Sentimental Analysis

By

[redacted]

A DISSERTATION

Submitted to

Manchester Metropolitan University

in partial fulfilment of the requirements

for the degree of

BSc (Hons) Computer Science

Table of Contents

1- ABSTRACT 3

2.1- LITERATURE REVIEW 4

2.2- INTRODUCTION 4

2.3- DEFINITION OF PROBLEM 4

2.4- PATTERN MATCHING 4

2.5- SENTIMENTAL ANALYSIS 5

2.6- TOOLKITS 6

2.6.1-Tokenizer: *6*

2.6.2- Sentence Split: *6*

2.6.3- Parts of Speech: *6*

2.6.4- Named Entity Recognition: *6*

3.6.5- Parsing: *6*

2.6.6- Coreference Resolution: *7*

2.6.7- Sentiment: *7*

2.6.7- Sequence Tagging: *7*

2.6.8- Topic Modelling: *7*

2.6.9- Document Classification: *7*

2.7-EXAMPLES OF TOOLKITS 8

*2.7.1- **Stanford's Core NLP Suite*** *8*

- PARTS OF SPEECH 8
- NAMED ENTITY RECOGNITION 8
- PARSING 8
- COREFERENCE RESOLUTION: 8
- SENTIMENT 8

*2.7.2- **Natural Language Toolkit** (NLTK)* *9*

- NAMED ENTITY RECOGNITION 9
- PARSING 9

*2.7.3- **Apache OpenNLP*** *9*

- PARTS OF SPEECH 9
- NAMED ENTITY RECOGNITION 9
- PARSING 9

3.7.4- Mallet *9*

2.8-SENTIMENT TOOLKITS AVAILABLE 10

2.8.1-AlchemyLanguage *10*

2.8.2- Lexalytics *11*

2.8.3- Stanford Core NLP *12*

2.9- REVIEW SENTIMENT ANALYSIS 12

3.1- PRODUCT DESIGN & IMPLEMENTATION 13

3.2- INTRODUCTION 13

3.3.1- DESIGN 13

3.4- IMPLEMENTATION 15

4.1- EVALUATION 19

4.2- Real world examples *21*

5- CONCLUSION 24

5.1- PROJECT REV EW 24

5.2- ISSUES 24

5.3- Future work 24
5.4- Summary 24
6- BIBLIOGRAPHY 25
7-APPENDIX 27
TERMS OF REFERENCE 27
Course Specific Outcomes: 27
Project Background: 27
Aims: 28
Objective: 28
Research: 28
Relevant work: 29
Development: 29
T me Table: 29
Del verables: 30
Requ red Resources: 30

Declaration

No part of this project has been submitted in support of an ap-plication for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project contain-ing citations to the work of others, this project is my own unaided work.

Signed _____[redacted]

1- Abstract

Natural Language Processing (NLP) is the method of giving meaning to a large amount of information, this is done by breaking down the text and making sense of each part of the sentence with annotators offered by the NLP toolkit. This project will focus on the sentiment annotator and its uses, furthermore, this paper will discuss the creation of a sentimental analysis system that outlines what the uses of sentimental analysis are and how beneficial it can be to companies it terms of saving time and removing human errors. The reader is encouraged to follow up the references provided for further detail. [1]

2.1- Literature Review

2.2- Introduction

Analysis of vast amounts of information on the internet has always been an interest to companies. This has into the research of the software Natural Language Processing (NLP). A natural language is a human spoken language, which is why the terminology NLP is the meaning of processing a human spoken language.

Machine learning algorithms have allowed for systems to understand more and more how humans speak and what the meanings are. [3][4]

2.3- Definition of problem

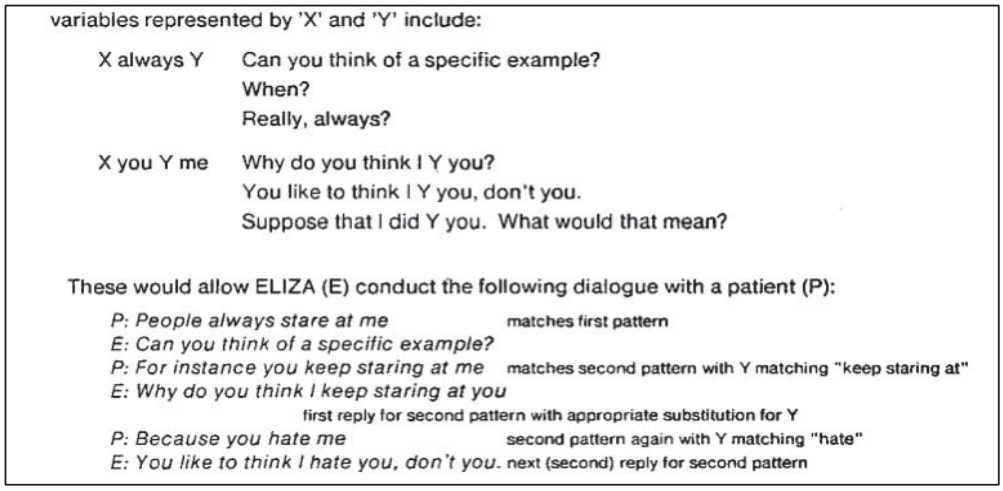
As powerful computers can be, they lack the intelligence to understand natural language. Computers can understand programming languages such as C and Java but due to the ambiguity of human generated languages, NLP designers have found it challenging to create systems that can completely understand natural languages [1]

Due to the fact that systems can not by themselves derive meaning from natural language, different NLP techniques can be used to achieve useful responses.

2.4- Pattern matching

Pattern matching takes a natural language utterance and assigns it with a pattern and depending on what pattern the system has given the given utterance the system will respond differently. A good example of this is a system called ‘Eliza’ [5]. Essentially Eliza is a system to simulate a physiologist having an interview with a patient. The patient would input text and Eliza would mimic the intelligent behaviour and respond relevantly. Eliza does not actually pattern match every word it was receiving, but rather would pick certain words and match a pattern to it and

give the patient a response. Even with little pattern matching Eliza would often fool its patients with the accuracy of the responses, Refer to (**Figure 1** for example)


variables represented by 'X' and 'Y' include:

X always Y	Can you think of a specific example? When? Really, always?
X you Y me	Why do you think I Y you? You like to think I Y you, don't you. Suppose that I did Y you. What would that mean?

These would allow ELIZA (E) conduct the following dialogue with a patient (P):

P: People always stare at me matches first pattern
E: Can you think of a specific example?
P: For instance you keep staring at me matches second pattern with Y matching "keep staring at"
E: Why do you think I keep staring at you
first reply for second pattern with appropriate substitution for Y
P: Because you hate me second pattern again with Y matching "hate"
E: You like to think I hate you, don't you. next (second) reply for second pattern

Figure 1[5]

2.5- Sentimental Analysis.

An attribute of NLP is sentimental analysis, where a machine learning algorithm can mine opinion and emotion from a text. For the English language, sentimental analysis has an 80% success rate and for this reason, companies are acknowledging its use and incorporating it into their process. [7] An example of how a company can incorporate sentimental analysis is with reviews management. Until now reviews have been posted where the customer can have given a rating from 1-10 or a star from 1-5, this information is useful but only to a certain extent. If your product has a high or low rating based on a number, the company can not see exactly where they have done well or poorly. Furthermore, there are text-based reviews where the customer has given their opinion of the product, this has given the opportunity for companies to see where they can improve their product correctly to meet the needs to the customers. Thousands of reviews can be processed for sentiment in seconds, compared to the hours it would take a team of people to manually complete. [1][2]

2.6- Toolkits

Different variations of natural language processing toolkits have been developed, each with their strengths and weaknesses in different areas. The list of the potential tools available are as follows;

2.6.1-Tokenizer:

A tokenizer divides the text into a sequence of tokens. Stanford core has a class that provides tokenization for English called PTBTokenizer. PTBTokenizer has Unicode compatibility.

2.6.2- Sentence Split:

Sentence Split (SSplit), splits a sequence of tokens into a sentence.

2.6.3- Parts of Speech:

Parts of speech is a category which is assigned to the words in a particular sentence depending on its syntactic functions. Following are the categories of POS: noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection.

2.6.4- Named Entity Recognition:

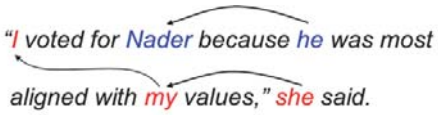
This is a process of searching and categorising named entities within some piece of text. Some examples of NER are persons, organisations, locations etc.

3.6.5- Parsing:

Provides full syntactic analysis, a parser works out the structure of the sentences. It figures out phrases in sentences and the object or subject of a verb.

2.6.6- Coreference Resolution:

Coreference resolution finds all the words that refer to the same entity in a text.



[9]

2.6.7- Sentiment:

This is simply a way to recognise the attitude of the person writing the text. This can be based on either just a small chunk of text or on the context of the whole document.

2.6.7- Sequence Tagging:

Sequence tagging is an algorithmic assignment of a label to the values of each sequence.

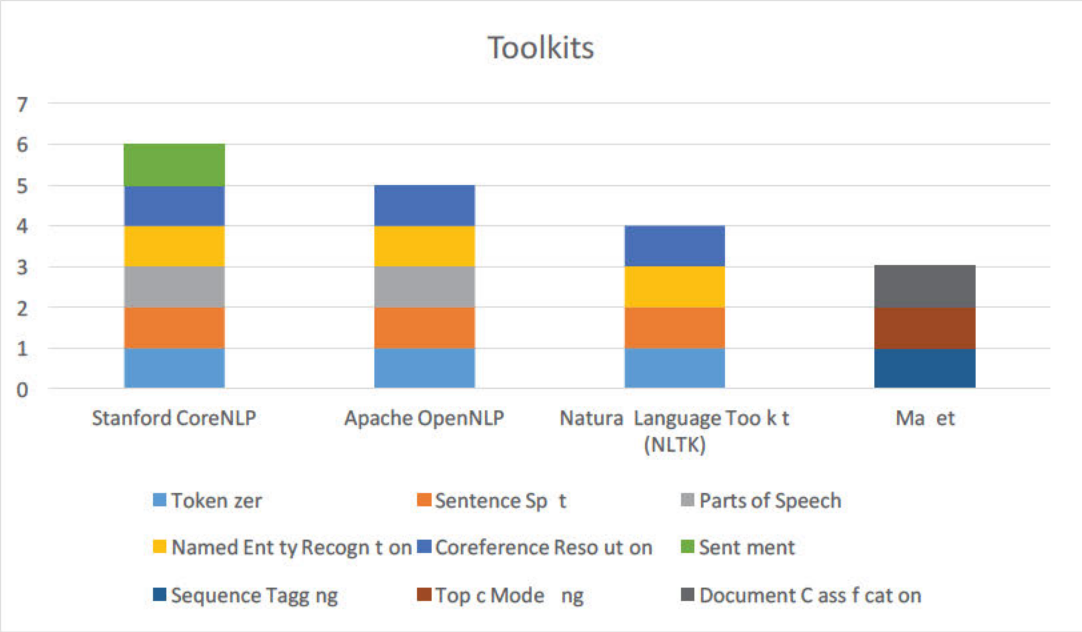
2.6.8- Topic Modelling:

Topic modelling allows you to analyse large pieces of text. Sometimes it can be hard to know and split each topic separately so this procedure lets you arrange sentences into their own individual topics.

2.6.9- Document Classification:

This is where the system places each document into its own categories and classes depending on the nature of text that document holds. Images, text files and music are the types of files that can be classified.

2.7-Examples of Toolkits



2.7.1- Stanford's Core NLP Suite

Stanford Core NLP is one of the most popular open source toolkits out there, this is due to its large amount of tools it offers.

- Tokenizer
- Sentence Split
- Parts of Speech
- Named Entity Recognition
- Parsing
- Coreference Resolution:
- Sentiment

With all these tools available, Stanford Core NLP is often the choice when choosing an NLP pipeline.

Stanford Core NLP also benefits from;

- Interfaces available for various major modern programming languages including Java

- Support for a number of major (human) languages

2.7.2- Natural Language Toolkit (NLTK)

NLTK is a Python-driven NLP and is available on Windows, Mac and Linux. NLTK makes the understanding of theory and the practical side simple which is why it's a favoured toolkit within teaching [10]. The tools available with NLTK are;

- Tokenizer
- Sentence Split
- Named Entity Recognition
- Parsing

Unlike Stanford Core NLP it lacks the sentiment tool which is the reason it will not be compatible for this project.

2.7.3- Apache OpenNLP

Apache OpenNLP is a machine learning based toolkit. While not necessarily state of the art anymore in its approach, it remains a solid choice that is easy to get up and running. The tools available in Apache OpenNLP are;

- Tokenizer
- Sentence Split
- Parts of Speech
- Named Entity Recognition
- Parsing
- Coreference Resolution

3.7.4- Mallet

Mallet is a Java based NLP, it has sophisticated tools for documentation classification, top modelling and sequence tagging. Mallet Is an open source software

[15]

- Document classification
- Top modelling
- Sequence tagging

2.8-Sentiment toolkits available

The following toolkits have been researched and found they have got the sentiment library, each of them will be critically analysed and compared to see which one is chosen for the project.

2.8.1-AlchemyLanguage

AlchemyLanguage is a collection of API's developed by IBM. AlchemyLanguage presents its sentiment result with different percentage of how much a particular emotion was used. Refer to (**figure 2)** for example.

Entity	Relevance	Sentiment	Type
Elliot Turner	0.89066	positive	Person
Emotion	**Score**		
Anger	0.248592		
Disgust	0.195077		
Fear	0.221098		
Joy	0.209939		
Sadness	0.210848		

Entity	Relevance	Sentiment	Type
PR Newswire	0.252507	neutral	PrintMedia
Emotion	**Score**		
Anger	0.107107		
Disgust	0.186492		
Fear	0.195534		

Figure 2

in this example, a paragraph was analysed and from the analysis, it is made clear what each emotion has scored. This is a useful tool when the user is trying to understand the core emotions of a particular text. The downfall of AlchemyLanguage and the reason it is not being used in

this project is due to the fact it is only free for a 30-day period and after that, it goes into a payment plan. [11]

2.8.2- Lexalytics

Lexalytics was the first to market with a commercial sentiment analysis engine way back in 2004, since then Lexalytics has made a big portfolio within the business. Here is an example of how Lexalytics analyses a text document. See (**Figure 3**) for example;

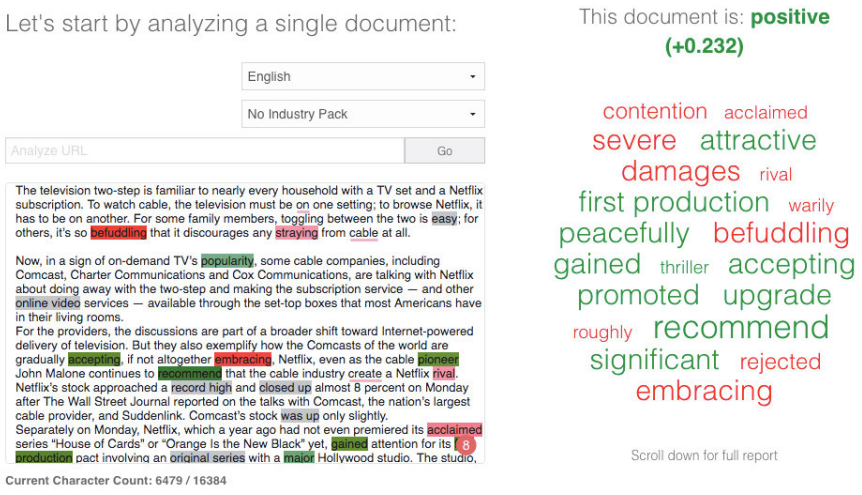


Figure 3 [12]

Lexalytics will give a Boolean value of the document whether it is a positive or negative, how Lexalytics differentiates itself from other sentimental tools is it will produce a graphically pleasing collage with different words that stand out with emotion and based on the impact that word has in the document the word will be displayed bigger or smaller in the collage, also if the word is negative it will be coloured red and green for positive. These features make the user's life easier and in a way more fun to do because instead of going through a 20-page document trying to decipher what the emotion of the document is the user can simply run the Lexalytics API on the document and it will do all the hard work for them. The reason it will not be used in this particular project is due to the similar reason AlchemyLanguage had, the

intro package to use the API starts at $999 a month which surpasses the resources available to accomplish the set goal.

2.8.3- Stanford Core NLP

As discussed, Stanford Core NLP comes with the sentiment tool and it is open source which allows it to be implemented to the project without any costs. Stanford CoreNLP uses a deep learning model which actually builds up a representation of whole sentences based on the sentence structure. It computes the sentiment based on how words compose the meaning of longer phrases. This way, the model is not as easily fooled as previous models. For example, the model learned that funny and witty are positive but the following sentence is still negative overall:

“This movie was actually neither that funny, nor super witty.” [13]

With all the advantages this toolkit offers such as Java language compatibility, strong sentiment structure to determine text with accuracy and finally its open source availability, it allows it to stand out from the crowd for this project. Due to these reasons, it will be the chosen toolkit.

2.9- Review Sentiment Analysis

This project is hoping to achieve a system that will analyse text-based reviews to simplify the process of review management. The goal will be to allow a company such as a café to implement this system to their app to take the reviews of their customer and produce and clear trend with the reviews so that the owner of the café can see what the customers are feeling about the service that is being provided. With this information, the manager of the café can always say on top of on problems the customers are reporting and this, in turn, will reflect the

customer satisfaction of the café. This way it is clear to see how big of an impact this system can have on a business. Many business owners can find themselves not being able to reach out to customer's demands because it is not clear what's going wrong to improve it. In the business of sentimental review systems, there is a large collection of products available but the majority of what's available is API's which is intended to be adopted by a company into their app/website. The problem with this is small businesses usually do not have the best websites or applications for customers to use makes it redundant for them to even implement and sentiment API to their review's page. This project will provide a business will a full set up of either a website or application which is easy to use for their customers and will have the reviews page with the sentiment system set up for the business owner. [2]

3.1- Product Design & Implementation

3.2- Introduction

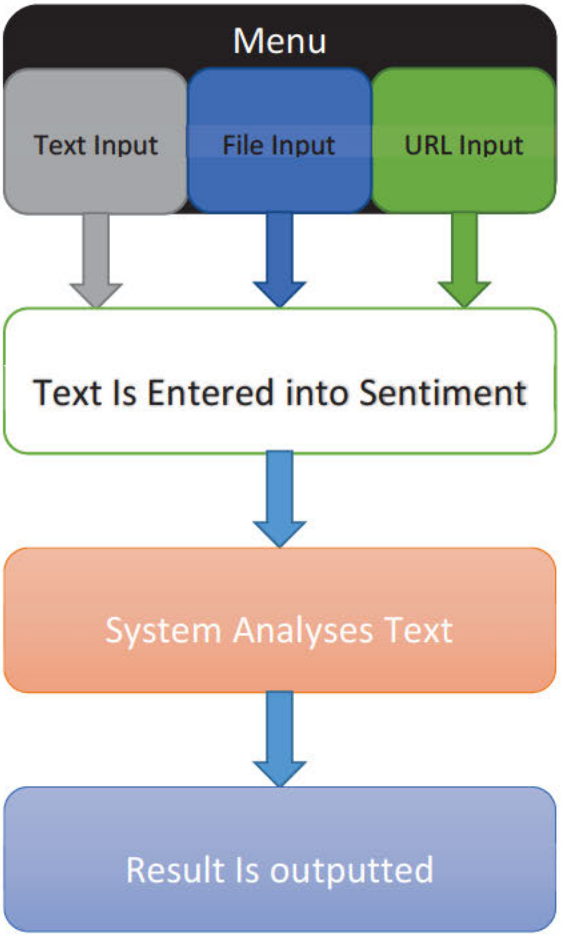
To introduce the design statement, the design aspect of the sentiment system will be discussed and what tools have been used help design it. The design if the system needs to be a simple yet practical at the same time. The back end of the system has a Stanford core NLP sentiment API built into it that does the analysis of the words and outputs the sentiment of those words.

3.3.1- Design

The back-end of the sentiment system has a built in API, the reason for using the API is because it is one of the most popular sentiment API on the market which ensures the results the system outputs is at a high level, the second reason for using the Stanford Core NLP sentiment API is due to the fact that it saves a lot of time on production of the system which helps meet the deadline. The way the API works is, it needs a source of input, this input needs to contain a

string of words which then the API assigns a sentiment value to the word and the sentence as a whole.

From this diagram its possible to get a visualisation as to how the system will work. Firstly, the user will get presented with a menu, from there the user has three input options to select from, the first method of input is text input. This method allows the user to manually enter text into the system and get a result from the text entered. The second method of input is file input, with this method the user gets a GUI menu where they can select a text file from their directory and then import that file to the sentiment system. Finally, the last method of selection is URL input, with this method of input the system asks for the user to enter an URL, making sure the page the user inputs is not too filled with text as it can take a long time for the system to analyse it. After the method of input has been selected and the text has been entered into the system, the next step the system will take is to process the text, the time taking to process the text varies on the size of the text, a normal sized paragraph can take up to 3-5 seconds. Finally, the user is presented with a small analysis report containing information about the sentiment of the text.



3.4- Implementation

This section will clarify as to how every part of the system works behind the scenes. the menu uses a simple while loop that takes a 'String' variable 'choice' which depending on what the user has inputted will select the right method to be used. A scanner listens for the user to enter a number between one and three. Once the user has selected their desired number the method accordingly will be run.

```
Scanner choose = new Scanner(System.in);
String choice= "";
while (!"end".equals(choice)) {

    System.out.println("\nHi and welcome to the sentimental analysis tool. \n\n"
            + "Select which method of analysis you want to run.\n"
            + "\nText Input=1 \n"
            + "Url input=2 \n"
            + "File input=3 \n ");
    choice = choose.nextLine();


    if ("1".equals(choice)) {
          choice = textInput();
         break;

        }

    else  if ("2".equals(choice)) {
        choice = urlInput();
        break;
    }
    else if ("3".equals(choice)) {
        choice = fileInput();

    break;
}
continue;


    }

choose.close();
```

This is the front end screen the user will be presented with; from this stage, the user has the ability to select which type of method they want to run.

```
Hi and welcome to the sentimental analysis tool.

Select which method of analysis you want to run.

Text Input=1
Url input=2
File input=3
```

The textInput method simply asks the user to enter text which they would like to have analysed and a scanner takes the text and imports it into the sentiment analyser.

```
  public static String textInput(){

      Scanner scanner = new Scanner(System.in);
        System.out.println("Please enter text");

        sentence1 = scanner.nextLine();

      return sentence1;

      //With this method the user is able to upload a t
  }
```

The fileInput method opens a dialoug where the user is able to select a certain text file. Once the user has selected their text file the method reads the text and stores the text in the 'line' variable.

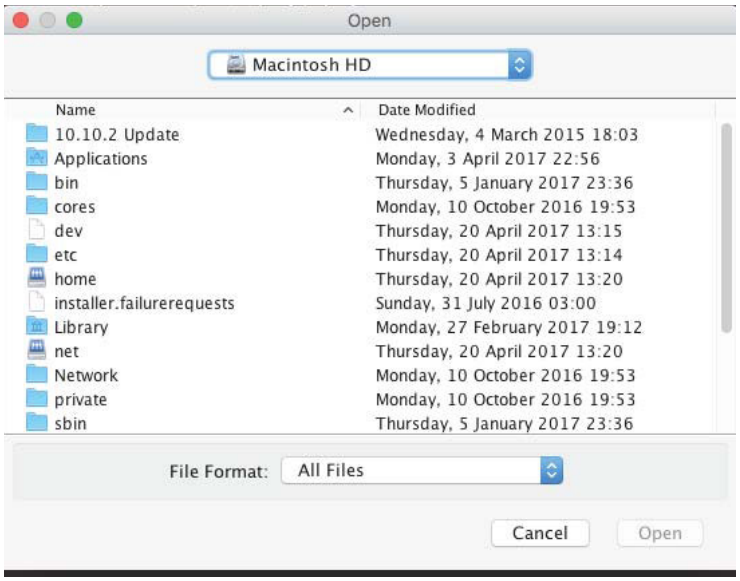
```
public static String fileInput() throws IOException{

      JFileChooser chooser = new JFileChooser();
        int result = chooser.showOpenDialog(null);
        // This assumes user pressed Open
        // Get the file from the file
        File file = chooser.getSelectedFile();
        // Open the file
        java.io.BufferedReader reader = new java.io.BufferedReader(new java.io.FileReader(file));
        String line = reader.readLine();

        reader.close();


    return line;
```

the figure bellow shows what the user is presented with when the fileInput method is run, using this GUI, the user can easily locate the file that needs to be analysed.



The final method the user is able to use is the URL method, the script establishes an open connection which allows it to connect to the page, furthermore a Buffer reader scans the page and brings back the text, the content is then converted into a String ready to be analysed.

```
public static String urlInput(){
    StringBuilder content = new StringBuilder();

    Scanner scanner = new Scanner(System.in);
      System.out.println("Please enter URL");

      urltext = scanner.nextLine();
    // many of these calls can throw exceptions, so i've just
    // wrapped them all in one try/catch statement.
    try
    {
      // create a url object
      URL url = new URL(urltext);

      // create a urlconnection object
      java.net.URLConnection urlConnection =  url.openConnection();

      // wrap the urlconnection in a bufferedreader
      BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));

      String line;

      // read from the urlconnection via the bufferedreader
      while ((line = bufferedReader.readLine()) != null)
      {
        content.append(line + "\n");
      }
      bufferedReader.close();
    }
    catch(Exception e)
    {
      e.printStackTrace();
    }
    return content.toString();
}
```

Whichever method the user has chosen to use, the text is passed into the annotation variable and from there the sentiment annotator analyses the text and outputs the sentiment result and finally at the end it prints out the overall sentiment of the text entered.

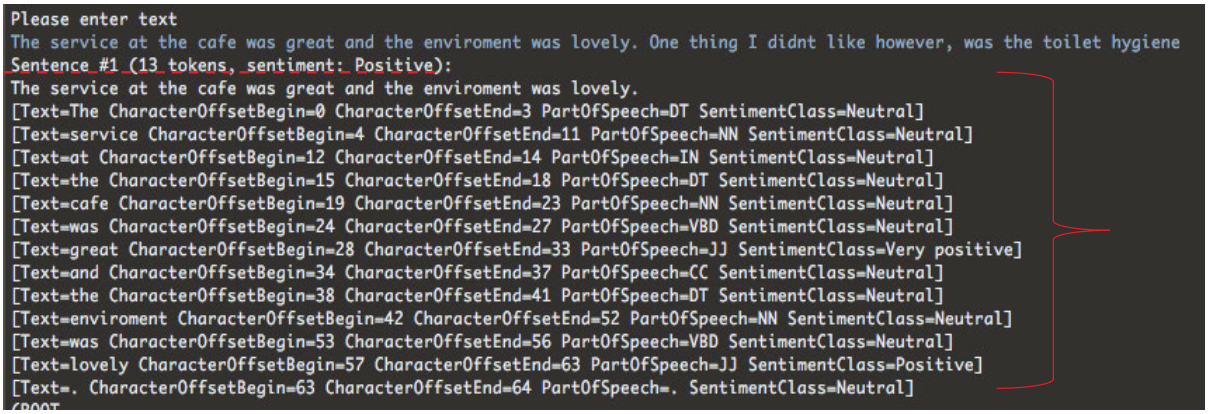
```
List<CoreMap> sentences = annotation.get(CoreAnnotations.SentencesAnnotation.class);
   if (sentences != null && ! sentences.isEmpty()) {
       CoreMap sentence = sentences.get(0);
      System.out.println("\n \n The overall sentiment of the text is " + sentence.get(SentimentCoreAnnotations.SentimentClass.class));
```

4.1- Evaluation

To introduce this section, it will cover how the system will work in a real life example and test the systems reliability to judge whether the does indeed perform well.

To show how the API works in a real life simulation, we can see the figures bellow demonstrating how the API has processed a customer's review, the string inputted into the system is highlighted in blue font.

"The service at the café was great and the environment was lovely. One thing I didn't like however, was the toilet hygiene."



As highlighted above, the red outlined text is where the sentiment of the first text has been displayed, "The service at the café was great and the environment was lovely". From that line it can be seen how many words are in the sentence, 13 tokens. Unfortunately, it does count the punctuation as a token. As shown this sentence has an overall positive sentiment, by looking at the sentiment of every word we can see how the API has processed this. The curly brace on the right side of the figure highlights the sentiment word analysis of each word, each line of the analysis has got a 'Text' word, which is a word from the sentence, also every word has a 'SentimentClass' assigned to it. As it can be seen every word has a 'Neutral' sentiment except for the word "great" which has a 'SentimentClass' of "Very positive" assigned to it and the word "lovely" which has "positive" assigned to it. Even though the 'SentimentClass' "Neutral" is the dominant sentiment, the API likes to chose a more prominent sentiment than Neutral.

```
Sentence #2 (11 tokens, sentiment: Negative):
One thing I didnt like however, was the toilet hygiene
[Text=One CharacterOffsetBegin=65 CharacterOffsetEnd=68 PartOfSpeech=CD SentimentClass=Neutral]
[Text=thing CharacterOffsetBegin=69 CharacterOffsetEnd=74 PartOfSpeech=NN SentimentClass=Neutral]
[Text=I CharacterOffsetBegin=75 CharacterOffsetEnd=76 PartOfSpeech=PRP SentimentClass=Neutral]
[Text=didnt CharacterOffsetBegin=77 CharacterOffsetEnd=82 PartOfSpeech=VBP SentimentClass=Neutral]
[Text=like CharacterOffsetBegin=83 CharacterOffsetEnd=87 PartOfSpeech=IN SentimentClass=Neutral]
[Text=however CharacterOffsetBegin=88 CharacterOffsetEnd=95 PartOfSpeech=RB SentimentClass=Neutral]
[Text=, CharacterOffsetBegin=95 CharacterOffsetEnd=96 PartOfSpeech=, SentimentClass=Neutral]
[Text=was CharacterOffsetBegin=97 CharacterOffsetEnd=100 PartOfSpeech=VBD SentimentClass=Neutral]
[Text=the CharacterOffsetBegin=101 CharacterOffsetEnd=104 PartOfSpeech=DT SentimentClass=Neutral]
[Text=toilet CharacterOffsetBegin=105 CharacterOffsetEnd=111 PartOfSpeech=NN SentimentClass=Negative]
[Text=hygiene CharacterOffsetBegin=112 CharacterOffsetEnd=119 PartOfSpeech=NN SentimentClass=Neutral]
```

Now for the second sentence, “One thing I dint like however, was the toilet hygiene”. We can see that this sentence is a clear negative comment and as such the system has given the sentence a negative overall sentiment. However, the word that was picked out to be ‘Negative’ was the word “toilet”. This result was a little confusing as the word “toilet” doesn’t really have a sentiment, the conclusion was that the collective words had an impact on the word’s sentiment and therefore giving the sentence an ‘Negative’ sentiment. To explore this conclusion, the sentence “The toilet hygiene was good” was tested and this was the result.

```
the toilet hygiene was good
Sentence #1 (5 tokens, sentiment: Positive):
the toilet hygiene was good
[Text=the CharacterOffsetBegin=0 CharacterOffsetEnd=3 PartOfSpeech=DT SentimentClass=Neutral]
[Text=toilet CharacterOffsetBegin=4 CharacterOffsetEnd=10 PartOfSpeech=NN SentimentClass=Negative]
[Text=hygiene CharacterOffsetBegin=11 CharacterOffsetEnd=18 PartOfSpeech=NN SentimentClass=Neutral]
[Text=was CharacterOffsetBegin=19 CharacterOffsetEnd=22 PartOfSpeech=VBD SentimentClass=Neutral]
[Text=good CharacterOffsetBegin=23 CharacterOffsetEnd=27 PartOfSpeech=JJ SentimentClass=Positive]
```

As it can bee seen what was earlier predicted was true, the sentence has now got a positive sentiment and the word “toilet’ has no longer got a ‘Negative’ sentiment but a ‘Neutral’ one and the word “good” is the ‘Positive’ word.

Going now back to the original testing, the system at the bottom of the results outputs a over all sentiment for both sentences, this way it is a simple way to generally know what sentiment the text has, and if the user wanted to they could go more in-depth as to what specific sentiment each word has. In this example the review text “The service at the café was great and the environment was lovely. One thing I didn’t like however, was the toilet hygiene.”, got a

positive sentiment, this is due to the fact that the sentence has got two positive words and only one negative word, which gives the text an over all a more positive outcome.

```
The overall sentiment of the text is Positive
```

4.2- Real world examples

To test the system, a series of reviews will be tested from Cat Café situated in Manchester. First of a positive review will be tested, the text of the reviews can be viewed in the figures bellow.

As it can be seen this person has given four stars for the café, but the owner of the café has to read through the whole review to get an idea as to what reasons this person has given such a high score.

[14]

From the first sentence we can see it has been given a positive sentiment, and the words that are positive in the sentence are "lovely", which the user can note as the environment they have set up is good and the second word that positive is "clean", this is also something for the owner can note, that the café is kept clean.

```
Sentence #1 (15 tokens, sentiment: Positive):
The cafe is lovely and very clean, and there are plenty of cats.
[Text=The CharacterOffsetBegin=0 CharacterOffsetEnd=3 PartOfSpeech=DT SentimentClass=Neutral]
[Text=cafe CharacterOffsetBegin=4 CharacterOffsetEnd=8 PartOfSpeech=NN SentimentClass=Neutral]
[Text=is CharacterOffsetBegin=9 CharacterOffsetEnd=11 PartOfSpeech=VBZ SentimentClass=Neutral]
[Text=lovely CharacterOffsetBegin=12 CharacterOffsetEnd=18 PartOfSpeech=JJ SentimentClass=Positive]
[Text=and CharacterOffsetBegin=19 CharacterOffsetEnd=22 PartOfSpeech=CC SentimentClass=Neutral]
[Text=very CharacterOffsetBegin=23 CharacterOffsetEnd=27 PartOfSpeech=RB SentimentClass=Neutral]
[Text=clean CharacterOffsetBegin=28 CharacterOffsetEnd=33 PartOfSpeech=JJ SentimentClass=Positive]
[Text=, CharacterOffsetBegin=33 CharacterOffsetEnd=34 PartOfSpeech=, SentimentClass=Neutral]
[Text=and CharacterOffsetBegin=35 CharacterOffsetEnd=38 PartOfSpeech=CC SentimentClass=Neutral]
[Text=there CharacterOffsetBegin=39 CharacterOffsetEnd=44 PartOfSpeech=EX SentimentClass=Neutral]
[Text=are CharacterOffsetBegin=45 CharacterOffsetEnd=48 PartOfSpeech=VBP SentimentClass=Neutral]
[Text=plenty CharacterOffsetBegin=49 CharacterOffsetEnd=55 PartOfSpeech=RB SentimentClass=Neutral]
[Text=of CharacterOffsetBegin=56 CharacterOffsetEnd=58 PartOfSpeech=IN SentimentClass=Neutral]
[Text=cats CharacterOffsetBegin=59 CharacterOffsetEnd=63 PartOfSpeech=NNS SentimentClass=Neutral]
[Text=. CharacterOffsetBegin=63 CharacterOffsetEnd=64 PartOfSpeech=. SentimentClass=Neutral]
(ROOT
```

As shown on in the figure bellow the system has predicted correctly that the review is positive.

```
The overall sentiment of the text is Positive
```

This is an example of a bad review, without reading the review contents we can tell from the title and the two out of 5 stars that this person has given the Cat café a bad review.

"Not great"

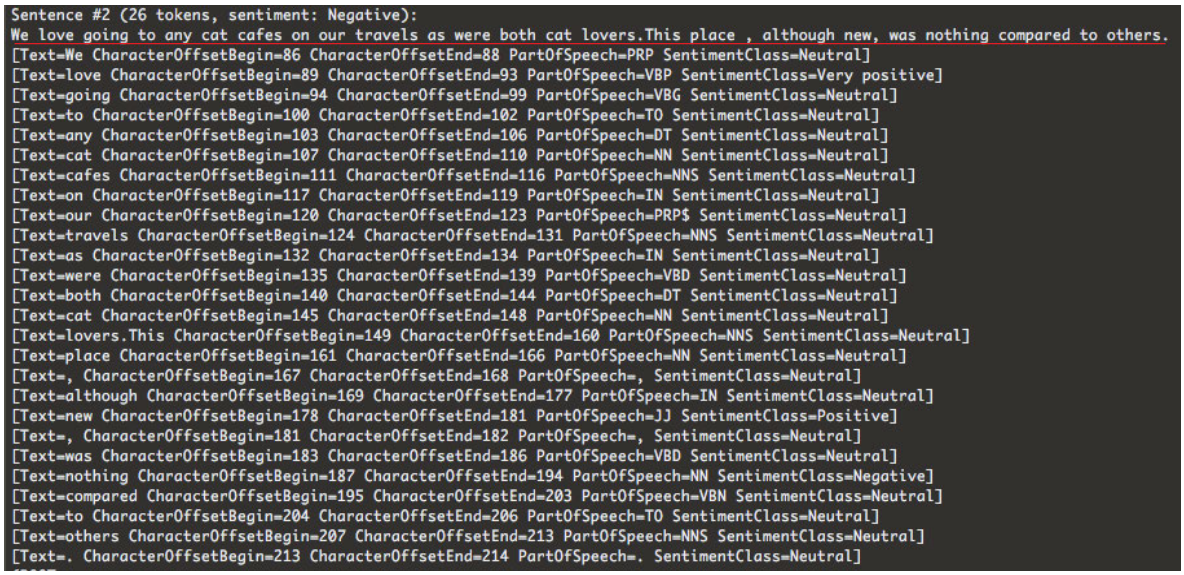
Reviewed 4 weeks ago ▫ via mobile

Me and my partner have been to a few cat cafes such as London , Edinburgh, Paris etc. We love going to any cat cafes on our travels as were both cat lovers.This place , although new, was nothing compared to others. The staff at the till were not very enthusiastic and read off a script to us for rules etc. We then entered the cafe,where there were no cakes just tasteless, shop bought cookies. The cafe staff were friendly however another member of staff was over bearing and very strict. This made our group feel uncomfortable to walk around the cafe or interact with cats. We also got the feeling that non of the cats really liked the company of the public. Controversial as it may be, but to open an establishment like this the cats need to have a certain nature that allows the public to enjoy them as well as the cats Feeling comfortable there. This place needs to review it's rules about stroking cats when they are lying down or looking like there going to sleep as this is not the case in other cafes. It is also ridiculous as cats sleep on average 20 hours of the day. As long as the cats are enjoying strokes this should not be an issue. We also heard that this place wants to open other stores which i believe is wrong and should be kept to independent, passionately run establishments to keep from the above problems we experienced.

[14]

Running the review through the sentiment system we get this result; This displayed result is of the second sentence, it has a negative sentiment, this might seem odd as from the list of words it shows that all there is a very positive word, a positive word and one negative word, the rest

are neutral. This means that there are more positive words than negative, however, the last part of the sentence which contains the negative word, it can be seen on the top part of the figure bellow. “although new, was nothing compared to others”, this part of the sentence totally discredits the first part of the sentence. This really does show the power of the API and how intelligent it is.



At the start of the paper these objectives were outlined;

- To create a usable system that detects sentiment in text
- Give user different types of inputs
- Gives user different levels of sentiment output

From the figures its clear that the system is usable and gives clear instructions as to how it is needed to be used, secondly, from the menu it can be seen that the user has the ability to chose from 3 different types of inputs for the user to input text into the sentiment system and finally the user does get different levels of sentiment output, the user gets a sentiment analysis of the whole text, each sentence and each word. This was the user is able to chose as to how specific they want the analysed text.

5- Conclusion

5.1 Project review

When creating this project, the objectives were to create a sentimental analysis system that gives the user different types of text imports. These objectives have been met and also outputs several different levels of sentiment output.

5.2 Issues

Although the objectives have been met, there were some complications on the way. First issue was there was no way to access the API file to modify it and give the system a more unique quality. This issue created a problem when trying to create a usable GUI for the system due to the fact that the API output a stream of sentiment data and there was no way of managing the stream of data to direct it to a GUI. Stanford was contacted asking permission to access the API in hopes that eventually the output data produced by the API could be managed but Stanford never got back.

5.3 Future work

In Future work, there would be several areas that would could be in need of improvement, first of all would be to be able to create a GUI for a user be presented the data in a more clearer way with less clutter, secondly it would be an achievement when this system has been integrated with an already established review system, this could perhaps be a web or mobile based review system for a company.

5.4 Summary

In summary, the aim of the project was to create a system that allows for a user to use different types of text imports to be analysed and give the user clear sentiment feedback, these aims have been achieved. Although the system is basic, the concept of the project is clear and does meet the objectives set.

6- Bibliography

Christopher, M., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J. and McClosky, B. (2014) *The Stanford CoreNLP Natural Language Processing Toolkit*. Available at: http://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf (Accessed: 7 October 2016).[1]

Kiser, M. (2016, August 11). Introduction to natural language processing (NLP) 2016. Retrieved October 20, 2016, from Blog Posts, http://blog.algorithmia.com/introduction-natural-language-processing-nlp/ [2]

Copestake, A. (2004) *Natural Language processing*. Available at: https://www.cl.cam.ac.uk/teaching/2002/NatLangProc/revised.pdf (Accessed: 10 October 2016). [3]

Introduction to Sentiment Analysis. Retrieved November 17, 2016, from lct-master, https://lct-master.org/files/MullenSentimentCourseSlides.pdf [4]

Weizenbaum, J., 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), pp.36-45.[5]

Hayes, P.J. and Carbonell, J.G., 1983. A tutorial on techniques and applications for natural language processing.PP 8. [6]

Rogynskyy, O. (2013, July 16). What are the applications of sentiment analysis? Why is it in so much discussion and demand? Retrieved November 23, 2016, from Quora, https://www.quora.com/What-are-the-applications-of-sentiment-analysis-Why-is-it-in-so-much-discussion-and-demand [7]

Ingersoll, G. (2015, July 8). 5 open source tools for taming text. Retrieved November 24, 2016, from Opensource, https://opensource.com/business/15/7/five-open-source-nlp-tools [8]

http://nlp.stanford.edu/projects/coref.shtml [9]

Bird, S., 2006, July. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions* (pp. 69-72). Association for Computational Linguistics.[10]

document. IBM Watson developer cloud. Retrieved November 25, 2016, from http://www.ibm.com/watson/developercloud/alchemy-language.html [11]

Lexalytics. (2016). Sentiment analysis and text Analytics demo. Retrieved November 25, 2016, from https://www.lexalytics.com/demo [12]

Deeply moving: Deep learning for sentiment analysis. (2013, August). Retrieved November 25, 2016, from http://nlp.stanford.edu/sentiment/ [13]

https://www.tripadvisor.co.uk/Attraction_Review-g187069-d10842872-Reviews-or20-Cat_Cafe-Manchester_Greater_Manchester_England.html#REVIEWS [14]

McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.[15]

7-Appendix

Terms of Reference

Course Specific Outcomes:

BSc (Hons) Computer Science (CS)

• To develop an understanding of the nature of databases and be able to develop, maintain and interrogate databases;

• To develop knowledge of computer hardware and an understanding of how the selection of hardware will affect the performance of an application;

• To investigate the interaction between hardware and software and the influence of this interaction on the design of computer systems;

• To study the fundamentals of computer network communications and communication protocols; and

• To study the management and security of networked systems

Project Background:

Natural language processing (NLP) is a process where a machine breaks down natural language into useful information, natural language is any human-generated language and not a programming language. This process of breaking down the natural language is a big challenge for programmers because human language is very ambiguous and leads to inaccuracy in the process. This process is usually done by machine learning, machine learning has complex algorithms which are used to recognise patterns and use that to grow its intelligence.

Having the ability to disassemble text to its core elements using natural language processing allows for data to be processed and manipulated to create meaningful data. This data has many real world uses for companies and individuals that may want to process large amounts of text

or speech. Uses may include processing reviews of a product using sentiment analysis to determine user attitude towards the product, saving countless hours required for a human to the same task.

The picture below is an example of how text is processed using the Stanford CoreNLP basic dependencies annotator. [1]

Aims:

The aim of the project is to use the sentiment attribute of Stanford core NLP and creating a software template for a business to apply it to their review system, this way the software can go through the reviews for the management and this way hours used to go through the reviews can be saved, the software will detect if the review is negative or positive and it will pick out the words that stand out such as hygiene, quality and service. Using this feedback, the manager can take the positive feedbacks and maintain the same standard to the applied areas and use the negative feedback to improve where the company is lacking the certain standard.

Objective:

For the outlined aims to be achieved, certain objectives have to be met. To meet the objectives thorough research has to be made on the subject of NLP to make sure the knowledge is sufficient when approaching the aims.

- To create a usable system that detects sentiment in text
- Give user different types of inputs
- Gives user different levels of sentiment output

Research:

Throughout the project, much research is required to achieve an easy to use and effective system. For this to be achieved other similar systems have to be looked into to see what their approach is to the scenario, from there, ideas and suggestions will be taken to improve our system. Research into sentimental analysis will be required to see what Stanford CoreNLP has

to offer to see whether any additions is necessary to make to the already available sentimental attribute.

Relevant work:

With this development in NLP we can see how we are ever so close to achieve emotions with robots. With the right tools such as facial recognition for the robot to recognize the person, the algorithm in the robot will adjust itself to fit the profile of the particular person and that way the robot can judge the context better. The robot will have voice recognition to translate the words heard into words and finally to take the words and run it through the sentiment analysis system that this project has created. The system would output a result as to what emotion the person is speaking in and the robot will know how to respond accordingly, for example if the robot can detect the person is being positive the robot can use positive vocabulary to match the person's emotion.

Development:

The development of the NLP system will require careful planning and appropriate testing, the system will be developed using the JAVA language and it will act as a template that can be applied to apps and websites. The sentiment attribute needs deep testing to evaluate whether it is suitable to be applied on reviews and if not the then changes will be applied to it so it can take the correct analysis. A simple and user friend's GUI will be developed with sentimental analysis will be used on the back end of the system.

Once the backend has been married to the GUI a thorough testing with real-life examples will be used to see how accurate the results of the system and from there, changes will be made if need be.

Time Table:

The Timetable bellow outlines what the main tasks are and when each one of them needs to be completed. By sticking to these task deadlines the product and paper will be able to be submitted on time.

Project Time Plan

Tasks	Start	End	Duration (days)
Overview of project	19/09/2016	26/09/2016	7
Project Planning	26/09/2016	26/11/2016	61
Write TOR	30/09/2016	21/10/2016	21
Ethics Form	20/10/2016	30/10/2016	10
Research	19/09/2016	01/02/2017	135
Litrature Survery	31/10/2016	11/12/2016	41
Prototype	30/10/2016	31/12/2016	62
Design Evaluation	13/01/2017	31/01/2017	18
Report Outline	01/12/2016	20/02/2017	81
Draft Slides	20/02/2017	03/03/2017	11
Presentation practice	03/03/2017	13/03/2017	10
Finalised Product and Report	01/04/2017	24/04/2017	23
Final Submission	24/04/2017	24/04/2017	0

Deliverables:

The Deliverables for the project will contain each the paper which has the following sections;

- Terms of Reference (ToR):
- Literature Review
- Product Design & Implementation:
- Evaluation Design:

In addition to the paper, the product will be delivered in Java code form.

Required Resources:

The resources required to be able to complete the proposed system are listed bellow;

- Github
- Eclipse
- Browser
- Sanford Core NLP