Sentiment Analysis Engine Using Natural Language Processing

A PROJECT REPORT

submitted by

Rishit Bhatia (ROLL NO.:6119)

Dhiraj Gurkhe (ROLL NO.:6133)

Niraj Pal (ROLL NO.:6149)

in partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY**

**Department of Information Technology
Fr. Conceicao Rodrigues College Of Engg
Fr. Agnel Ashram, Bandstand, Bandra (W),
Mumbai - 400050
June - 2014**

CERTIFICATE

Certified that this project report **”Sentiment Analysis Engine Using Natural Language Processing”** is the bonafide work of **”Rishit Bhatia(6119),Dhiraj Gurkhe(6133), and Niraj Pal(6149) ”** who carried out the project work under my supervision.

Certified by

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
BANDRA - MUMBAI
MOULDING ENGINEERS WHO CAN BUILD THE NATION

Internal Guide	**Principal**	**HOD**
Mrs. Anusha Jayasimhan	Dr. Srija Unnikrishnan	Mr. Mahesh Sharma
Information Technology		Information Technology

Collage Seal

....	....
Internal Examiner	External Examiner

Abstract

Sentiment analysis using Natural Language Processing involves extraction of subjective information from documents like social media dataset to determine the polarity with respect to certain keyword. It is useful for identifying trends of public opinion in the social media, for the purpose of determining brand popularity. It aims to determine the attitude of a speaker or a writer with respect to some topic or simply the contextual polarity of a sentence. This project introduces a approach for automatically classifying the sentiment of social media data. The data is fed to the Sentiment Analysis Engine that analyses and interprets these messages which are then classified as either positive or negative with respect to a query term.

Contents

1 Introduction **2**

1.1 Motivation . . . 2

1.2 Objectives . . . 3

2 Literature Review **4**

2.1 Machine Learning . . . 4

2.1.1 Supervised Learning . . . 4

2.1.2 Unsupervised Learning Methods . . . 5

2.2 Feature Extractors . . . 5

2.3 Machine Learning Classifiers . . . 6

2.3.1 Decision Tree . . . 6

2.3.2 Naive Bayesian Text Classification . . . 6

2.3.3 Support Vector Machines (SVM) . . . 7

2.3.4 Maximum Entropy . . . 7

3 Problem Statement **8**

3.1 Drawbacks of Current System . . . 8

3.2 Possible Solution To Above Problem . . . 8

4 Project Description **9**

4.1 Overview of the project . . . 9

4.2 Diagrams . . . 10

4.2.1 Architecture Diagram . . . 10

4.2.2 Data Flow Diagram . . . 11

4.2.3 Activity Diagram . . . 13

4.2.4 Sequence Diagram . . . 18

5 Implementation Details **19**

5.1 Status . . . 19

6 Conclusion And Summary **20**

Chapter 1

Introduction

Large datasets are available on-line today, they can be numerical or text file and they can be structured, semi-structured or non-structured. Approaches and technique to apply and extract useful information from these data have been the major focuses of many researchers and practitioners lately. Many different information retrieval techniques and tools have been proposed according to different data types. In addition to data and text mining, there has seen a growing interest in non-topical text analysis in recent years. Sentiment analysis is one of them. Sentiment analysis, also known as opinion mining, is to identify and extract subjective information in source materials, which can be positive, neutral, or negative. Using appropriate mechanisms and techniques, this vast amount of data can be processed into information to support operational, managerial, and strategic decision making[1].

Sentiment analysis aims to identify and extract opinions and attitudes from a given piece of text towards a specific subject [2]. There has been much progress on sentiment analysis of conventional text, which is usually found in open forums, blogs and the typical review channels. However, sentiment analysis of microblogs like twitter is considered as a much harder problem due the unique characteristics possessed by microblogs (e.g. short length of status updates and language variations).

1.1 Motivation

The emergence of social media combined with microblogging services easy-to-use features have dramatically changed people's life with more and more people sharing their thoughts, expressing opinions, and seeking for support on such open social and highly connected environments. Monitoring and analysing opinions from social media provides enormous opportunities for both public and private sectors. For private sectors, it has been observed that the reputation of a certain product or company is highly affected by rumours and negative opinions published and shared among users on social networks. Understanding this observation, companies realize that monitoring and detecting public opinions from microblogs leading to building better relationships with their customers, better understanding of their customers needs and better response to changes in the market.

For public sectors, recent studies show that there is a strong correlation between activities on social networks and the outcomes of certain political issues. For example, Twitter and Facebook were used to organise demonstrations and build solidarity during Arab Spring of civil uprising in Egypt, Tunisia, and currently in Syria. One week before Egyptian presidents resignation the total rate of tweets about political change in Egypt increased ten-fold. In Syria, the amount of on-line content produced by opposition groups in Facebook increased dramatically. Another example is the UK General Election 2010. It has been shown that activities at Twitter are a good predicator of popularities of political parties . Thus tracking and analysing users activities on social media are they key to understanding and predicting public opinions towards certain political event or brand popularity etc.

1.2 Objectives

We can look at the project from a software engineering perspective, where the problem here can serve as the main functional requirement of the system. This actually helps us to frame our work and formulate our objectives as follows:

- **Objective 1:** Data scarcity should be alleviated; this implies that data should be preprocessed before it is getting fed into classier training.

- **Objective 2:** Sentiment classiers should be able to operate on the data streaming paradigm of microblogs. This means they should have the ability to work with limited resources of time and space.

- **Objective 3:** The problem of imbalanced sentiment distribution (sentiment drift) should be considered when building sentiment classiers. This means classiers are expected to work with imbalanced numbers of training instances in different classes.

- **Objective 4:** Classiers should be easily adapted to work with different microblogging services like Twitter and Facebook.

Chapter 2

Literature Review

In this section, some related work on Sentiment Analysis Engine based on natural language processing will be discussed

2.1 Machine Learning

Machine learning is used for automatic classification of the documents[3]. There exist two main types of machine learning: supervised and unsupervised learning. The difference between them is that in the former the class labels (e.g. positive, negative, or neutral) are present in the data set before learning while in the latter the class labels are not provided that is why it is the task of the learning algorithm to analyse the internal documents (sentences) structure and to assign class labels to them. The objective of the supervised learning is to create mapping (model) between the documents (sentences) and the class labels. Unsupervised learning is aimed at finding the intrinsic structure in the documents and to organize documents into the similarity groups (clusters) according to these common structures[4].

2.1.1 Supervised Learning

The supervised learning has as a goal in the end to map the unlabeled documents (sentences) to their real classes. It usually consists of the following four steps[5]:

- **Data collection and preprocessing**. At this step the tweets are collected, the tweets (sentence) classes are labelled, the features (e.g. n-grams, POS) are identified and the vector space representation of tweets (sentences) is created. The collected data can be divided into two main subsets: training set, which is used for creating the model and the test set which is used for testing the model. Sometimes the training set is split into two subsets: the actual model construction subset and a model validation subset, which can be employed for tuning the learner parameters.

  The following standard preprocessing techniques are used:

  - **Words splitting:** Words splitting or tokenizing is decomposing the sentences into words.
  - **POS tagging:** POS tagging assigns to every single word a label which correspond to its part of speech e.g. noun, adjective, verb, adverb etc.
  - **Stopwords removing:** Not all words in the sentences carry useful information for classification task and that is why it is beneficial to get rid of such useless words. Removing these stopwords make the classification routine easier.
  - **Stemming:** Stemming mainly deals with removing suffixes and prefixes from the words. The procedure of stemming can be explained by two sentences below. I like to watch this movie. And the second I liked her performance, it was marvellous. Here words like and liked would be treated by classification algorithm as two completely different words. That is why it is beneficial to stem these words to one common word.
  - **Feature selection:** Features are the indicator of sentiment in the sentences. Often considering every single word in a sentence cannot be as indicative of positive or negative sentiment as when considering higherlevel n grams. Bigrams to help with tweets that contain negated phrases like not good" or not bad."

- **Building the model:** At this step actual learning (training of the classifier) is done. It is usually the iterative and interactive process which is aimed at receiving the best model in the end. It includes the feature selection, learning algorithm application, and if needed tuning the learning algorithm.

- **Testing and evaluating the model:** At this step the model is applied to the tweets from the test set and their actual class labels are compared to the predicted ones.

- **Classification of the new documents:** Using the model to classify the unlabeled tweets [5]

2.1.2 Unsupervised Learning Methods

The unsupervised learning algorithms try to find some intrinsic structure in data and to organise the documents into clusters. Two types of clustering are used: partitional and hierarchical clustering. K-means clustering algorithm is used in partitional clustering. Clustering utilises similarity function or distance function in order to measure how similar two objects are or to measure a distance between two data points. A document is represented usually as a bag of words in document clustering. A document can be represented as a vector and usually the cosine similarity function is used in order to compute the similarity between two documents [4]

2.2 Feature Extractors

- **Unigram:** (Pang Lee)The unigram feature extractor is the simplest way to retrieve features from a tweet. The machine learning algorithms clearly perform better than our keyword baseline. They report 81.0%, 80.4%, and 82.9% accuracy for Naive Bayes,MaxEnt, and SVM, respectively.
- **Bigrams:** Bigrams are used to help with tweets that contain negated phrases like not good" or not bad". Negation as an explicit feature with unigrams does not improve accuracy as compared to using bigrams. However, bigrams tend to be very sparse and the overall accuracy drops in the case of both MaxEnt and SVM. Even collapsing the individual words to equivalence classes does not help. The problem of sparseness can be seen in the following tweet: @stellargirlIlooooooooovvvvvveee my Kindle2. MaxEnt gave equal probabilities to the positive and negative class for this case because there is not a bigram that tips the polarity in either direction as suggested by [6].

2.3 Machine Learning Classifiers

2.3.1 Decision Tree

It is a very efficient classification algorithm in which the learned classification model is represented as a decision tree. It is beneficial to have a small tree as it tends to be more accurate and is easier to understand by human users. The tree only covers a subset of rules that exist in data, which is sufficient for classification. A decision tree partitions the training data set into disjoint subsets so that each subset is as pure as possible (contains training examples of a single class). Divide-and-conquer strategy is used for learning of a tree, it recursively partitions the data to produce the tree. The best attribute to partition the data at the current node is chosen with the aim to maximise the purity [4].

2.3.2 Naive Bayesian Text Classification

It is the probabilistic approach to the text classification. Here the class labels are known and the goal is to create probabilistic models, which can be used to classify new texts. It is specifically formulated for text and makes use of text specific characteristics. The Naive Bayesian classifier treats each document as a bag of words and the generative model makes the following assumptions: firstly, words of a document are generated independently of context, and, secondly, the probability of the word is independent of its position. This is why the name naive was used for this algorithm. In real text documents the words often correlate with each other and the position of the word in text may play role.[4]

2.3.3 Support Vector Machines (SVM)

SVM is one of the most popular classification algorithms. It performs very accurate classification in many applications especially those involving high dimensional data. It is also one of the most accurate algorithms for text classification. In general SVM is a linear learning system that builds two class classifiers. This algorithm finds the maximal margin decision boundary to separate positive and negative examples. Learning is formulated here as quadratic optimisation problem. It has also a solution for finding the nonlinear decision boundaries; to do this, the original data is transformed to the much higher dimensional feature space. But it has the limitation as it allows only two classes, i.e. binary classification. For multiple classification additional strategies should be applied [4].

2.3.4 Maximum Entropy

The idea behind Maximum Entropy models is that one should prefer the most uniform models that satisfy a given constraint by (Alec Go, RichaBhayani, Lei Huang). MaxEnt models are feature-based models. In a twoclass scenario, it is the same as using logistic regression to and a distribution over the classes. MaxEnt makes no independence assumptions for its features, unlike Naive Bayes. This means we can add features like bigrams and phrases to MaxEnt without worrying about features overlapping.

Chapter 3

Problem Statement

3.1 Drawbacks of Current System

- The short length of status updates coupled with their noisy nature makes the data very sparse to analyse using standard machine learning classiers.
- The lack of labelled data needed for classiers training.
- Open nature of microblogs poses an open-domain problem where classiers should work in a multi-domain environment.
- The streaming fashion of microblogs where data arrives at a high speed. This means data should be processed in real time and classiers should adapt quickly with the newlyarrived data.

Thus, the **problem statement** can be formulated as :
"How to build a learning classifier that analyzes concise data from social media dynamically in a streaming manner so as to extract sentiments from it".

3.2 Possible Solution To Above Problem

The problem can be disentangled by developing an engine which amalgamates all the views, opinions or sentiments for the user on the basis of a keyword provided. This can be implemented by using concepts from natural language processing and machine learning. The data sets can be extracted from social media portals like twitter, facebook etc. These data sets can be analyzed using the proposed classification algorithms and lexicons which will thereby help in judging the polarity of sentences.

Chapter 4

Project Description

4.1 Overview of the project

The project will be a web application which will provide a search engine for culmination of sentiments of the keyword from the different social media datasets. The datasets data would be classified upon their polarity of negative, positive or neutral, this polarity classified data will be represented by different design in the User Interface. The overall sentiment will also be calculated and displayed separately.

4.2 Diagrams

4.2.1 Architecture Diagram

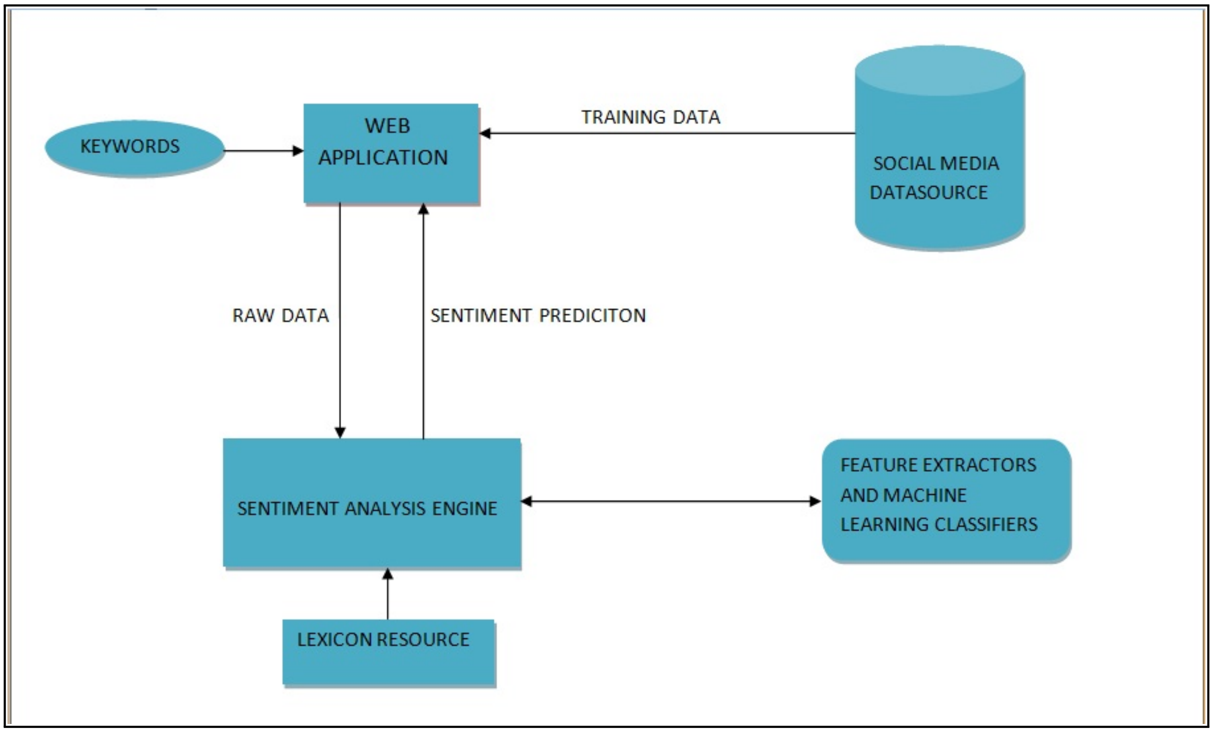

Figure 4.1: The Architecture Diagram Of System.

4.2.2 Data Flow Diagram

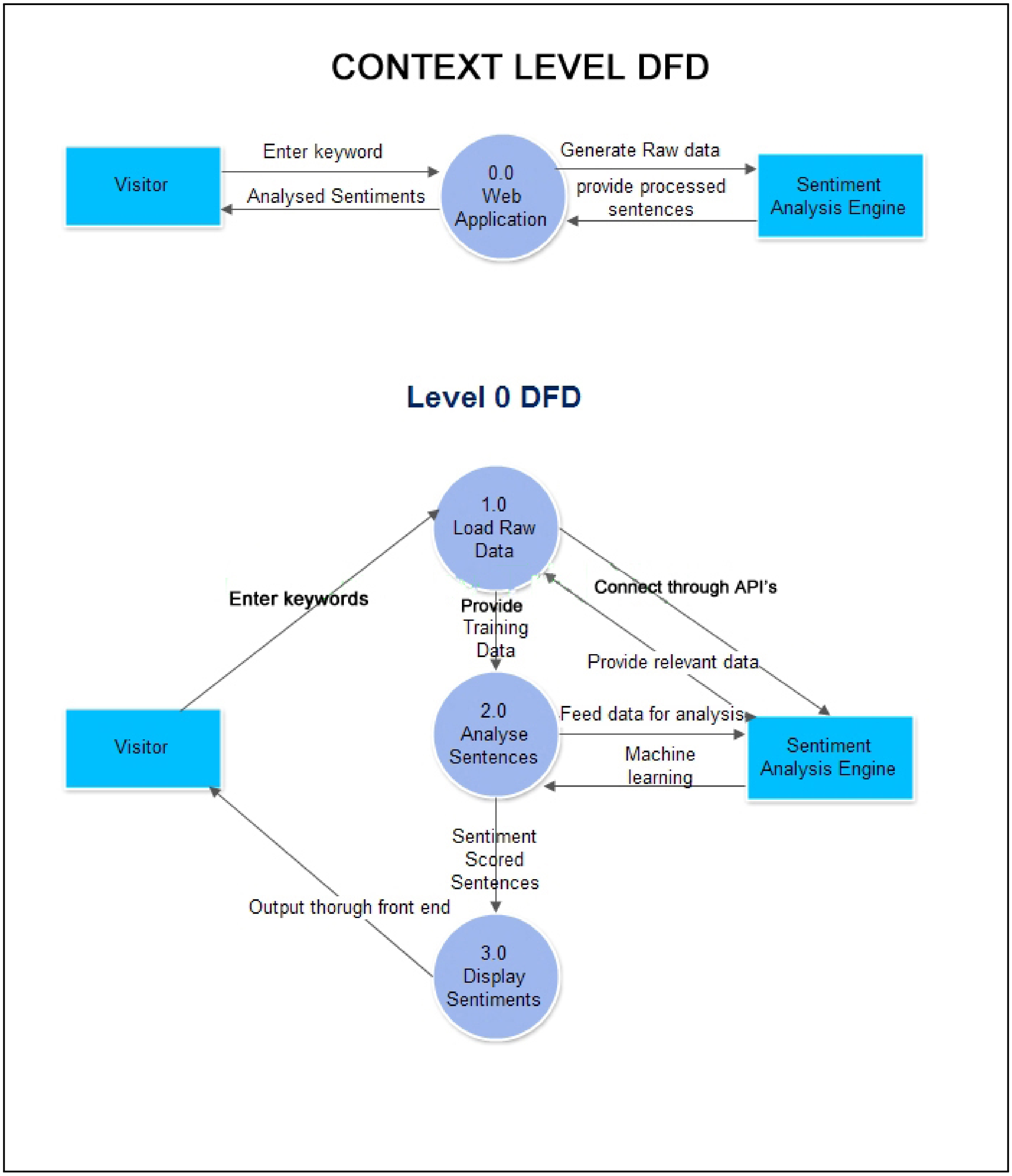

Figure 4.2: Context and level 0 DFD

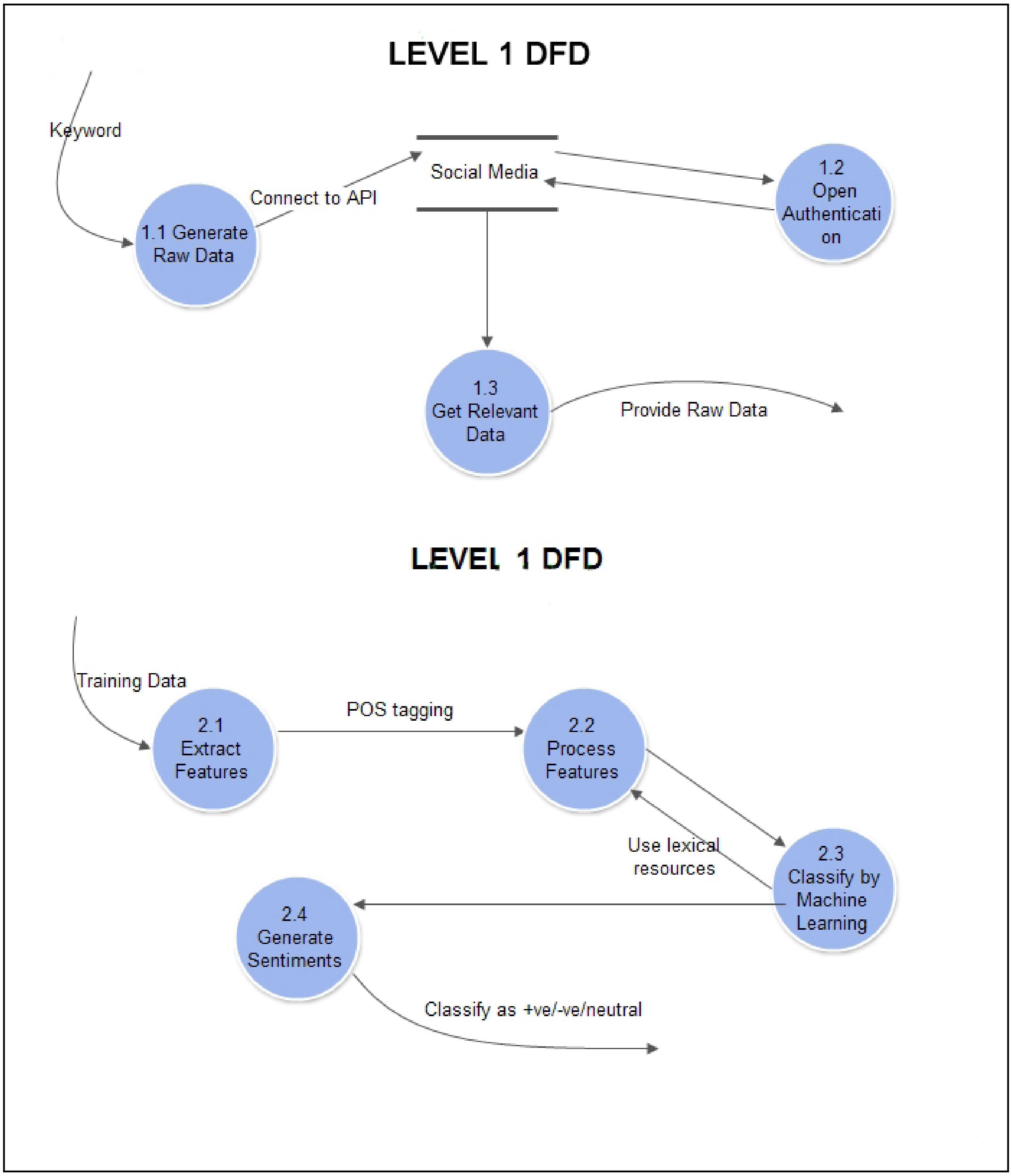

Figure 4.3: level 1 DFD

4.2.3 Activity Diagram

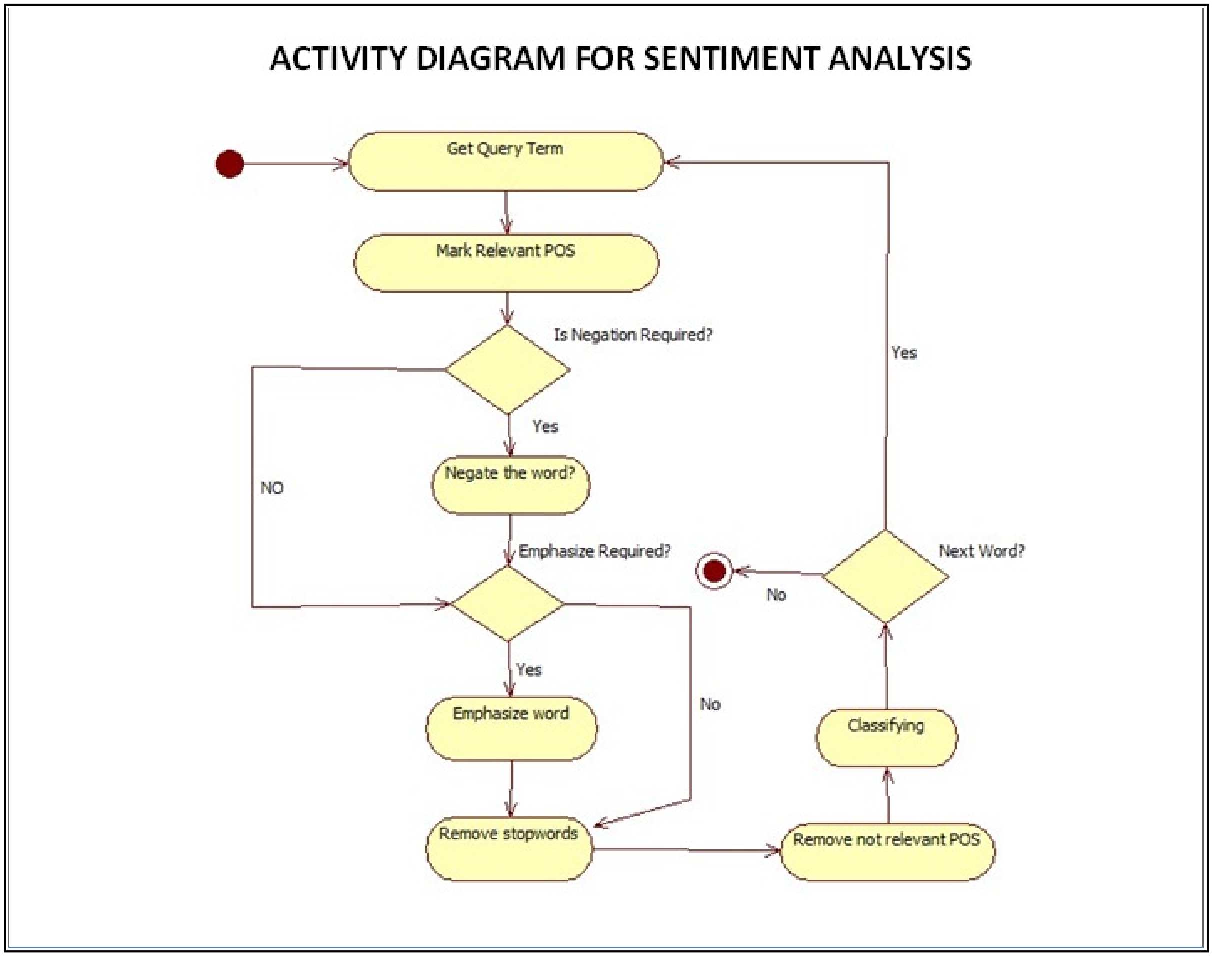

Figure 4.4: Sentiment Analysis Engine.

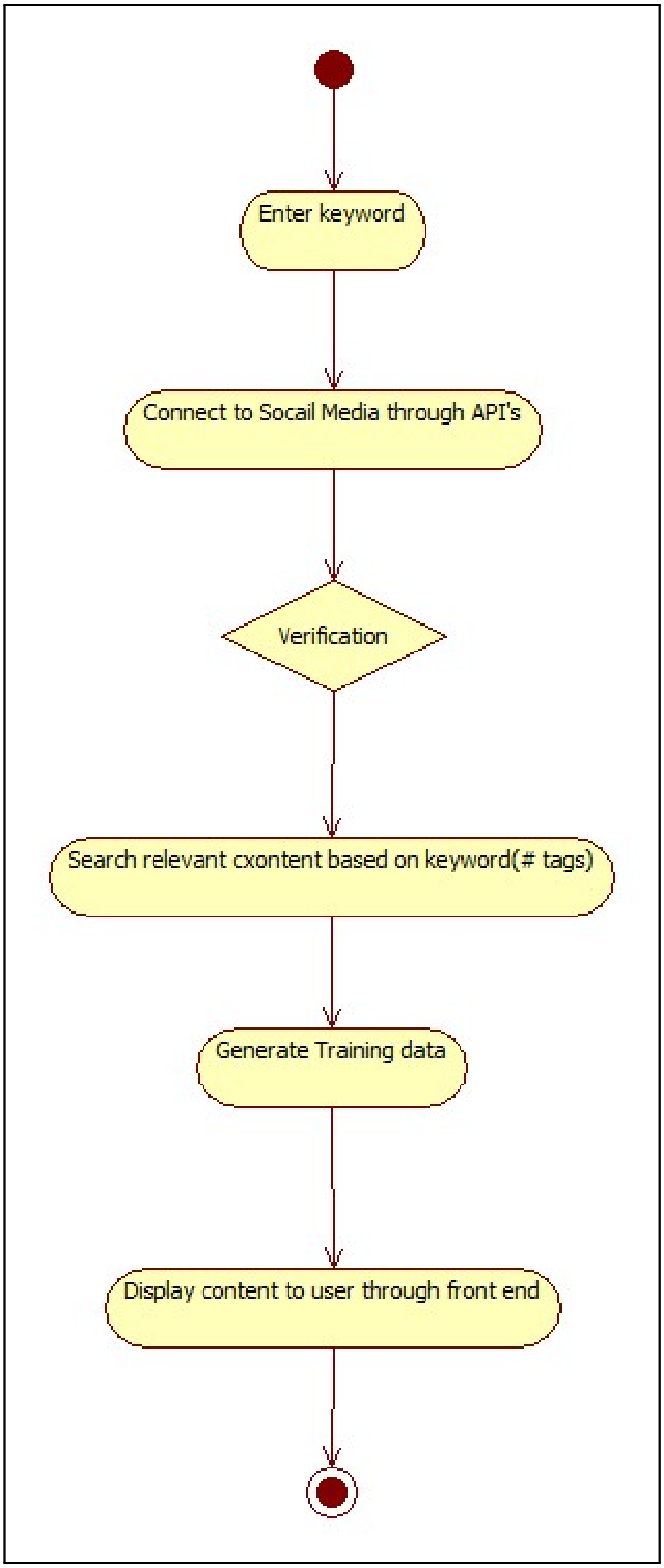
Enter keyword
Connect to Socail Media through API's
Verification
Search relevant cxontent based on keyword(# tags)
Generate Training data
Display content to user through front end

Figure 4.5: Data Fetching.

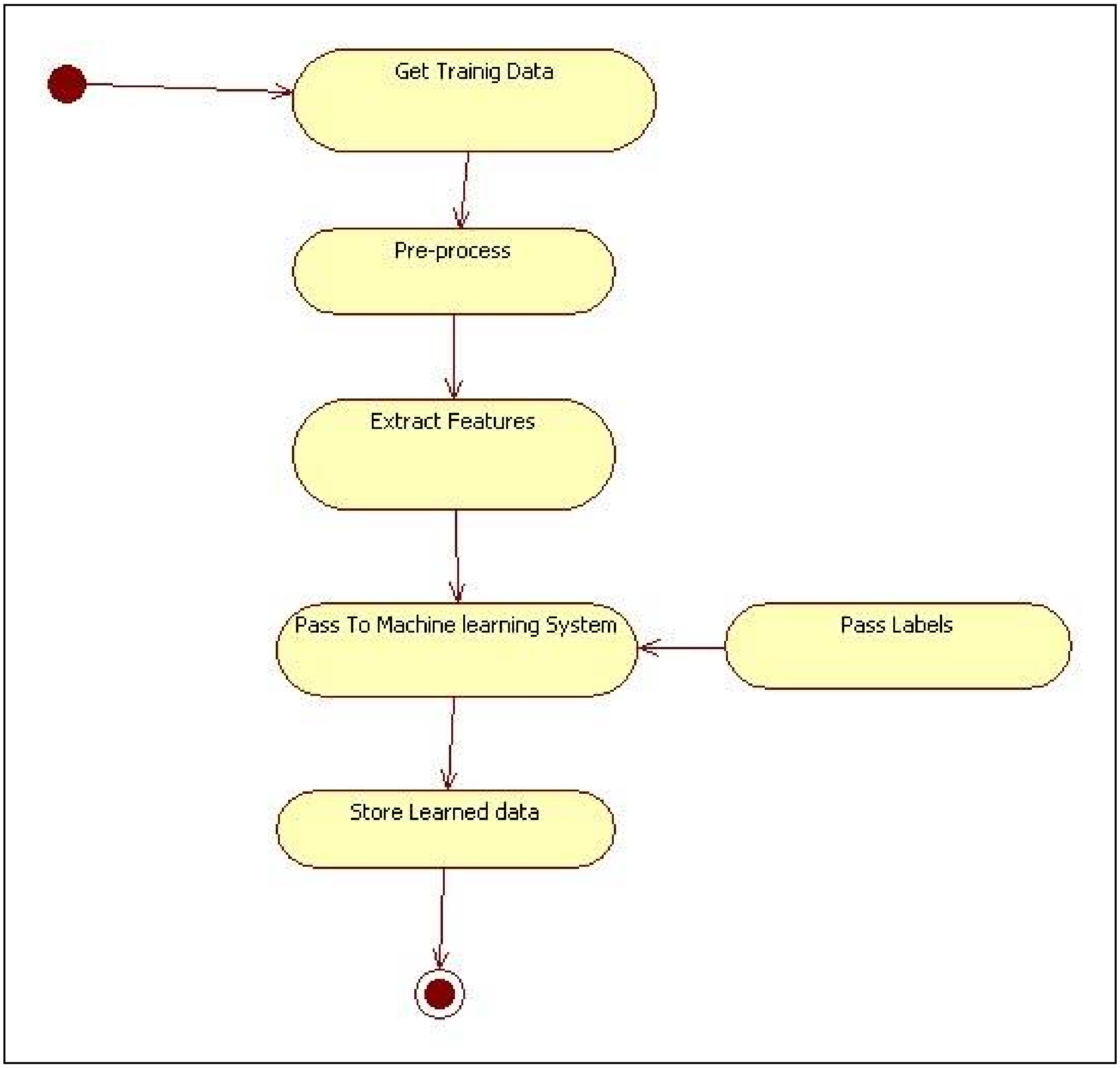

Figure 4.6: Training Data to Machine Learning Engine.

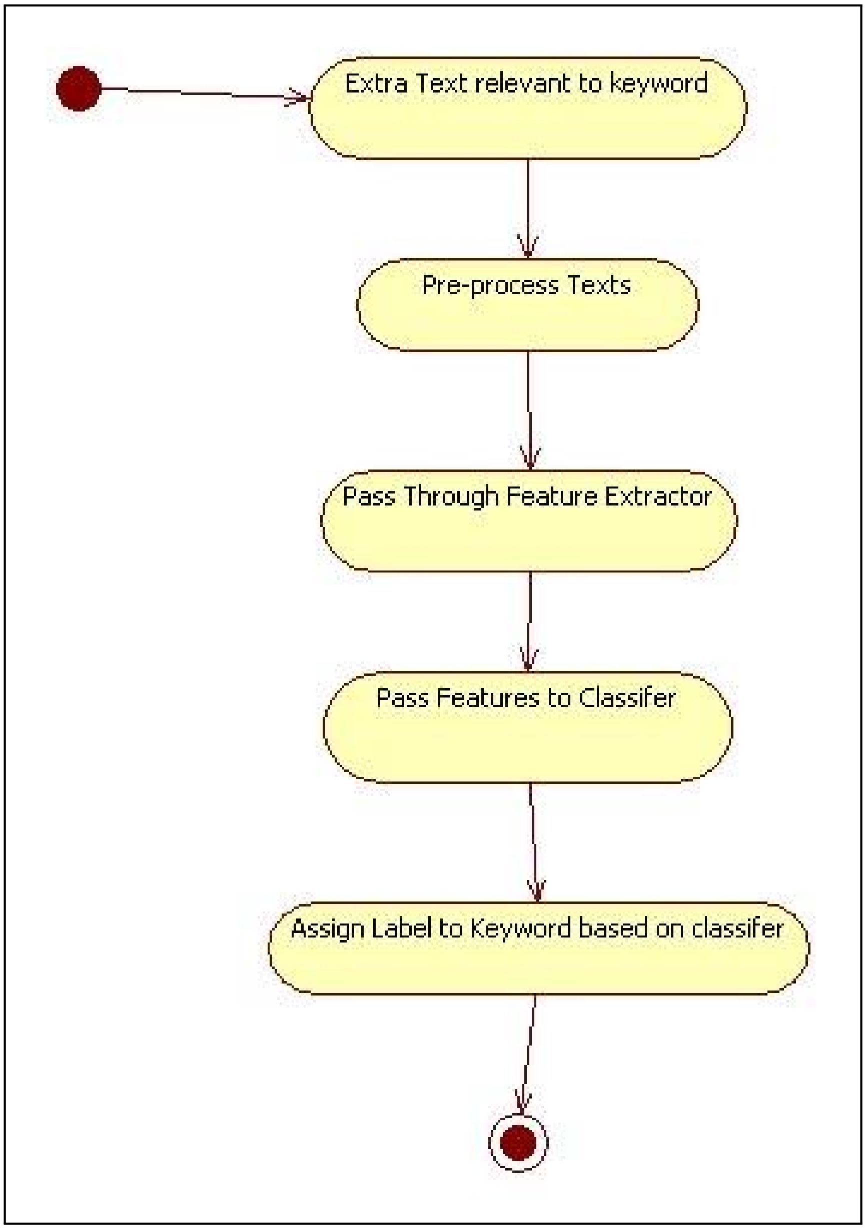
Extra Text relevant to keyword
Pre-process Texts
Pass Through Feature Extractor
Pass Features to Classifer
Assign Label to Keyword based on classifer

Figure 4.7: Test Data To Classifier.

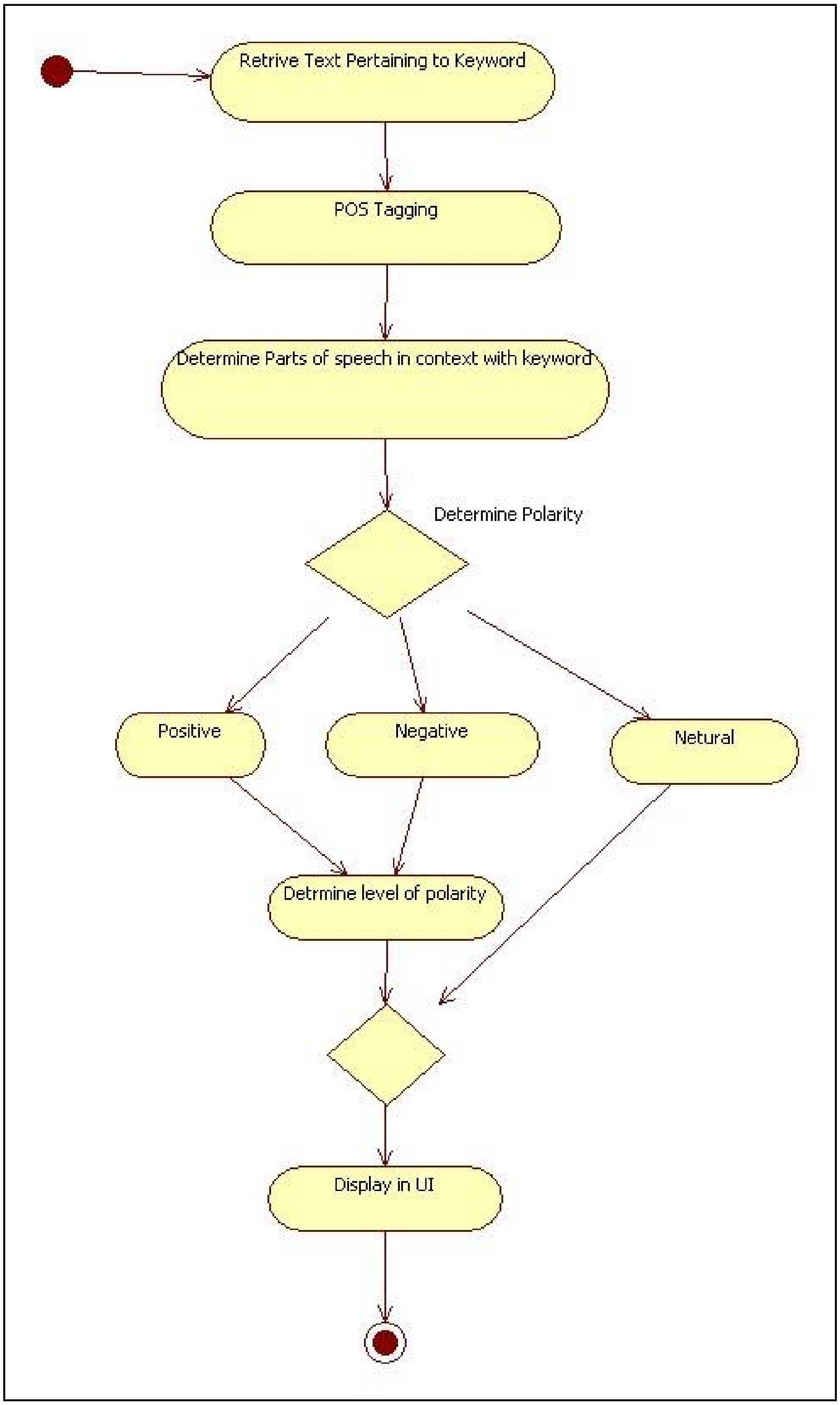
Retrive Text Pertaining to Keyword
POS Tagging
Determine Parts of speech in context with keyword
Determine Polarity
Positive
Negative
Netural
Detrmine level of polarity
Display in UI

Figure 4.8: Polarity Detection.

4.2.4 Sequence Diagram

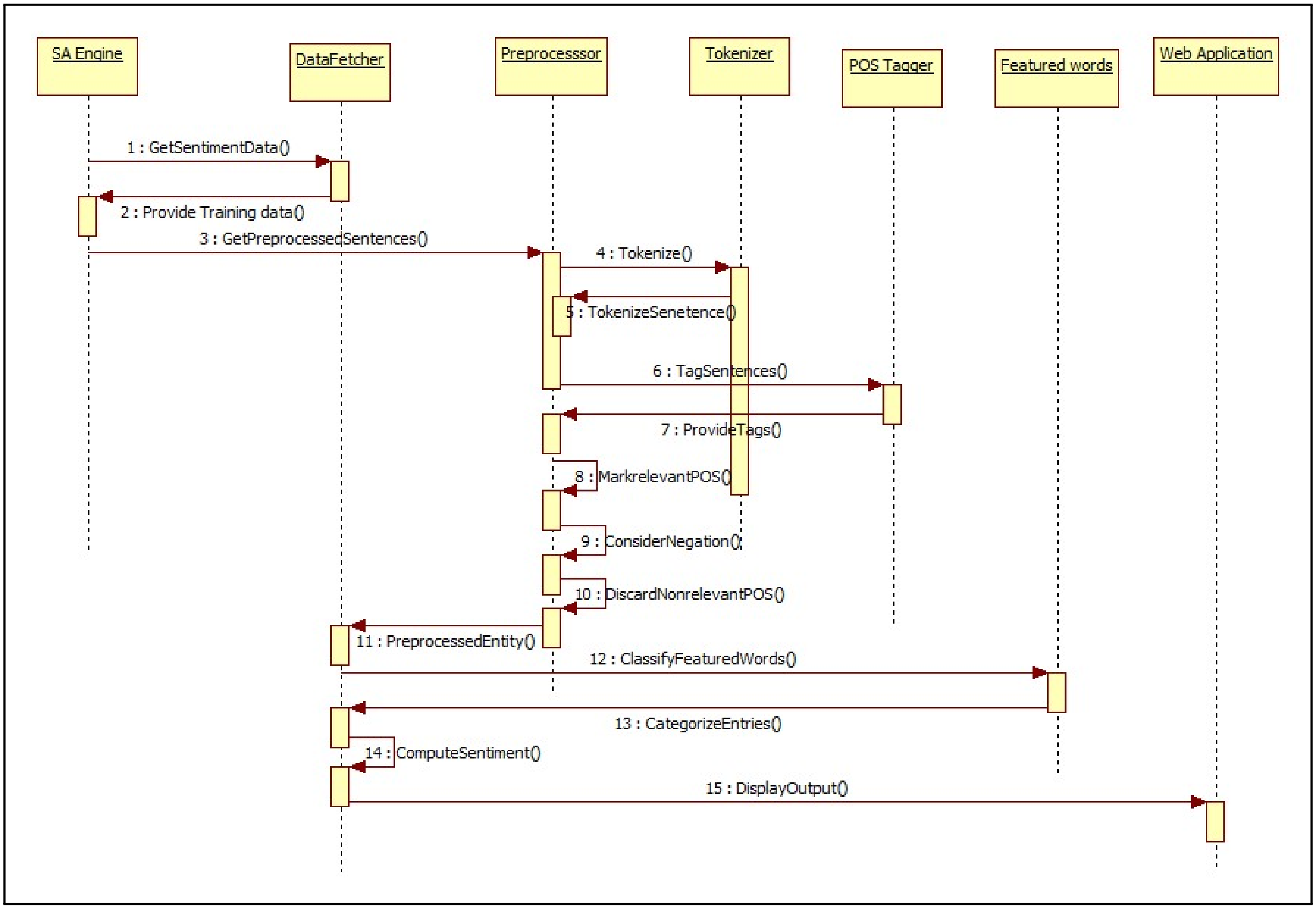

Figure 4.9: Overall Sentiment Engine.

Chapter 5

Implementation Details

5.1 Status

- **Selection of Dataset APIs :** The apis for extraction of datasets from social media sites have been searched and finalized. For twitter Twitter4j[7]and for facebook Facebook4j[8]will be used for the implementation of the project.

- **Development environment set-up :** Project will use eclipse IDE as an execution environment for development of our project.

- **Selection of Programming Language :**Python plus NLTK(Natural Language Tool Kit)[9] has been elicited against the available choices of Stanford core libraries for NLP[10] and Apache openNLP[11].

- **User Interface :** Implemented a basic front end using FLAT UI Tool-kit[12]

- **Building Dataset and Extraction Module :** Project has successfully implemented text extraction based on keyword from twitter media sets, program prompts for the keyword to search for tweets and outputs tweets which has the keyword in it. This is a very important step as this data is what will be feed to the sentiment analysis classifiers to get the needed sentiments of the queried keyword.

- **UML Modelling :** Visualization of the system and made the required UMl diagrams of the project namely activity, sequence, DFD, and architecture diagram has been done. This diagram will help the reader/evaluator to understand the project better.

Chapter 6

Conclusion And Summary

After a thorough literature review and a detailed understanding of the project, the documentation has reached the advanced stages and the implementation has been given a head start. The UML diagrams and the architecture of the system have been designed to provide an abstract conceptual overview of the whole system. The initial implementation involves extraction of data from twitter by using the twitter4j API as well as a basic theme for the front end. The machine learning algorithms and the tools to be used have been selected from a final short-list. In the following stages, the project is anticipated to include other social media datasets (e.g. Facebook), classification of that data using improvised machine learning algorithms and feature extractors that have been selected. The final implementation incorporates the outputs from the sentiment analysis engine by harnessing the power of natural language processing and displaying the polarized sentences in a user friendly format.

Bibliography

[1] B. Liu, “Sentiment analysis and subjectivity,” *Handbook of natural language processing*, vol. 2, p. 568, 2010.

[2] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[3] L. L. Bo Pang and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceeding EMNLP ’02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10 Pages 79-86*, 2008, pp. 81–82.

[4] B. Liu, *Web data mining: exploring hyperlinks, contents, and usage data.* Springer, 2007.

[5] Z. Markov and D. T. Larose, *Data mining the Web: uncovering patterns in Web content, structure, and usage.* Wiley. com, 2007.

[6] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N Project Report, Stanford*, pp. 1–12, 2009.

[7] Y. YANAMOTO, “Twitter4j: A java library for the twitter api,” 2011.

[8] “Facebook4j: A java library for the twitter api.” [Online]. Available: http://facebook4j.org/en/index.html

[9] Natural language toolkit. [Online]. Available: http://nltk.org/

[10] The stanford natural language processing group. [Online]. Available: http://www-nlp.stanford.edu/

[11] The apache opennlp library. [Online]. Available: http://opennlp.apache.org/

[12] Flat ui tollkit: Free user interface kit. [Online]. Available: http://designmodo.github.io/Flat-UI/