CricTwee – Tweet Analysis for the Game of Cricket

Kunal Parakh
CS Graduate Student / USC
kparakh@usc.edu

Preetam Shingavi
CS Graduate Student / USC
shingavi@usc.edu

1 Introduction

The project focuses on applying NLP techniques to do analysis of relevant, unstructured data of tweets on a given match day for a game of cricket. The game of cricket itself makes the project interesting. There were around 62 million tweets in the last week about the game. Around 101.7 million people watched the game and many of them expressed their opinions and emotions about the game on Twitter. Twitter also has official hash tags for IPL (#IPL, #IPL2015) and attracts a lot of people all around the world because of the buzz of the game.

Our solution can be used by marketing companies to increase user engagement on the targeted website. The researchers of Human Behavior can make a difference by analyzing the pattern changes in the human reactions during the various events of the game. The business analysts can monetize the trends to increase the website traffic. The International Cricket Council (ICC) can broadcast the results on television and websites to engage more users.

The major challenge which we faced was annotating the tweets because of their unstructured and semi-formal nature and because of our different angles to look at a tweet as two different people can annotate a particular tweet differently based on the individual nature. We had to do a lot of manual analysis in order to get the results evaluated. Evaluation of models was difficult as every match presented different set of data.

2 Related Work

There is a lot of work done on Tweet classification into sentiments, but those works mostly include positive, negative and neutral sentiments. But, here we have proof of concept that multiclass classification with all the five classes does comparable work with the four models. We came across several papers that talk about summarization of the tweets. We modified the concept a bit by including our own gazettes and boosting the tweet scores further up.

2.1 Citations

Tweepy Documentation[1] helps to understand the APIs provided for collecting Twitter data. The Advantages of Careful Seeding[2] explains the benefits of using k-means++ over k-means as the prior one takes into account pre-defined cluster centroids. The paper "Summarizing Sporting Events Using Twitter"[4] specifies methods for tweet summarization which takes into account chunking of the tweets based on time stamp and summarizing each chunk. On top of that we added our average function to average out all the tweets from the chunks to find a threshold value above which a chunk is considered as a peak valued chunk. Only from these chunks we summarized top scored tweets to give better summary.

3 Data

3.1 Collection

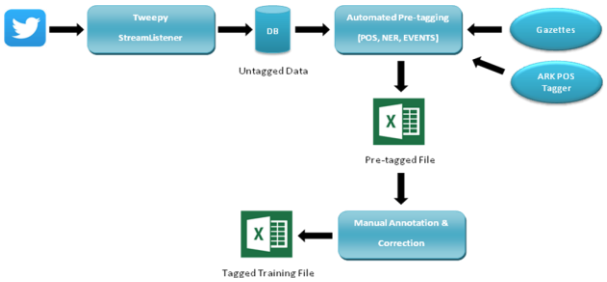

Figure 3.1. Data Collection and Processing

Origin:
The base origin of data (here tweets) is twitter. Tweets had to be captured from twitter at real time when the match began. Option was to use Streaming API of twitter and ingest the tweets. We used a wrapper project of this Streaming API (Tweepy) to build our system to capture tweets on real time basis. See references.[1]

Other data set required was for the gazettes used. The data for NER (Players, Location, Teams, Venue, Stadium) were scrapped manually from the official website of Indian Premiere League 2015. Further known events were formed a part of gazettes and domain experts (here self) made available events for the game of Cricket for IPL game format.

Size and other details:
Tweets of 3 different matches were captured in 3 different buckets (Match_1, Match_2, Match_3). The size of each bucket is around 8,000 to 10,000 tweets. The tweets were only filtered for English language using the API configuration. The buckets were further sub-divided in data store containing maximum of 200 tweets each just to be safe if any debugging would be required on raw data in future. Each tweet has all the attributes like tweet Id, time stamp, tweet text etc available in JSON format in the data set.

3.2 Annotation

Tweets from raw text JSON had to be converted to an intermediate format which could be a seed to start manually annotating data for the tasks in the objective. Pre-tagging the data set is always a tough and intuitive task, if done intelligently, would help manual tagging go faster.

We parsed the JSON tweets in order of time stamp and dumped them to excel file. Before dumping we tokenized and applied ark based POS tags for each tweet. Furthermore each tweet was pre-tagged with the events and NER from the gazette as mentioned above.

The dump for each tweet in excel file had the following parts :
Line-1 Tweet tokens found by ARK tokenizer with the tweet id prefixed
Line-2 Empty line for manual tagging a sentiment class (Default all value - 5 which is neutral)
Line-3 POS tag from ARK for each token in respective column of the token on the tweet's token row
Line-4 Events if any in the first column recognized by the gazette

4 Technical Approach

4.1 Sentiment Analysis

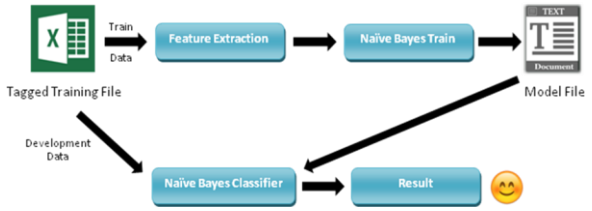

Figure 4.2.1. Sentiment Analysis Flow

Feature Extraction -

587300957345619968	The	beast	man	comes	un	Pollard	fear	#ipl
	9		9	9				
POS	D	N	N	V	G	^	V	#
NER	O	O	O	O	O	B-PER	O	B-HASH

Figure 4.3.2. Annotated Data

We defined our own sentiment classes as **Unpleasant, Sad, Neutral, Happy, Ecstatic**. Before forming a training set, we took out some of the words(marked as yellow in above figure) like proper nouns, determiners which really don't contribute to the classification task. As we have followed Naive Bayes approach, the keywords are added in the bag of words feature set and the model is trained on that data. We had round about 1000 tweets given as a training dataset.

A development data was then provided along with the model file to the classifier to get sentiment classified tweets. The results were then post processed to generate a JSON to be given to the Data Driven Document object in order to get the graphical representation of the results. We tested our results using four approaches - **Megam Multiclass Classifier, Naive Bayes Classifier, NLTK Naive Bayes Classifier and Bi-gram Naive Bayes Classifier** and then formed the evaluation matrix.

4.2 Named Entity Recognition

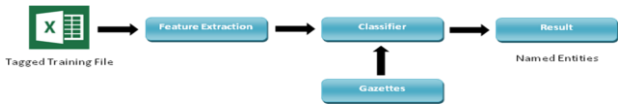

Figure 4.2. Named Entity Recognition Flow

The features used for NER task were fetched from the pre-tagged training file for Named Entities. We defined our own named entities as Person, Location, Team and Venue. The tagged named entities from the training file were given as input to the classification module and on top of it, we defined Gazettes for all the four named entities which consist of all the player, location, team and venue names along with the acronyms and pet names of the players. The classification module then gave the separate named entities for all the four classes. The output was processed to form a JSON file which was then given as an input to Data Driven Document module for graphical representation.

4.3 Clustering

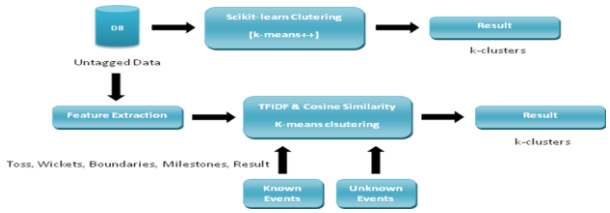

Figure 4.3. k-means++ Clustering

To explore something interesting in an unexplored data set was the aim of clustering. Further refining the aim to cluster using NLP techniques on known set of events that were pre-tagged. Our own k-means clustering with initial seeds to known event tweets :
Here we build our own k-means clustering algorithm based on centroid model clustering and initialization using known k-events.

Initialization using seed tweets :
We had tagged the tweets with different events (TOSS, BOUNDARY, WICKET, RESULT, MILESTONE). We randomly select 1 tweet from each event and form a initialization cluster with centroids mapping to the vector representation of the tf-idf values of tokens filtered.

Features are extracted to remove stop-words, and few tokens matching POS tags (!, ', #, @, P). This feature representation form a bag of words and we created a tf-idf vector representation for each tweet and applied the k-means algorithm with the above initialization setup. For similarity we calculated the cosine distance between the tweets and matched tweets to nearest clusters. The convergence of the algorithm took place around at 6th iteration for around 8000 tweets and k=5 (number of unique events).

Off-the-shelf technique to explore random clusters:
We used scikit learn k-means++ module[3] to visualize the data for clusters. The clusters start at unknown seeds, so it's difficult to predict the evenness of the tweets in one cluster. There were few interesting clusters explored which formed clusters on some named entities, some were on events too while few were random with some similar features.

4.4 Summarization

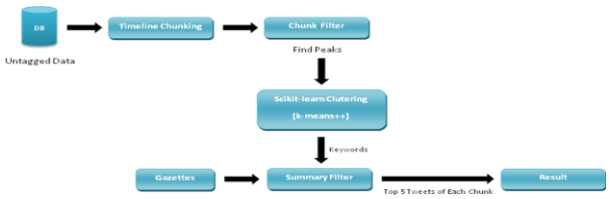

Figure 4.4. Summarization Flow

Various summarization techniques have been used to summarize a event based on social context available. But for the game of cricket very few have tried to explore. We referred a paper "Summarizing Sporting Events Using Twitter" by Jeffrey Nichols, Jalal Mahmud, Clemens Drews.[4] The approach is based on finding the peaks in the frequency of tweets and applying summarization technique only on the tweets in the window of peaks. Finally merging all the intermediate summaries to form a match summary.

Initially we read the raw dump of tweets as explained in the data collection, and formed timeline model of tweets. The tweets here were bucketed in a bucket size of "m"-minutes. Each "m"-minute bucket has some number of tweets

which vary from bucket to bucket. We selected heuristically the value m = 3 min bucket. Thus data was chunked on timeline series with 3-minutes bucket.

Next task was to identify the peaks in the time lined buckets, this was found based on average frequency of tweets as threshold and mark only those buckets that have number of tweets more than this threshold. The resultant was a array of consecutive-chunks each consecutive chunk depicting something exciting happening. That's the reason more users have tweeted during that time. The tweets from each consecutive-chunk are considered as a data-set for clustering on known events (TOSS, BOUNDARY, WICKET, RESULT, MILESTONE). Finding which event has really occurred. This was done using k-means with initial seeds as above method explained and then applying scores to those tweets that are more relevant to the maximum frequency words(add +1 since it has some relevancy to the cluster) and the known events (add +5 to score since more specific to an event). Then rank the tweets based on the final score and take the top-5 tweets to summarize the event happening in the consecutive-chunk selected.

This was repeated to all the consecutive-chunks in the array as mentioned above. Finally all the results were combined to generalize the summary of the match. Observations were interesting to find events like TOSS at the start, wickets, mile-stones during the course of match and then a result towards the end with the man-of-match.

5 Evaluation and Analysis

5.1 Sentiment Analysis

Initially we had started with 9 sentiment classes and ended up with accuracy around 60%. So, we decided to merge the classes and finalized 5 sentiment classes. We followed an incremental approach by training the dataset on a chunk of 200 tweets at a time and classifying the tweets to identify trends in the accuracy level. Our baseline for sentiment analysis was results from the Naive Bayes classification for the first assignment. We did comparatively well with the accuracy as the data here was unstructured. We wrote the accuracy calculation script and tested the accuracy on development dataset. Following is the Evaluation Matrix -

Method	Approximate Accuracy
Megam	76%
Naive Bayes Classifier	72%
NLTK Naive Bayes Classifier	72%
Bi-gram Naive Bayes Classifier	58%

Table 5.1. Evaluation Matrix for Sentiment Analysis

5.2 Named Entity Recognition

We evaluated the classified named entities by crosschecking them with our predefined gazettes for Person, Location, Team and Venue. The gazette files were formed after collecting data from the official IPL site[5]. While evaluating named entities, the major challenge was to consider short names and acronyms as well. We did fairly enough to consider all these scenarios while forming gazettes.

5.3 Clustering

Evaluation of Clusters formed was more of an exploratory task. We went through several clusters and categorized our results based on the text similarity of the clustered tweets. Relevancy of results was cross checked with our pre-defined event based clusters. Sometimes, the clustered data is not relevant to our interest as it is purely based on text similarity.

5.4 Summarization

Evaluation of the summaries generated was based on the manual analysis and comparing the summaries with the online sport featuring channels like cricbuzz.com[6].

Summarization is based on similarity between the tweets and their scores are based on the events. So, we did find some irrelevant summaries from the clusters. The summaries are given as a set of five highly relevant tweets. Our future work involves summarizing these tweets into a one line summary using Parse Graphs.

6 Results

6.1 Sentiment Analysis

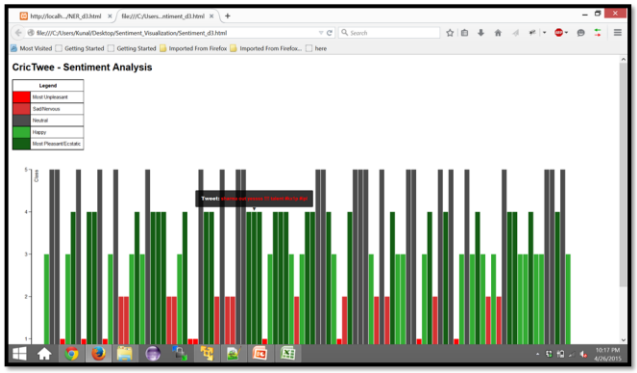

Figure 6.4. Tweets classified into Color-coded Sentiment Classes

6.2 Named Entity Recognition

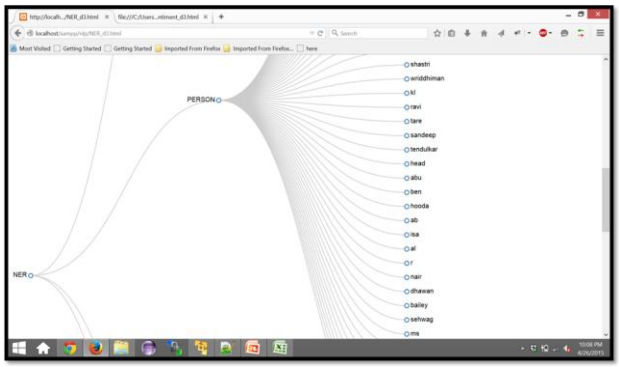

Figure 6.2. Named Entities classified into Person, Location, Venue and Team

6.3 Clustering

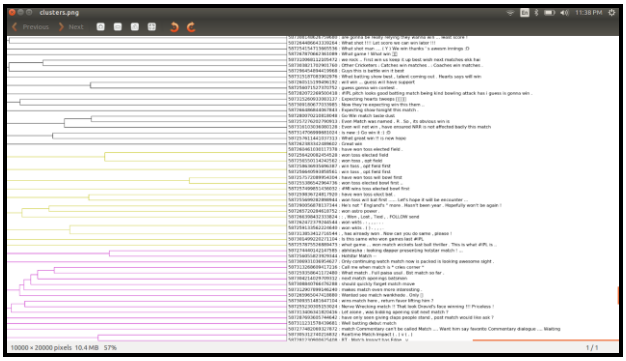

Figure 6.3.1. Clustering shows all "Toss" related tweets clustered together

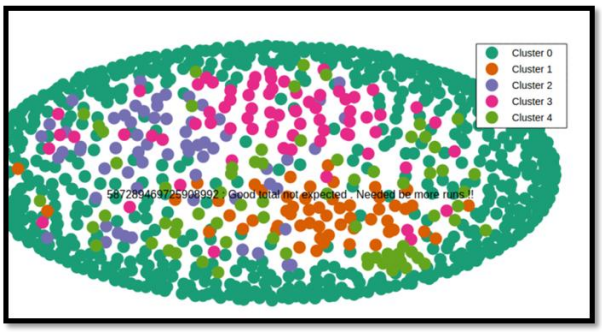

Figure 6.3.2. scikit clustering shows all the tweets clustered into 5 clusters

6.4 Summarization

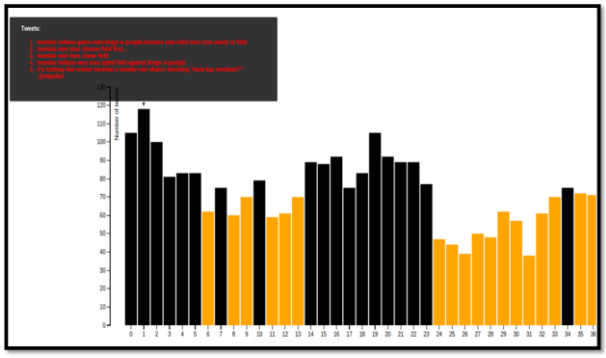

Figure 6.4. Summarization of Tweets based on first five highly scored tweets from each chunk

7 Contribution

All the algorithms and methods were discussed and formalized together and implemented individually, so both of us are aware of all the details and flows of the project. We both contributed equally in the Poster and Report preparations.

Individual Contributions -

1.Kunal Parakh - Sentiment Analysis, Named Entity Recognition, Results and Evaluation.

2.Preetam Shingavi - Clustering, Summarization, Results and Evaluation.

Combined Contributions -

Data collection, pre-processing, pre-tagging and manual annotation.

Link to the Public Repository -
https://bitbucket.org/shingavi/csci-544-project-cricktweetsanalysis

8 References

1. Tweepy Documentation
   http://tweepy.readthedocs.org/en/v3.2.0/
2. The Advantages of Careful Seeding
   David Arthur | Sergie Vassilvitskii
3. http://scikit-learn.org/stable/modules/clustering.html
4. Summarizing Sporting Events Using Twitter –
   Jeffrey Nichols, Jalal Mahmud, Clemens Drews
5. http://www.iplt20.com/
6. http://www.cricbuzz.com/