***Abstract***

*The task was created and completed February 2020. It was aimed at the application of Information Retrieval skills into building a processing pipeline which then converts a website into a structured language.*

*The system recorded two HTML pages as inputs and processed them further using techniques such as; HTML parsing, sentence splitting, tokenization, part of speech tagging, keyword selection as well as stemming (aka morphological analysis).*

*After that, an output is generated in an index of the terms listed in the document.*

***Keywords*** *– HTML parsing, Sentence splitting, stopwords, pre-processing…*

**1.0 INTRODUCTION**

The Python Jupyter notebook framework for developing this processing pipeline was implemented.After a processing pipeline was built, two URLs were provided for conversion into structured language. Different Information Retrieval techniques will be deployed in this task.

First, the systembuilt was designed to read web pages from specified sets of URLs, resulting in properly formatted output. The next step was carried out by parsing the HTML by removing the HMTL tags to get a plain text and ensuring that meta tags were not removed in the process. Thirdly, the data was pre-processed, by normalizing the input text. The fourth step was to tag the input with the most suitable part of speech for easier processing of the result into subsequent steps. Next, words and phrases that were most useful were classified for indexing purposes in the text.

Finally, the stemming / morphological analysis was performed to write word stems into databases to avoid inflected words treated differently.

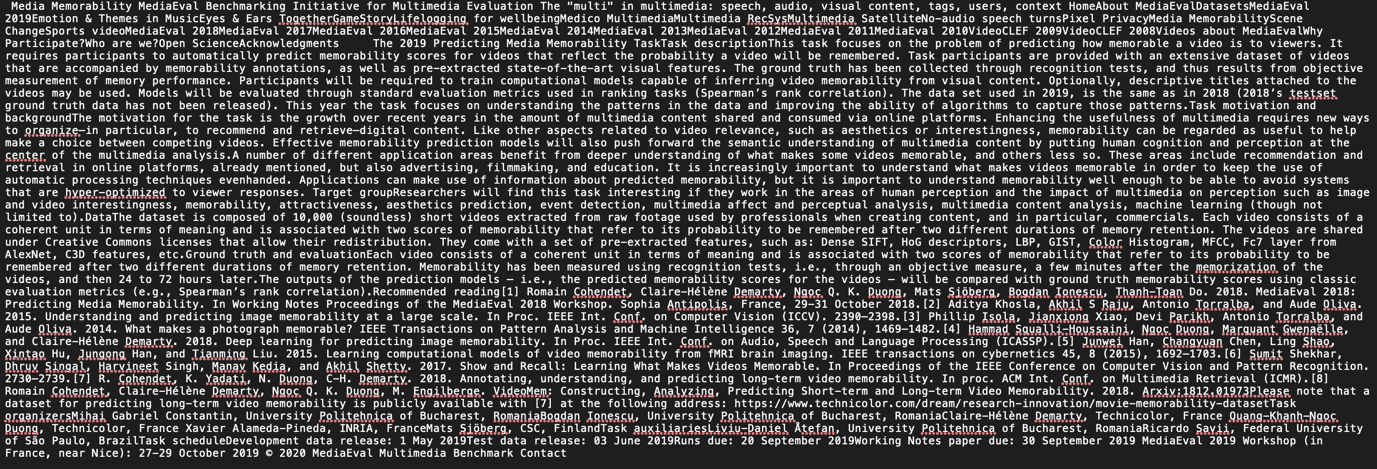
**2.0 ENGINEERING A COMPLETE SYSTEM**

For the system developed to be able to read webpages properly from a given set of URLs thereby producing suitably formatted input, the webpages were processed individually. However, for smooth running of this process, some important libraries and collections such as ‘spaCy’, ‘nltk’ etc. were loaded into the Jupyter environment.

**3.0 HTML PARSING**

HTML tags can be referred to as HTML codes that describe each structure on an HTML page, including text and image placement, hypertext links [1]. For proper text analysis, these tags were removed, by discarding texts on the webpages consisting of information that are not vital during the analysis. Nevertheless, deletion of the HTML tags would result in loss of information which include the meta tag keywords. The meta tags keywords are a type of meta tag that tells search engines the location of the heading of the webpage. To avoid the loss of this vital information, HTML parsing was then deployed to cut out the tags and extract the useful data from the webpage.

The ‘*beautifulsoup*’ property was used in creating a parse tree for parsed pages that were used to extract the data. This therefore resulted in a plain text that can be below for both webpages:



***Figure 1:*** *HTML Parsing for URL 1: ‘http://www.multimediaeval.org/mediaeval2019/memorability/’*

Figure 2: HTML Parsing for URL 2: 'https://sites.google.com/view/siirh2020/'

Description automatically generated

***Figure 2:*** *HTML Parsing for URL 2: 'https://sites.google.com/view/siirh2020/'*

**4.0 PRE-PROCESSING**

In the pre-processing step, a function was created to take in the text as input and perform operations like making the text lowercase, lemmatizing each token, removing punctuation symbols and removing stopwords. This converts the text into a format that can be properly analysed.

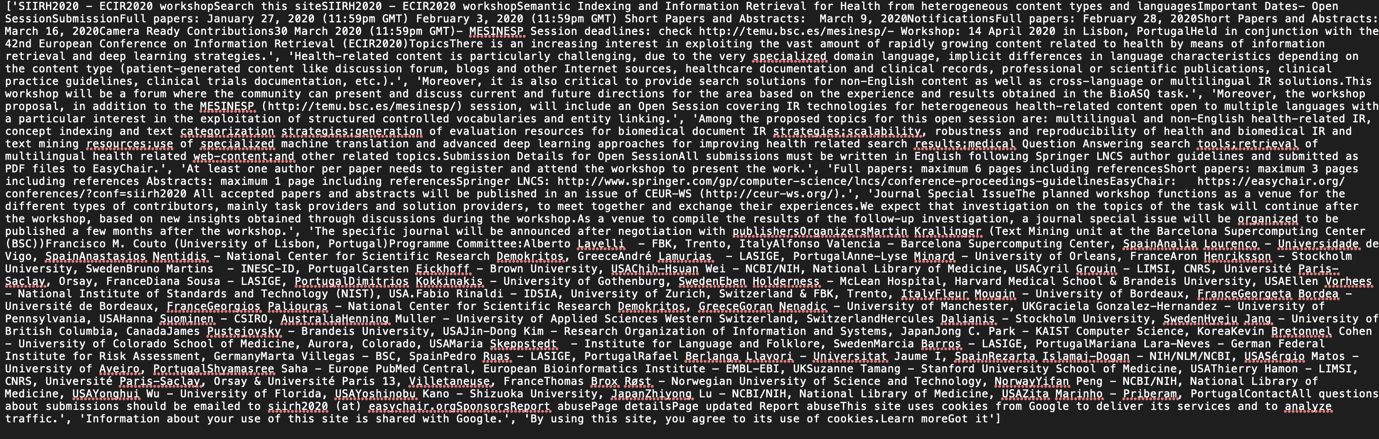
**4.1 Sentence Splitting / Detection**

This is a process in which the start and end of sentences are located in a given text [2]. That is, this helps in breaking the text into units which are linguistically important. The Spacy library was used in the sentence detection. The *sents* property was used to obtain the total number of sentences in the plain text. The figures below indicate the sentences detected from the two webpages:



FIgure 3: Sentence Detection for Preprocessed TextA

***Figure 3****: Sentence Detection for Webpage 1*



***Figure 4:*** *Sentence Detection for Webpage 2*

**4.2 Word and Sentence Tokenization**

Tokenization which is the next step after sentence splitting and detection was used in defining meaningful units (tokens) in the text, as these unit will be used for further analysis. Word and Sentence tokenization was deployed in this analysis.

**5.0 PART OF SPEECH TAGGING**

Part of Speech is used to describe how words are used in a sentence, such as: nouns, verbs etc. Here, the tokens are assigned with Part of Speech tags (POS tags) which is an attribute in spaCy [3]. The postag function has been used to assign the tags.

**6.0 SELECTING KEYWORDS**

In this analysis, for proper indexing, words and phrases which are most useful were identified for indexing terms. Frequent or stopwords were however removed to properly identify the phrases suitable as index terms, as they did not necessarily draw useful perspectives from the input text. This analysis was performed by deploying the stopwords removal technique. The tf.idf property was used as part of the selection and weighting step. The most frequent words in reszapective documents were however selected and their respective feature scores were derived for them.

**7.0 MORPHOLOGICAL ANALYSIS (NER)**

Morphological Analysis in Information Retrieval is carried out after tokenization is done on the text and meaning words/sentences have been identified. In morphological analysis, grammatical constructions such as: name of animal, place, organizations, idiomatic expressions, abbreviations, employment, concatenated words, are identified and considered/ grouped as a single word [4]. The Named- Entity Recognition task was carried out for this classification.

That is, entities in the text are classified into predefined categories. The ‘*en\_core\_web\_sm’* function was used to implement this step.

**DISCUSSION**

In this project, due to the minimal number of unstructured information gotten from these webpages, the procedures implemented above can limit getting a useful result. Therefore, to properly utilise the information provided on these webpages, other procedures/techniques such as; stemming, chunking, relation extraction, collocation extraction and conference resolution can be utilised.

Reference:

[1] HTML Tag, retrieved from webpage; ‘https://www.pcmag.com/encyclopedia/term/html-tag’ ; Tuesday 11, February, 2020.

[2] Sentence Detection, retrieved from webpage; ‘https://realpython.com/natural-language-processing-spacy-python/#sentence-detection’; Tuesday 11, February, 2020

[3] Part of Speech Tagging, retrieved from webpage, ‘https://realpython.com/natural-language-processing-spacy-python/#part-of-speech-tagging’; Tuesday 11, February 2020

[4] Semmar, Nasredine & Laïb, Mariama & Fluhr, Christian. (2006). Using Stemming in Morphological Analysis to Improve Arabic Information Retrieval.