Assignment Specifications

Aims	• Enable students to analyse and employ appropriate Natural Language Processing (NLP) techniques to solve problems and design intelligent systems. • Enable students to use relevant tools and technology, such as Python programming, to develop intelligent computer programs.
Learning Outcomes Assessed	• Discuss the concepts and importance of natural language in real-world scenarios (C2, PLO1). • Practice the NLP algorithm using the features and capabilities of NLP software (P3, PLO3).
Tutor	Noor Aida binti Husaini Kathleen Tan Swee Neo Nicholas Lim Jun Jie
Outline of Problem	For your assignment, you should critically evaluate current technologies, and then propose a project for the selected NLP topic, and implement an NLP solution to solve the problem in the proposed project by using Python/any suitable tools/ Framework.
Details	• The Assignment consists of **TWO (2)** parts, please refer to **Project Details: Part 1** and **Project Details: Part 2**. Both parts are related. For Part 2, you are required to develop the program using Python. • Group yourself into a team of **TWO OR THREE (2-3)** members. You are given **TWO (2) OPTIONS** for research areas as listed below. Your group shall select one of them for this assignment purpose. • Please note that the following are the basic requirements of the assignment. Fulfilling the requirements may lead you to an **Average** or **Good** grade. Achieving an **Excellent** grade requires **extra effort**, such as learning new skills, introducing new ideas, and complex NLP algorithms, demonstrating the ability to process big data, and/or producing excellent reports with working prototypes. • Your group is expected to produce ideas that originated from the members, but not to take the work or an idea of someone else (including the Web) and pass it off as your own. **NO GROUP IS ALLOWED to share the same idea**, i.e., each group must propose a unique title or solution.
Type	**Type A:** Develop one approach/system, but each member takes on different phases (pre-processing, main processing, post-processing, user interface (not necessarily GUI)). **Type B:** The group members developed different approaches to achieve the same goal and then compared them.

Title	**1. Interactive Chat Application for Conversations with PDF or Document Content Using a Language Model (LLM)** You are required to design and implement a chat application that utilises a Language Model (LLM), such as GPT-3.5, to facilitate conversations between users and the content of a PDF or a document. The application should have the following functionalities: a. User Interface: Create a user-friendly interface that allows users to upload a PDF or a document file. b. Text Extraction: Implement a mechanism to extract the text content from the uploaded file. c. Language Model Integration: Integrate the LLM into the chat application to process the extracted text and generate meaningful responses based on user queries or inputs. d. Chat Functionality: Develop the chat functionality that enables users to have interactive conversations with the content of the PDF or document. Users should be able to ask questions, provide inputs, and receive relevant responses based on the extracted text, with the LLM providing natural and contextually appropriate replies. e. User Experience: Ensure that the chat application provides a smooth and intuitive user experience. Consider factors such as response time, error handling, and overall usability, while utilising the capabilities of the LLM to enhance user interactions. f. Testing and Validation: Conduct thorough testing of the application to ensure its functionality and accuracy. Validate the responses generated by the LLM against the original content of the PDF or document. g. Documentation: Provide clear and comprehensive documentation that includes instructions on how to use the application and details about the implementation, including the technologies and libraries used. Explain the integration of the LLM and how it contributes to the chat functionality. **Note:** You can choose any programming language and framework of your preference for this assignment. Make sure to justify your choices in the documentation, including the selection and configuration of the LLM.

2. Sentiment Analysis on Out-Of-Vocabulary (OOV) Malaysia Rojak Language

You are required to design and implement a sentiment analysis system that can accurately analyse the sentiment of text written in out-of-vocabulary (OOV) Malaysia Rojak language. The application should have the following functionalities:

a. Data Collection: Collect a sufficient amount of Malaysia Rojak language data from various sources, including social media, online forums, and news articles. This dataset will serve as the training data for your sentiment analysis model.

b. Preprocessing: Develop preprocessing techniques specifically tailored for the Malaysia Rojak language. This may include tokenisation, normalisation, and handling of linguistic nuances specific to the language.

c. Model Selection: Choose an appropriate sentiment analysis model or algorithm that can effectively handle the OOV nature of the Malaysia Rojak language. Consider techniques such as transfer learning, domain adaptation, or multilingual models.

d. Training and Evaluation: Train your selected model using the collected Malaysia Rojak language dataset. Split the dataset into training and evaluation sets to assess the performance of your model accurately. Use appropriate evaluation metrics such as Human Level Performance (HLP) with/without survey.

e. Testing: Evaluate the performance of your trained model on a separate test dataset consisting of Malaysia Rojak language text samples. Measure the model's accuracy in predicting sentiment labels for the given texts.

f. Error Analysis and Improvements: Conduct an error analysis to identify common mistakes or challenges faced by your model in analysing the sentiment of Malaysia Rojak language. Propose improvements or refinements to enhance the model's accuracy and robustness.

g. Documentation: Provide clear and comprehensive documentation that includes details about the data collection process, preprocessing techniques, model selection, training methodology, evaluation results, error analysis, and proposed improvements.

Note: You can choose any programming language and existing libraries or frameworks for sentiment analysis and text processing. Justify your choices in the documentation, specifically regarding the handling of OOV Malaysia Rojak language.

Submission Deadlines	Submit Prototype Source Code and Documentation by: **Part 1: Documentation: Week 12, Sunday, before 11.59 pm** **Part 2: Source Code: Week 12, Sunday, before 11.59 pm** *Please submit it to Google Classroom. Late submissions will be penalised. A demo session to present the prototype is required.*
Contribution	**Part 1: Documentation - 40%** **Part 2: Source Code - 60%** Part 1: Each member must submit their documentation through Google Classroom. Part 2: Only one member from each group should submit the source code on behalf of the group through Google Classroom.
Academic Integrity and Plagiarism	Your work must have originality, i.e., **DO NOT** copy or refer to other group(s). You may only work with your team member(s) to produce the solution of this assignment. You **MUST NOT** share with nor refer to any part of the assignment (including the code) of anyone else except your team member(s) and your tutor. Before submitting your assignment, please ensure that you have complied with **TAR UMT Plagiarism Policy**. Any cheating attempt to cheat, plagiarism, collusion, and any other attempts to gain an unfair advantage in assessment will cause the students concerned to be penalised. **IMPORTANT:** Students found to be dishonest are liable to disciplinary action.
Late Submission	Late submission without a valid reason will **NOT** be tolerated. For late submissions, there will be a reduction of total marks: • Late 1 to 3 days after the deadline of submission: minus 10 marks • Late 4 to 7 days after the deadline of submission: minus 20 marks • Late more than 7 days after the deadline of submission: 0 marks In certain circumstances, a student may be allowed to submit the assignment late with a valid reason. S/he must contact the tutor at least one week before the assignment is due. The tutor will evaluate whether the circumstance warrants submitting the assignment late, but no guarantee that the students will not be penalised. Failing to submit the reports & code will lead to failure of the coursework. **Note:** For Extenuating Mitigating Circumstances (EMC) cases, students need to apply the Leave of Application from the FOCS office as a record where proof of MCs, Death Certificate. SPM Examination, MUET Examination, etc. is given.

Project Details: Part 1 - Proposal Writing

Introduction	Your task for this part of the assignment is to **identify a problem** for the selected NLP topic, perform a **literature review,** and propose your respective **NLP solution(s)** that helps in solving the problem in the proposed project.
What to hand in?	Each student is expected to **produce individual work** regarding the specified items and must not repeat what others have done. The report shall be completed using the given Google Doc template in the Google Classroom.
Format for Deliverable	For your proposal, you are required to complete the report using the Google Docs template available in the Google Classroom. You must **turn in the report** to Google Classroom before the stipulated deadline. Otherwise, the late submission penalty will be applied.

Project Details: Part 2 - Prototype Development

Introduction	Your task for this assignment is to implement the NLP solution using Python or any other relevant tools, perform testing and evaluation on the system, and finally present the work to your tutor.
What to hand in?	Submit all your source code to the Google Classroom submission page. Each team member is required to do a presentation of the work and Q&A session on Week 13 or Week 14 based on the arrangement with the tutor.
Format for Deliverable	**Compress** the entire source code using zip format. Submit the source code individually.