CE706 - Information Retrieval
Assignment 1: Indexing for Web Search

Ajith Vajrala(1802560), Cynthia Masetto(1802774)

22nd February 2019

1 Introduction

The following work consists on describing a system that turns a web site into structured knowledge. This system takes HTML pages as input, process them using the following techniques: sentence breaking, script segmentation, encoding and language detection, natural language processing, among others and output an index of terms identified in the documents. We use Python notebook environment to perform this system.

The work is organized as follows. First, we read the web pages provided and do text cleaning by getting rid of the HTML tags. Second, text is pre-processed and we part–of–speech tagging and utilized entity–recognizer are performed to identify entities. Third, we carry out sentence splitting, tokenization and normalization. Forth, we removed words that are not useful and do tf-idf and finally, we created .txt files for each of our text analysis and json file to store the top tf-idf scores in each document.

2 System Development

The following URLs were used to develop the system:

- URL1 = *"https://csee.essex.ac.uk/staff/udo/index.html"*
- URL2 = *"https://www.essex.ac.uk/departments/computer-science-and-electronic-engineering"*

2.1 URL reading and text cleansing

All websites raw information are contained in HTML format. For that reason, the text is needed to be pre-processed to remove all the HTML tags present in the raw data. We took urllib.request and beautiful soup libraries to perform text reading and cleansing. Beautiful soup is a python library that pulls data out of HTML and XML files.[1] This library helps transform a complex HTML document into a complex tree of Python objects that aids to get insights about the text we analysed. In this case we used it to remove the HTML and tabulations and after that we extract the links that are in the URLs.

2.2 Getting links

The links included in a website provide a lot of information and relevance about the site. In this section, we extracted the links included in the website and during analysis we find: for the first URL1, we obtained 254 links and for the second URL2 we obtained 129 links.

2.3 Meta Data Extraction

Metadata data is present for each website, which is generally the summary information or the abstract information about each website. Some examples are: author, date created, date modified, information about the files included, and for that reason when we filter trhough metadata the location of specific documents is much more easier. In this case, from each website we extract its own meta data information present. We have extracted the metadata information for the two links provided.

2.4 Extracting Nouns and Noun forms

On a website, most of the information is unique when we take a closer look into words rather than in phrases. In this section we want to perform tokenization, to do that we used the NLTK tool. With this tool we can split the text into sentences and then the sentences are split into individual words. Also with this tool, we are able to follow the meaning of a sentence based on the words that are used by categorizing and tagging the words. According to the NLTK tool documentation, over 25 categories can be added to the words, in this system, we selected adverbs and nouns. Adverbs modify verbs to specify the time, manner, place or direction of the event described by the verb, moreover they can also modify adjectives. On the other hand, nouns generally refer to people, places, things, or concepts and can appear after determiners and adjectives that can be subject or object of the verb.[2]

Therefore, we assigned the selected taggers for all the text in the websites, by using parts of speech tagging. In this process, the sequences were split to sentences, then, the sentence tokenizer was applied and after that, we split into words and then applied the word tokenizer. At this point the words were tagged and the nouns and adverbs were extracted from each website.

2.5 Entity and Relation Extraction

Another important step when extracting information identifying entities. Each website is different from each other because not only nouns and adverbs are different but also the topic and places, for that reason we seek to locate and classify named entities that are defined within this pages such as: names of persons, organizations, locations, expression of items, quantities, monetary values, percentages, etc. For that, we used "ne_chunk" tool in order to identify the them. This tool segments and labels multi–token sequences meaning that once the word–level tokenization has happened the chunking unifies them and also classify them just like tokenization. Therefore, for each URL we extracted and stored these entities and relations into text files.

2.6 Text pre-processing and TF-IDF

Term Frequency, Inverse Document Frequency it's a way to score the importance of words or terms in a document based on how frequently they appear across multiple documents. This means that if a word appears frequently it's because it is important and after that a high score is given. However, if a word appears in other many documents, the identifier is not unique and a low score is given. On the other hand, this method is commonly used when the goal is to model each document into a vector space, ignoring the exact ordering of the words in the document while retaining information about the occurrences of each word.[3] It is composed by two terms: the first one computes the normalized term frequency, which is the number of times a word appears in a document divided by the total number of words in that document, and the second term is the inverse document frequency, which is computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the term appears.

Also, the TF–IDF gives how important is a word to a document in a collection since it takes in consideration not only the isolated term but also the term within the document collection.[3] Because most of the text that is present in the website are words that don't give any new information, at this step, we store the relevant information about each website by performing TF–IDF. For each of the URLs we analysed the terms that occur frequently and scale them to know if they're important or not.

2.7 Web search and Cosine similarity

The last step for the system is to perform a web search that retrieves the most relevant link. For that we applied the cosine similarity, which is a measure of similarity between two vectors that measures the cosine of the angle between them. In this system, as we converted web pages text into TF-IDF numbers, if a new text comes, we perform the process of converting the text into TF–IDF numbers again, and after that we calculate the cosine distance between this new text and our previous stored TF–IDF of the links. Finally, the best link with less cosine distance is returned in the web search.

3 Discussion

This system process textual data and extract interesting and significant patterns to explore knowledge from the sources given. The system performs the following steps: collecting unstructured data from URLs, pre–processing and cleansing to detect and remove anomalies, and by doing this, we make sure we capture the real essence of the text included on the sites. We remove stop words, process and control tokenization to classify and extract relevant information. Then with the use of Natural Language Processing we performed entity recognition that allows to identify relationships and other information within the text and, we perform TF–IDF to scale the words frequency and perform web search trhough cosine similarity. Finally, 16 documents were generated that summarize all the system outputs.

However, as the size of data is increasing at exponential rates day by day this tool may one day be incapable to handle a lot of textual data, since it requires time to extract information. On the other hand, some issues arise when letting the tool interpret the sense of a document, although it is true that we can have an idea of what is the site or the text on the site about with all the filtered words, the problem is when some words have same spelling but give diverse meaning, these tools keep considering these words as similar that's why we have to be careful with grammatical rules according to the nature and context so the insights that we gained from the text are the closest as possible with the reality or the truly meaning of the text.

References

[1] Richardson, Leonard, "Beautiful Soup Documentation". Online resource: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[2] NLTK toolkit documentation, "Categorizing and tagging words". Online resource: https://www.nltk.org/book/ch05.html

[3] Stevenloria, "Tutorial: Finding important words in text using TF–IDF". Online resource: https://stevenloria.com/tf-idf/