School of Computing

CA326 Year 3 Project Proposal Form

SECTION A

Project Title: Summify

Student 1 Name: Benjamin Olojo **ID Number:** 19500599

Student 2 Name: Przemyslaw Majda **ID Number:** 20505049

Staff Member Consulted for supervision: Dr. Jennifer Foster

PROJECT DESCRIPTION

Description

Summify is a web application that allows users to generate summaries of videos through Natural Language Processing. The application aims to automate the process of summarising lengthy videos through creating extractive summaries of video transcripts. Within the application, video transcripts will be fetched through user-specified URLs using the python YouTube Transcript API. These video transcripts will then be used as source material for generating extractive summaries containing key information.

The video transcripts will be summarised using a pre-trained NLP model that we will fine-tune for summarising video transcripts. Within the NLP space, conventional machine learning approaches for text summarisation are built upon summarisation data from written material rather than spoken text, leading to a decline in the performance of models when applied to NLP tasks for videos. Training the model on a benchmark dataset for video transcript segmentation and summarisation will result in more accurate summaries for videos within the application.
The fine-tuned model could then be accessed within the application through publicly available APIs such as the OpenAI API for GPT-3 or open source libraries such as the Hugging Face Transformers library for the BART model.

The application will additionally feature an Information Retrieval component that will leverage NLP Keyword Extraction libraries to identify key terms within the video transcripts and provide the user with relevant web-based resources through the python Google Search API.

The project will consist of a frontend application created using Node.js, React and Redux as well as a backend Flask application containing the NLP components for text summarisation, keyword extraction and information retrieval.
Each application will consist of API endpoints to facilitate HTTP requests between the frontend and backend of the project.

Hardware/Software Development Platform

PC, MacOS/Unix

Programming languages

- Python
- JavaScript

Programming tools

- React Library
- Redux Library
- Node.js Runtime Environment
- Flask Framework
- spaCy NLP Library
- Hugging Face
- OpenAI API

Learning Challenges

The main learning challenges of the project include becoming familiar with the React and Redux libraries, working with large datasets to fine-tune the summarisation model for video transcripts, leveraging keyword extraction algorithms to recognize key terms and utilising text summarisation evaluation metrics within testing.

Special Hardware/Software Requirements

A GPU accessible through Google Colab would be required to fine-tune a pretrained model such as BART.

Division of work

Benjamin Olojo
Backend Flask Application
Fine-tuning summarisation model
Video transcript summarisation component
Testing

Przemyslaw Majda
Frontend React application
Keyword Extraction component
Information Retrieval feature
Testing