that has significantly advanced the field of deep learning. BERT is based on the Transformer architecture and has revolutionized language understanding by effectively capturing contextual information bidirectionally. Unlike earlier models that only considered unidirectional representations, BERT introduced a unique pre-training objective called the masked language model.

The pre-training phase of BERT involves exposing the model to a vast amount of text data. During this process, BERT randomly masks certain tokens in the input and learns to predict them based on the surrounding context. By predicting the masked tokens, BERT is compelled to understand and represent the relationships between different words within a sentence. Importantly, BERT considers both the preceding and succeeding words, enabling it to leverage the entire context to make accurate predictions.

In addition to the masked language model, BERT employs another pre-training objective called the next sentence prediction. This objective allows BERT to comprehend sentence-level relationships by training the model to determine if two sentences appear consecutively in the original text or if they are randomly paired. By training on this task, BERT gains the ability to understand and capture dependencies between sentences, which is crucial for many downstream NLP tasks.

Once the pre-training phase is complete, BERT can be fine-tuned for specific tasks by adding task-specific layers and optimizing the model on task-specific datasets. Fine-tuning involves training BERT on labeled data for tasks such as text classification, named entity recognition, or question answering. By incorporating task-specific information and adjusting the model's parameters, BERT can adapt its learned representations to perform well on the specific target task.

BERT's success lies in its ability to learn deep contextual representations from large-scale unlabeled data and subsequently transfer this knowledge to a wide range of downstream tasks. By leveraging the bidirectional context and employing the masked language model and next sentence prediction objectives, BERT captures rich semantic relationships and achieves state-of-the-art performance on various NLP benchmarks. BERT has had a significant impact on the academic community, inspiring subsequent research in contextual language representation models and facilitating advancements in language understanding across multiple domains.

### 

### 3.7 Transfer Learning

Transfer learning, in the context of machine learning, refers to the technique of leveraging knowledge gained from one task or domain to improve performance on another related task or domain. By transferring learned representations, weights, or parameters from a pre-trained model to a new task, transfer learning aims to overcome data scarcity, reduce computational requirements, and enhance generalization. This process involves fine-tuning or retraining the pre-trained model on the target task or domain, allowing it to adapt and specialize its learned knowledge. Transfer learning has demonstrated remarkable success across various domains, including computer vision, natural language processing, and audio analysis, enabling efficient and effective utilization of pre-existing knowledge to improve the performance of models on new and related tasks.

### 3.8 CNN

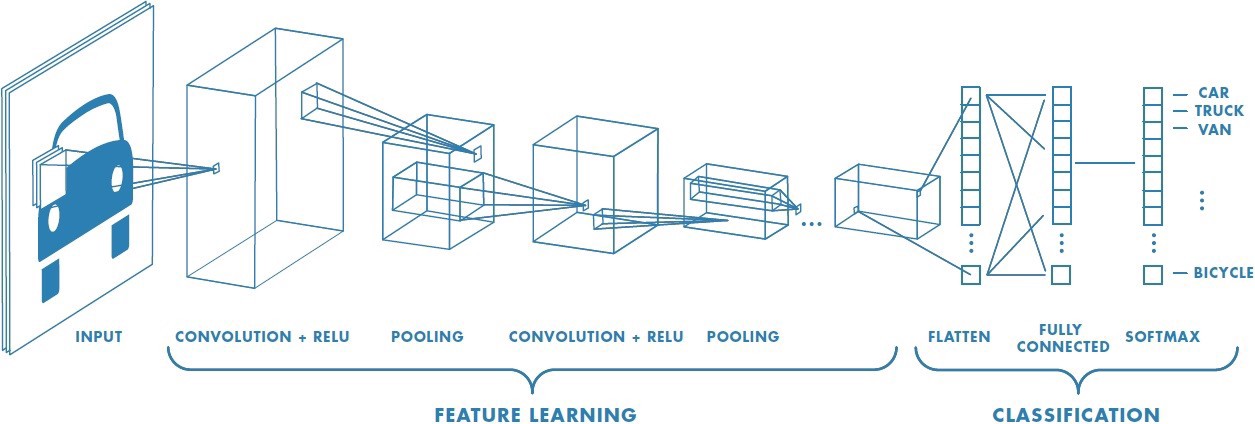
A Convolutional Neural Network (CNN) is a deep learning algorithm widely used for image classification and recognition tasks. It is designed to automatically learn and extract meaningful features from input images. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

At the core of a CNN are convolutional layers, which apply a set of learnable filters to input images. Each filter scans the input image with a small receptive field, performing element-wise multiplication and aggregation to produce a feature map. This process captures local spatial patterns, such as edges, corners, and textures. Multiple filters are used to detect various features simultaneously.

Pooling layers follow the convolutional layers and down sample the feature maps, reducing their spatial dimensions while preserving important information. Pooling helps in achieving translation invariance by focusing on the most salient features within a region. The most common pooling operation is max pooling, which selects the maximum value within each pooling region.

The final layers of a CNN typically include fully connected layers, which are used for classification. These layers connect all the extracted features to a classifier that outputs the predicted class probabilities. The parameters of the CNN, including the filter weights and biases, are learned through a process called backpropagation, where the network adjusts its internal representations to minimize the difference between predicted and true labels.

In summary, CNNs use convolutional layers to extract relevant features from input images, pooling layers to reduce spatial dimensions, and fully connected layers for classification. The network learns to recognize patterns and objects by adjusting its parameters based on the provided training data.



## 4. Research Process

### 4.1 PROBLEM DEFINITION

In this research project, our main objective is to identify text writers by analyzing their unique writing style. To achieve this, we adopt a methodology that involves converting the input texts into short portions, like tweets, and considering their characteristic conciseness and tendency to express opinions.

The main idea is to employ the sentiment analysis methodology. Since “tweets” generated from the texts can convey both positive and negative sentiments, we encounter the challenge of sentiment analysis. Sentiment analysis involves determining the underlying sentiment expressed within a piece of text. In our connotation “positive” and “negative” means different authorships. To address this challenge, we intend to employ a BERT-based aspect-level sentiment analysis algorithm for cross-domain text. BERT, a pre-trained language model, is known for its ability to comprehend contextual information and capture intricate language nuances.

By utilizing BERT as the backbone of our approach, we aim to train a specialized dataset tailored specifically for sentiment analysis in the mentioned manner. This dataset will contain diverse examples of tweet-like text, enabling the model to learn and generalize patterns associated with different style categories. Through this training process, we expect to develop a model that can accurately discern and classify the tweet-like text expressed by writers.

By leveraging the power of BERT and employing a dataset tailored for sentiment analysis, we anticipate achieving precise and nuanced identification of styles based on their sentiments expressed within tweet-like content. This approach has the potential to contribute valuable insights into understanding individual writing styles and their associated sentiments, facilitating various applications such as author profiling, opinion mining, and social media analysis.

### 4.2 Flow chart

A picture containing text, font, screenshot, poster

Description automatically generated

### 4.3 Data Collection

The research commenced with the collection of suitable datasets for training and evaluating the proposed algorithm. Datasets covering a diverse range of domains were sought to ensure comprehensive coverage of cross-domain sentiment analysis scenarios. Various publicly available datasets, such as customer reviews, social media data, and product feedback, were considered. Attention was given to ensure adequate representation of different domains and aspect-level sentiment annotations for training the algorithm.

### 4.4 Algorithm Development

Building upon the existing BERT-based models for aspect-level sentiment analysis, the proposed algorithm was developed. The algorithm leveraged BERT's contextualized word embeddings and attention mechanism to capture fine-grained sentiment information towards specific aspects or entities mentioned in the text. Modifications and enhancements were made to the existing models to address the cross-domain sentiment analysis requirements. Consideration was given to optimize the algorithm's performance, computational efficiency, and generalizability.

### 4.5 Experimental Setup

To evaluate the performance of the proposed algorithm, a comprehensive experimental setup was established. The collected datasets were divided into training, validation, and testing sets. The training set was used to fine-tune the BERT model and optimize the algorithm's parameters. The validation set was employed to perform hyperparameter tuning and model selection. Finally, the testing set was utilized to assess the algorithm's performance in aspect-level sentiment analysis across different domains.

### 4.6 Evaluation Metrics

To measure the algorithm's effectiveness, several evaluation metrics were employed. Precision, recall, and F1 score were calculated to assess the accuracy of aspect identification and sentiment prediction. Additionally, other metrics, such as accuracy, were used to evaluate the overall performance of the algorithm. These metrics provided insights into the algorithm's ability to correctly identify and classify sentiment towards various aspects across diverse domains.

### 4.7 Result Analysis and Interpretation

The obtained results will be thoroughly analyzed and interpreted to draw meaningful conclusions about the proposed algorithm's effectiveness. Statistical analysis and visualizations are employed to identify trends, strengths, and limitations of the algorithm. The implications of the results are discussed in the context of cross-domain sentiment analysis and the potential real-world applications of the proposed algorithm.

### 4.8 Limitations and Future Directions

By following this rigorous research process, the project aimes to provide comprehensive insights into the effectiveness and applicability of the proposed BERT-based aspect-level sentiment analysis algorithm for cross-domain text. The findings contributed to the existing body of knowledge in sentiment analysis and paved the way for future advancements in this domain.

### A screenshot of a computer Description automatically generated with medium confidence4.9 UML class diagram

## 5. Graphical User Interface

### 5.1 First Starting Page

### A computer screen shot of a stack of books Description automatically generated with low confidence5.2 Second screen for selecting the original author documents and selecting the tested document

### 5.3 Third screen to select the type of BERT you would like to work with

A computer screen shot of a stack of books

Description automatically generated with low confidence

### A computer screen shot of a stack of books Description automatically generated with low confidence5.4 A screen to select the size of the chunk you would like to work with

### 5.5 A screen that appears when the tested document belongs to the same author

### A screenshot of a computer Description automatically generated with low confidence

### 5.6 A screen that appears when the test was successful and you choose to see the graphs

### A picture containing text, screenshot, font, design Description automatically generated

### A picture containing text, screenshot, businesscard, font Description automatically generated5.7 A screen that appears when the tested document does not belong to the same author

## 6. Evaluation/Verification Plan

To evaluate the system, a rigorous assessment is required to determine its effectiveness. This assessment involves verifying the trained network's performance on a set of documents that are known to belong to a specific author. By inputting an unknown document into the system, we aim to ascertain whether the system will yield a positive response if the document indeed belongs to the author, or a negative response if it does not. To conduct the evaluation, we intend to train the network using a substantial corpus of documents known to be authored by a particular writer, such as Shakespeare, who is renowned for a vast literary output. Subsequently, we will curate a subset of approximately 100 documents, half of which were written by the author being tested, while the remaining 50 were composed by other authors. The evaluation will then be expanded by incorporating additional untested documents to assess the continued accuracy of the algorithm.

### 6.1 GUI test plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Test module** | **Test description** | **Expected result** |
| **1** | Home window | Click exit (“X”) | Close the window |
| **2** | Home window | Click “Start” | Open the Browser window |
| **3** | Browser window | Click exit (“X”) | Close the window |
| **4** | Browser window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **5** | Browser window | Click the first “Browse” button | Open a file selection dialog for the user to browse and select the author data set file |
| **6** | Browser window | Click the second “Browse” “button | Open a file selection dialog for the user to browse and select the text file. |
| **7** | Browser window | Click “Next” button | Enable the Next button only when two files (a text file and an author data set file) are selected. Clicking Next should proceed to the BERT size window |
| **8** | BERT size window | Click exit (“X”) | Close the window |
| **9** | BERT size window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **10** | BERT size window | Click “Base” Button | Selecting the Base button should enable it, and the Large button should remain unselected |
| **11** | BERT size window | Click “Large” Button | Selecting the Large button should enable it, and the Base button should remain unselected |
| **12** | BERT size window | Click “Next” button | Enable the Next button after selecting either the Base or Large button. Clicking Next should proceed to the Chunk size window |
| **13** | Chunk size window | Click exit (“X”) | Close the window |
| **14** | Chunk size window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **15** | Chunk size window | Select Chunk Size from Drop List | The user should be able to select a desired chunk size from the drop-down list. |
| **16** | Chunk size window | Click “Run” button | After selecting the chunk size and clicking the run button. If successful, a "Test Succeeded" window will open, otherwise, a "Test Failed" window will open. |
| **17** | Test Succeeded window | Click exit (“X”) | Close the window |
| **18** | Test Succeeded window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **19** | Test Succeeded window | Click “Open graphs” button | Clicking the Open Graphs button open the Graph window |
| **20** | Test Succeeded window | Click “Finish” button | Clicking the Finish button should close the current window and terminate the application |
| **21** | Graph window | Click exit (“X”) | Close the window |
| **22** | Graph window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **23** | Graph window | Click “Finish” button | Clicking the Finish button should close the current window and terminate the application and pop-up save graph option |
| **24** | Test Failed window | Click exit (“X”) | Close the window |
| **25** | Test Failed window | Click Menu button | Display a drop-down menu with relevant options for selection. Selecting an option should trigger the corresponding action |
| **26** | Test Failed window | Click “Start over” button | Clicking the Start Over button open "BERT Size" window, resetting any progress made in the current window. |
| **27** | Test Failed window | Click “Finish” button | Clicking the Finish button should close the current window and terminate the application |

## 7. Work plan for part B

In this phase of the project, the comprehensive understanding and research conducted will be leveraged to meticulously implement an algorithm for identifying literature sources, capitalizing on the principles and techniques elucidated earlier. The objective is to develop a functional algorithm that facilitates text writer detection by employing a combination of state-of-the-art methodologies, including Transformers, Encoders, and specifically, BERT. By harnessing the power of these advanced techniques, the algorithm will be designed to effectively analyze textual data and accurately identify the authors of various literary sources. This implementation will encompass the integration of key components, such as pre-processing techniques, feature extraction, and model training, ensuring the robustness and reliability of the algorithm. Through this endeavor, we aim to contribute to the field of text writer detection by developing a sophisticated and dependable solution that capitalizes on the advancements in transformer-based models and their associated components.

## 8. References

[1]: Liu, N., & Zhao, J. (2022). A BERT-Based Aspect-Level Sentiment Analysis Algorithm for Cross-Domain Text. Available at: <https://www.hindawi.com/journals/cin/2022/8726621> (Accessed: May 4, 2023).

[2]: Vaswani, A. *et al.* (2017) *Attention is all you need*, *arXiv.org*. Available at: <https://arxiv.org/abs/1706.03762> (Accessed: May 4, 2023).

[3]: Devlin, J. et al. (2019). Bert: Pre-training of deep bidirectional Transformers for language understanding. arXiv.org. Available at: <https://arxiv.org/abs/1810.04805> (Accessed: May 4, 2023).

[5]: Pérez-Rosas, V. *et al.* (2017) *Automatic Detection of Fake News*, *arXiv.org*. Available at: <https://arxiv.org/pdf/1708.07104v1.pdf> (Accessed: May 4, 2023).

[6]: Albawi, S. *et al.* (2017) *Understanding of a Convolutional Neural Network*. Available at: <https://www.researchgate.net/publication/319253577_Understanding_of_a_Convolutional_Neural_Network> (Accessed: May 4, 2023).

[7]: Chowdhary, K.R. (2020) *Fundamentals of Artificial Intelligence*. In Natural Language Processing, pp 603-649 (Accessed: May 4, 2023).

[8]: Jordan, M.I and Mitchell, T.M (2015) *Machine learning: Trends, perspectives, and prospects*. Available at: <https://www.cs.cmu.edu/~tom/pubs/Science-ML-2015.pdf> (Accessed: May 4, 2023).

[9]: Lin, T. *et al.* (2022) A survey of transformers. Available at: <https://www.sciencedirect.com/science/article/pii/S2666651022000146> (Accessed: May 4, 2023).

[10]: Towardsdatascience. (2018, November 10). BERT Explained: State of the art language model for NLP. From <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

[11]: Hugging Face. (2022, March 2). BERT 101. From <https://huggingface.co/blog/bert-101>