Proposed System Analysis and Design

The proposed system involves several sub-tasks.

1. Data collection:

Data in the form of raw tweets is retrieved by using the Tweepy library in python which provides a package for real time twitter streaming API. The API requires us to register a developer account with Twitter and fill in parameters such as consumer Key, consumer Secret, access Token access, and Token Secret. This API allows to get all random tweets or filter data by using keywords. Filters supports to retrieve tweets which match a specific criterion defined by the developer. We used this to retrieve tweets related to specific keywords which are taken as input from users.

2. Data Processing:

Data processing involves Tokenization which is the process of splitting the tweets into individual words called tokens. Tokens can be split using whitespace or punctuation characters. It can be unigram or bigram depending on the classification model used. The bag of-words model is one of the most extensively used model for classification. It is based on the fact of assuming text to be classified as a bag or collection of individual words with no link or interdependence. The simplest way to incorporate this model in our project is by using unigrams as features. It is just a collection of individual words in the text to be classified, so, we split each tweet using whitespace. The next step in data processing is normalization by conversion of tweet into lowercase. Tweets are normalized by converting it to lowercase which makes its comparison with a dictionary easier.

3. Data Filtering:

A tweet acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, these tweets are further filtered by removing stop words, numbers and punctuations. Removing non-alphabetical characters: Symbols such as "#@" and numbers hold no relevance in case of sentiment analysis and are removed using pattern matching. Regular expressions are used to match alphabetical characters only and rest are ignored. This helps to reduce the clutter from the twitter stream. Stemming: It is the process of reducing derived words to their roots. All the data filtering part is done using Text Blob library in python.

4. Feature Extraction:

TF-IDF is a feature vectorization method used in text mining to find the importance of a term to a document in the corpus. This feature is useful for a case where we need to find trending topics or to create word clouds. However, this project is more focused towards finding sentiment in twitter streams so TF-IDF is not implemented.

5. Sentiment Analysis

Sentiment analysis is done by using custom algorithm which finds polarity as below. Finding polarity: For discovering the polarity, we used Text blob function which applies a simple algorithm of counting positive and negative words in a tweet. For both, positive and negative words, different lists were made. Next step is to compare every word in a tweet against both these lists. If the current word matches a word in positive list, then a score of 1 is incremented and if a negative word is found then it is decremented.