**Exercise Project (EP) Proposal**

**I. Executive Summary**

A.L.I.C.E. or Analyzing Language Interface Created for Everyone is a text summarizer, that accepts a raw text body or .txt file and converts the key information into useful visualization tools for the user. Later advancements

- Docker

Flask UWSGI

React EngineX

**II. Objectives (Learning/Project goals)**

**III. Background**

Define the scope/domain

- Granularity of analysis: News article (one page), book (multiple chapters), journal (one document)

**IV. Outline**

1. **~~Text Summarizer?~~**
2. **Topic Modelling**

* Topic modelling is a process to automatically identify topics that are present in the text.
* With the increasing amount of data in the recent years, it is highly difficult to process the vast amount of information and retrieve relevant and desired information.
* Hence, we aim to use LDA to mine through the text and fetch the key idea of the text via listing the top 10 most frequent occurrence.
* We plan to use either Gensim or Sklearn as our tool for machine learning, depending on their accuracy level.
* Ultimately, these most frequent words would be displayed in the format of a word cloud that enables users to be able to understand the topics at a glance.

1. **Named Entity Recognition**

* In a text document, we want to understand what the key entities are, which are particular terms that represent real-world objects, such as people, organizations, and locations. We plan to utilize named entity recognition, also known as entity extraction, which identifies and segments named entities in a text document and classifies them under various predefined categories.
* We plan to use either the nltk or spacy NER model for the NER process and our final output would be in the form of a network diagram or table which displays the key entities and their respective categories
* Some challenges we would face would be training and updating the NER models to include more entities under preexisting categories given that we are using a fictional storybook, ‘Alice in Wonderland’, and to extend the NER model to include new entity types.

1. **Relationship Extraction**

* Relationship Extraction is the process of extracting sematic relationship from text, between two or more entities. These entities are identified using the Name Entity Recognition mentioned above. There are 5 different methods for Relationship Extraction:

1) Rule-based RE

2) Weakly Supervised RE

3) Supervised RE

4) Distantly Supervised RE

5) Unsupervised RE

* This is important in a text summarizer as it helps to map the different interaction between various entities present in the text
* We plan to use Spacy to conduct the relationship extraction. The output will ideally be in the form of a network diagram.
* One particular challenge is to accurately select which of the five different models to use. Different text such as story books and news articles might need to use different models to produce a more accurate result.

1. **Sentiment Analysis**

* Sentiment analysis, also known as opinion analysis/mining, is used to extract subjective and opinion related information, such as emotions, attitudes, and moods. This can be done at the individual sentence level, the paragraph level, or the document as a whole. For this project, we will conduct sentiment analysis at the chapter and document level. A polarity weight is given to the text, based on whether it expresses a positive, negative, or a neutral sentiment.
* There are two major techniques for sentiment analysis: supervised machine learning and unsupervised lexicon-based learning. Lexicons refer to dictionaries or vocabularies specially constructed to be used for sentiment analysis and compute sentiment without any supervised techniques. Some examples are AFINN lexicon, Bing Liu’s lexicon, MPQA subjectivity lexicon, SentiWordNet, VADER lexicon, and Pattern lexicon. The team will explore both techniques and discuss which would be more suitable.
* We plan to visualize our sentiment analysis output as a color-coded word cloud, graph or network diagram.

1. **Classifier**

* Classify documents according to category (e.g. sports, finance, geopolitics, etc.)
* Semi-supervised: features are important -> grid search (combine different features and train the model)
* Sklearn grid search function

1. **Clustering?**

* Cluster a variety of documents based on a similar characteristic
* Within the document level, cluster

**V. Proposed Timeline**

|  |  |
| --- | --- |
| **Week 1** | |
| **NER** | **Sam, Cornelius** |
| **Topic Modelling** | **Tiffany** |
| **Relationship Extraction** | **Sam, Cornelius** |
| **Sentiment Analysis** | **Sam, Cornelius, Tiffany** |
| **Week 2** | |
| **Classifier** |  |

|  |  |
| --- | --- |
| **Features** | **Visualization tool** |
| Text summarizer | Text |
| Topic Modelling | Word cloud |
| Entity extraction (e.g. date, time, location, currency) | Table format |
| Relationship detection | Network diagram |
| Sentiment Analysis | Color-coded word color, graph, network diagram |
| Classifier (Bayesian, **SVM, Linear regression, RNN,** **LSTM**) | Graph/Table |

**VI. Implementation Plan**

* ML models
* React (Frontend)
* Visualization (Frontend)
* Potential visualization libraries suited for React are Nivo, D3, Vis, and ReGraph
* Nivo seems promising to implement network graphs for entity extraction and relationship detection.
* Flask (Backend)
* Docker implementation (Backend)













