Popularity prediction with Combined Social Network analysis for Sports Domain

Charan Thej Aware
caware1@asu.edu

Vikranth Doosa
vdoosa@asu.edu

Tushar Jain
tushar-jain@asu.edu

1. Introduction:

In the domain of sports, fans are strongly connected to Social Media like Facebook and Twitter. People love to comment about their favorite teams, players and ongoing matches on the social media. Not only the fans but also the players, tournament conducting authorities, sponsors, franchises owing the teams participating in the sporting events, top brands and news organizations are also members of these sites and share the updates almost continuously.

Even though the underlying structure of the Facebook and Twitter is different, they share the same idea of containing the opinions of their users on the teams, players or events going on in the world of sports.

We attempted to cluster the data from the two social network websites and perform a combined analysis on the clustered datasets to estimate the sentiments of the topic solely based on the user interest. We further used a formula to calculate the popularity of the posts which will list the popular trends from among the groups the user is following.

2. Problem Description:

Given	:	Two social network datasets (F- Facebook and T- Twitter)
Find	:	Similar links between the two datasets by mapping them together and perform clustering and then applying sematic logic to calculate popularity of the trending events.
Domain	:	Sports (Cricket)

2.1 State of the Art Methods:

Sports are undoubtedly one of the biggest entertainment sources in the current trends of human life. With the advent of the Internet, especially the social networking websites, it has become very easy for the users to get any information related to anything. Realizing this fact very early, all the sporting event authorities have been using the social networking sites to publicize their events and their teams. They create groups related to a particular event and post all the details of the events and teams online. Further the activities on the social media can be analyzed to gain insights about the fans likings and preferences which has tremendous applications with respect to marketing.

Facebook and Twitter topped the list of the social media websites which are being used by the sports authorities to share the information. In addition to providing the information to the users, these two websites are listing popular events/trends/topics being discussed online. Their listings are based on the volume of the users discussing the topic. The more the number of users post/share/comment on a topic, the higher will be its popularity. The current applications can also list the popular trends region wise taking into consideration the locality of the user.

Apart from that, a few online tools are available already over the internet which can provide such analysis over the social media. IProspect [1] does the same thing and perform sentiment analysis over many well-known social media websites. Google prediction API [2] does a similar job, rather performing prediction on social media, they provide a tool to create a supervised model and make predictions. Repustate [3] is another website that provides sentiment analysis as a web service.

While all the above mentioned online websites are good at performing sentiment analysis, none of them explicitly mentioned of doing such analysis on a collaborative manner taking named entities as grounds for clustering the two or more datasets.

2.2 Out of the box approach:

The current applications:

a. Does not list the popular trends relative to the user. Listing out a generalized list of popular events may not be of much use or interest to a person if that list does not contain any event/topic which the person is interested in.
b. Does not perform the combined popularity prediction by integrating the similar topics from both these social networks.

The Facebook is a public-user driven website (public users with a lot of user comments/likes on any post made by the other users) whereas the Twitter is verified-user driven (posts are from professionals of different fields and the public users like or retweet their favorites tweets). According to research published by David Murphy, around 44 percent of user accounts have never tweeted [4]. These two websites are listed in the top 100 most popular websites worldwide according to Alexa Internet [5]. Therefore combining these two datasets can provide the results with much more accurate score of sentiment and popularity analysis.

We tried to integrate the datasets from the verified-user driven network (Twitter) acting as the prime data sources of the topics/events and interests of the users and the datasets from the public-user driven network (Facebook) acting as the reflections of the views of the users on those topics/events and interests. We estimated the personalized popularity prediction, solely based on the interest of the particular user within a time frame of one day. We implemented the technique on the sports domain, especially on the sport of Cricket, the world's second most popular sport [6].

3. Data Mining:

We used Facebook and twitter social graphs as the sources of the data for our implementation.

3.1 Twitter Datasets:

The version 1.1 of the Twitter's REST APIs provides programmatic access to read and write Twitter data. A user can author a new tweet, read other user's profile and follower data, and much more. The REST API identifies Twitter applications and users using OAuth, which sends secure authorized requests to the Twitter API to fetch the data [7] [8]. All the responses are available in JSON format.

Attributes primarily used from the Twitter API:

screen_name : The screen name of the user for whom to return results for. Used to the get the timelines of a user consisting of the tweets, retweets, favorites, user mentions, hash tags etc

id : The tweet ID

id_str : The tweet ID in string format

created_at : UTC time when this Tweet was created.

count : Specifies the number of tweets to retrieve, a maximum of 200 per distinct request.

since_id : Returns the tweets with an ID greater than the specified ID. This is used to recursively extract the tweets from the user timeline, 200 at a time.

max_id : Returns the tweets with an ID less than or equal to the specified ID. This is used to recursively extract the tweets from the user timeline, 200 at a time along with since_id.

text : The actual UTF-8 text of the tweet

user : The user entity object who posted this Tweet. This object has many attributes

favorite_count : The count of the likes to a particular tweet.

retweet_count : Number of times a tweet has been retweeted

Entity object:

Entities like user, tweets, retweets have sub-attributes which gives structured data from tweets without having to parse the tweet text for those details.
Examples are urls, user_mentions and hashtags.

user_mentions : An array of Twitter screen names extracted from the Tweet text. It has attibutes like id, screen_name, name.

hashtags : An array of hashtags extracted from the Tweet text. It has attributes like text, indices urls : An array of URLs extracted from the Tweet text. It has attributes like url, display_url

Extracting tweets:

- Using the screen_name of the user and the count parameters, we first extract 200 tweets using the first request.
- From the database of the tweets we get the Tweet id with the least value (Id of the last tweet), subtract a value of 1 to it and specify that value as max_id which will extract the next 200 tweets.
- We repeat the process till the created_at value of a tweet exceeds one day.

3.2 Facebook Datasets:

The Version 2.3 of the Facebook's Graph API is the primary way to get data in and out of Facebook's social graph. It's a low-level HTTP-based API that you can use to query data, post new stories, upload photos and a variety of other tasks that an app might need to do. To access the Facebook graph we need to generate an access token, which we can get either registering an app on the Facebook developer platform [9] or using Facebook Graph Explorer [10][11]. The responses are in JSON format. The underlying structure of the Facebook social graph is as follows:

Nodes – They are the objects such as a User, a Photo, a page, a comment
Edges - The connections between the objects, such as photo on a page, or a comment on a photo.
Fields – Attributes that describe the objects such as the birthday of a user, or the name of a page.

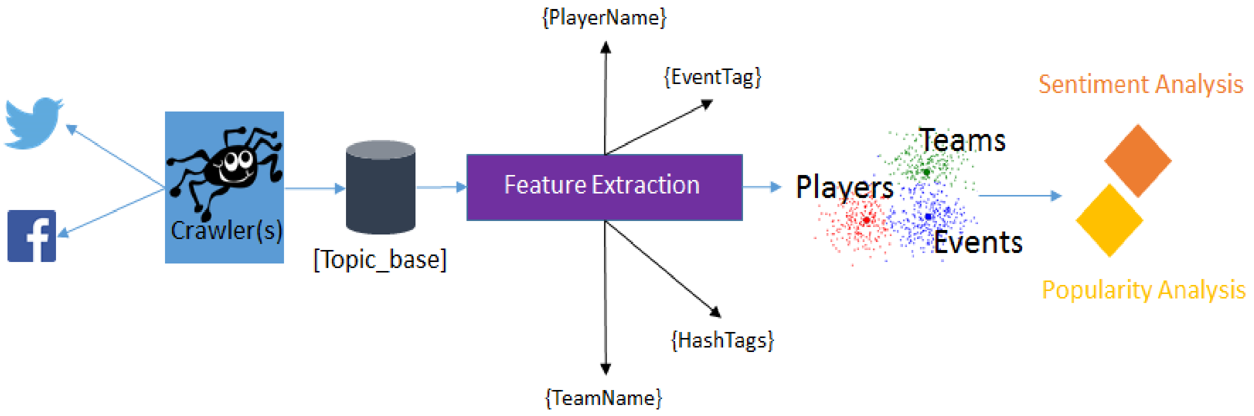
https://graph.facebook.com/v2.3/search?q=cricket&type=page

```
{
  "data": [
    {
      "category": "Sports league",
      "name": "ICC Cricket World Cup",
      "id": "505510849485097"
    },
    {
      "category": "Community",
      "name": "Cricket on Facebook",
      "id": "727565050588014"
    },
    {
      "category": "News/media website",
      "name": "cricket.com.au",
      "id": "85633169312"
    },
    {
      "category": "News/media website",
      "name": "Cricket Tracker",
      "id": "600228316674560"
    },
    {
      "category": "Sports team",
      "name": "Cricket South Africa",
      "id": "88795929350"
    }]
}
```

https://graph.facebook.com/v2.3/69553328633_10152886709678634

```
{
   {
        "category": "Sports team",
        "name": "Mumbai Indians",
        "id": "198358615428"
      }
    ]
  },
  "message":        "INTERV                "id":
"69553328633_10152886709678634",
  "from": {
    "category": "Sports league",
    "name": "IPL - Indian Premier League",
    "id": "69553328633"
  },
  "to": {
    "data": [
IEW: Have learnt from watching #Tendulkar, #Dravid
play: Simmons...Have to play every game as final
says    Mumbai    Indians    opener.    READ    -
http://www.iplt20.com/news/2015/features/6497/hav
e-learnt-from-watching-tendulkar-dravid-play-
simmons #MI #PepsiIPL",
  "message_tags": {
    "111": [
      {
        "id": "198358615428",
        "name": "Mumbai Indians",
        "type": "page",
        "offset": 111,
        "length": 14
      }
    ]
{
  "picture":
"https://scontent.xx.fbcdn.net/hphotos-
xpf1/v/t1.0-
9/s130x130/11188349_10152886709678634_53286793439
29741499_n.jpg?oh=22cf2b900035b117aad48ced8a4be49
d&oe=55C2B4E2",
  "link":
"https://www.facebook.com/IPL/photos/a.7825442863
3.80500.69553328633/10152886709678634/?type=1",
  "icon":
"https://www.facebook.com/images/icons/photo.gif"
,
  "actions": [
    {
      "name": "Comment",
      "link":
"https://www.facebook.com/69553328633/posts/10152
886709678634"
  },
    },
    },
    {
      "name": "Like",
      "link":
"https://www.facebook.com/69553328633/posts/10152
886709678634"
    }
  ],
  "type": "photo",
  "status_type": "added_photos",
  "object_id": "10152886709678634",
  "created_time": "2015-05-04T04:00:01+0000",
  "updated_time": "2015-05-04T04:02:42+0000",
  "is_hidden": false,
  "likes": {
```

4. Methodology:



4.1 Combining the datasets:

The common attributes from the Facebook API and Twitter API are taken and stored in the database. We used all the attributes mentioned above to perform the clustering and popularity predictions.

4.2 Entity/Feature Classification:

In the domain of sports, almost every sport consist of teams that participate in the tournaments conducted by the sporting event authorities. Each team comprises of players and the coaching staff. Each tournament will get its popularity based on many factors like number of teams participating in the tournament, popular players in each team, the entertainment level provided by each tournament.

For instance, the cricket tournament between two teams (countries) may not be as popular as the tournament like World cup where all the teams (countries) participate in the tournament. A tournament between popular rival teams will have a lot of expectations from the fans. A tournament that engages the best players of the sport will be popular among the fans. For instances the contest between one of the best batsmen Sachin Tendulkar and one of the best bowler Brett Lee will create a lot of hype among the fans.

Considering all the above factors, the main features of the sport can be classified as player, team and event (tournament).

4.3 Entity/Feature Extraction:

As the step of the entity extraction, we used the Stanford NER entity extraction tool to extract the named entities related to the player, team and tournament. The Stanford tool is a 3 class NER tagger that can label PERSON, ORGANIZATION, and LOCATION entities. It is trained on data from CoNLL, MUC6, MUC7, ACE, OntoNotes, and Wikipedia [12].

One of the prime factors we need to consider is that the tweets/retweets from the Twitter and posts/comments from the Facebook will not be both grammatically and syntactically complete. Users normally tend to use shortcuts, emoticons, non-dictionary words, abbreviations etc.

Both in Twitter and Facebook, the data will be represented primarily using Hashtags (#Hashtag) and User Mentions (@<Username>) in almost all tweets or posts.

- Hashtags represents any tournament, a particular match of a tournament, a tagline of a team, a player, a team etc.
- User Mentions represents the active user about whom the people are tweeting. It can be about a player account, team account, team owner's account, sponsors account etc.
- Emoticons represents a symbolic representation of the user opinion like sad, happy, excited, etc. They cover a wide range of feelings.
- Abbreviations that represent a wide range of emotions.

The NER tool extracted entities like person names, locations and non-sport organizations. But they are not correctly extracting all the entities related to the sports domain. The below examples are the output of the NER tool, when we ran it on the tweets from Twitter and posts from Facebook.

4.3.1. NER Features of tweet:

TEXT: See the wickets from day one with 4 for Matt Coles and 1 for @AdamRiley92: https://t.co/fPpkPHa0Mq

NER TOOL: See/O the/O wickets/O from/O day/O one/O with/O 4/O for/O Matt/PERSON Coles/PERSON and/O 1/O for/O @AdamRiley92/O:/O https://t.co/fPpkPHa0Mq/O

In this example, the NER tool did not extract the @AdamRiley92 which is the name of the player Adam Riley.

4.3.2. NER Features of tweet:

TEXT: Keaton Jennings wants Durham to press on against Middlesex: http://t.co/txywcKSBJi http://t.co/B5nrFcAXp2

NER TOOL: Keaton/PERSON Jennings/PERSON wants/O Durham/LOCATION to/O press/O on/O against/O Middlesex/ORGANIZATION:/O http://t.co/txywcKSBJi/O http://t.co/B5nrFcAXp2/O

In this example, the NER tool classified Middlesex as an organization, but in this context of this tweet, it is mentioned as a team name.

4.3.3. NER Features of Facebook post:

TEXT: Time for the most relaxing post-IPL day checks I've ever had!

NER TOOL: Time/O for/O the/O most/O relaxing/O post-IPL/O day/O checks/O I/O've/O ever/O had/O!/O

In this Facebook post, NER tool did not recognize the entity IPL, which is the most popular Cricket tournament after the World Cup.

The above examples suggests that the NER tool was not able to extract all the Cricket related features completely. We can train the NER tool with all the sports related data so that it can identify all the cricket related entities by itself. Due to time constraints, we have manually extracted all the features related to the sports domain which are missed out by the NER tool and created a static collection of all the features in the database.

4.4 Feature Vectors (Manual Feature Extraction):

As discussed in the previous section, we extracted the features manually by creating feature vectors. Each feature vector is a tuple of key, alias and category. Each feature is classified as either Player, Team or Tournament.
Examples of the feature vectors are given below:

4.4.1. Player

```
{
  "_id" : ObjectId("554694951eca218ff15efd74"),
  "key" : "Virat Kohli",
  "alias" : ["Virat", "Kohli","iamvKohli","viratkohli","Chiku"],
  "category" : "Player"
}
```

The above example is a feature which is classified as a player. One observation here is that the alias contains a word "chiku" which is the nickname of the player. There is every chance that a post may contain a hash tag with the nick name of the player. Even though nicknames may not be related to the player's name, but it should be considered as an alias of that player.

4.4.2. Team

```
{
  "_id" : ObjectId("5546978a1eca218ff15efd76"),
  "key" : "Royal Challengers",
  "alias" : ["RCB", "Royal Challengers Bangalore","playbold"],
  "category" : "Team"
}
```

The above example is a feature which is classified as a Team. The observation here is that the alias contains a word “playbold” which is the caption of the team. Many of the posts from the teams and fans will contains hashtags related to the captions of the teams. We have to consider this data in describing the alias of the feature.

4.4.3. Tournament

```
{
  "_id" : ObjectId("554696061eca218ff15efd75"),
  "key" : "IPL",
  "alias" : ["IPL", "IPLT20", "Indian Premier League", "Pepsi IPL","IPL2015"],
  "category" : "Tournament"
}
```

The above example is a feature which is classified as a Tournament. The observation here is that the alias contains a word “Pepsi IPL” who is the main sponsor the tournament. Many posts from the teams and fans will contains hashtags related to the tournament.

4.5 Generating the Topic Sets

As mentioned in the section 3, Facebook and Twitter have different graph structure and we need to define a common structure to align the different data sets. We do this by creating our own structure for storing the Twitter and Facebook data. We picked similar attributes from Facebook and Twitter which are required to perform our sentimental and popularity analysis.

Source **:** The field mentioning the source of the post. It will be either Facebook or Twitter
Id **:** The id of the post.
Data **:** The text of the post.
Timestamp **:** The time of creation of the post.
Tags **:** The calculated Clusters that this post belongs to.
Comments **:** The array of comments of a post.
Likes **:** The number of likes/favorites of a post.
Spread **:** The number of retweets/shares of a post.
PS **:** The calculated popularity score of each post

Creating this structure for every post/ tweet from Facebook and Twitter becomes the central point for our clustered data analysis.

4.6 Clustering the data:

The challenge is to cluster the post/tweet using the feature extraction. Using the features we have generated already, we generate tags corresponding to each post/ tweet. These tags represent the most appropriate proximity of a given post/tweet to an entity or feature.

For example, the below tweet from Twitter has 5 hash tags and one user mention.

"What a match, Congrats @imVkohli for a splendid #Century #Icc"

This post is talking about the player Virat Kohli and we need to cluster post across the feature "Virat Kohli".

```
{key:"Virat Kohli",alias:["Virat","Kohli","vkohli","imvKohli"],category:"Player"}
```

In addition, there will be many post that belong to more than one player or team. Most of the times the tags will not be straight forward. The post mentioned below is updating the score of a particular match having hashtags that represents two teams.

Example post: "#**RCBvsMI**, RCB Score 92/1 9.0 overs. #royalschallengers are scoring at a good rate"

This type of hashtags are quite common and we must cluster this post to both the teams. We have implemented programmatic logic which handles the processing of such hash tags using regular expressions. Each hashtag is checked for the presence of "vs". If we find the keyword we split the hashtag into 2 separate entities and match each of them against the feature vectors. This way we can identify the different features present in each hashtag.

4.6.1. How to generate tags:

Given a post P, we find the term frequency (TF) score of P against each feature vector. Each feature vector is already classified as $F_C \in$ {Team, Player, Tournament}.

We compare the hashtags and user mentions in each post with the alias values stored in each of the features that we classified earlier.

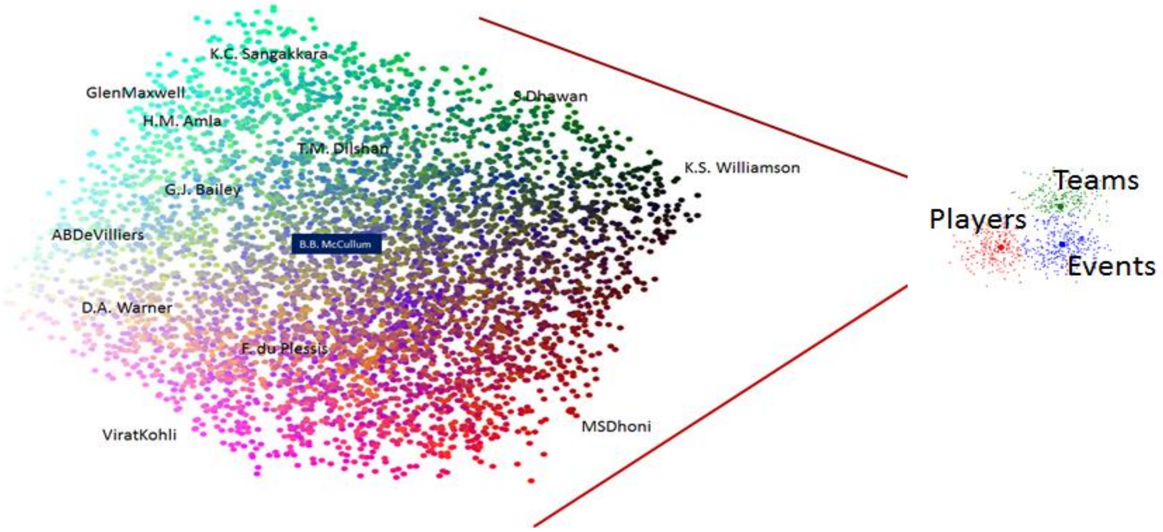
- If the post contains more than one hashtag or user mention of a same feature, then we cluster that post to the single feature about which it is talking.
- If the post contains more than one hashtag or user mention of different features, then we cluster that post into different features about which the post is talking.

For example,

- If the post contains "#RCB, #Royal Challengers and #RC Bangalore", we cluster this post to the single feature "Royal Challengers Bangalore" which is a team.
- If the post contains "#RCB, #MI and #IPL", we relate this post to the 3 features "Royal Challengers Bangalore", "Mumbai Indians" which are teams and to "Indian Premier League" which is a tournament.

4.6.2. Compound Clustering:

After assigning tags to every post/ tweet we can perform our clustering logic, which is simple enough to collect all the posts that have a specific tag. For instance, to cluster posts for feature "Virat Kohli" we collect all the post that contain the tag of "Virat Kohli". Using this approach we can cluster posts around every feature in the feature set and evaluate the score of sentiment and popularity for that feature. To get the desired results we compare the popularity scores of each feature within a category. That is, compare the popularity score of each player and we can get which player was most popular in the time period used, similarly we evaluate the most popular team and event.



The image describes the process of clustering the posts/tweets around each player and then using the category player as whole to generate the desired results.

4.7 Sentiment Analysis

We are using two different sentimental analysis approaches.

4.7.1 Model for Sentiment Analysis using Bag of words:

To Start with, we collected sentiment bearing datasets from various sources. To develop and test our sentiment analysis using bag of words we used online available collection of 4 Lakh tweets. Also we trained our model with a datasets of 2011 positive opinion words and 4783 negative opinion words [13].

To improve the accuracy we used a subtle approach to handle the negation in the sentences which is described below:

If we have a sentence as below:

"This is not a cool night"

When we get a negation token such as not/ no/ none, we prefix every word following the negation token with not_word until the sentence ends. So the original sentence is transformed to:

“This is not_a not_cool not_night”

Now, when the parser encounters words with prefix not_, it just reverse the polarity of that word considering the polarity of the original word.

As we are not performing the sentiment analysis using contextual semantics, we sanitized the sentences further by removing the common English grammar words. Pruning English grammar words could be useful during Naive Bayes approach described in the next section.

We performed a test run on the trained model against the section of 10000 tweets from datasets of 400 thousand tweets and was able to polarize 43% statements (4305 tweets). This result can be improved if we can further train our model against the jargons used in the social media, especially with the jargons used in the sports fan pages. Also training the model with real time opinions will increase the accuracy of the overall sentiment analysis.

4.7.2 Model for Sentiment Analysis using Naive Bayes + Bag of Words (Hybrid Approach):

To improve the accuracy of the sentiment analysis we decided to train our model with real time opinions, that is the opinions which are posted online over social media like Twitter or Facebook by the users. To enhance the accuracy of our system we extracted those opinions from the sports fan pages over the social media. After extracting the opinion sentences we manually classify them as positive, negative and neutral sentence. After the process of extraction and classification we train our model and use most popular Naive Bayes algorithm to get the prediction on sentiment value of a given sentence. Finally we implemented the hybrid approach to generate the sentiment analysis over a given sentence by appending our custom logic over the Naive Bayes algorithm.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

```
If w ∊ {bag of words}
          p(w | C) = 0.2
else
     p( w | C ) = Number of occurrences(w) / Total Terms size of the
given class
p( x | C ) = ∑ p( w | C )

p(C | x) = p(C) p( x | C )
```

where:

- p(C | x) is the probability of class being either positive , neutral or negative given a document(or a sentence)
- p(C) is the prior probability of a given class (positive, neutral, or negative),
- p(x | C)is the likelihood probability of a given document to a class.

After evaluating the p(C | x) for every class, we take the maximum probability score for classifying the given document as positive, neutral or negative.

To accomplish the task of training the model with classified statements, we extracted over 100000 comments from Facebook page ICC Cricket World Cup and extracted all the comments that have been made during the year 2014 and did the same process to get the tweet comments from the page of ICC World Cup over twitter.

Label My Data: http://apps.thecodestreet.com/labelmydata/

For our purpose of classifying the huge dataset, we created a web site and asked the public to classify the post into either of the three (Positive, Negative and Neutral) categories. We merged the datasets from Facebook and Twitter and sanitized it before uploading the combined dataset of phrases over 50K samples. Each sentence has to be classified into positive, negative or neutral. The accuracy of the model depends on the accuracy of the classification of the test data.

4.8 Sentiment Calculation for Individual posts:

After preparing our hybrid model for sentiment analysis, we performed the sentiment evaluation over the crawled data from Facebook and Twitter. This evaluation is done on individual post/ tweet that is being crawled and the results are stored for being used after the clustering.

4.9 Popularity calculation:

Each post is assigned a popularity score using a formula. We will sum up the popularity score of all the posts that belong to a cluster to get the total score of each clustered feature. We store the values in a separate table along with the cluster value and the post with the highest score in each cluster.

We considered many factors before formulating the formula. The factors we assumed are as follows:

- A post or tweet which has a strong message or a serious sentiment value is more likely to get retweeted or shared than liked. It is most of the times liked and shared.
- A post or tweet which is not reflecting any serious message is more likely to be liked than shared.
- A post which has high share/retweet count should be given higher importance when compared with a post with low shares/retweets.
- A post with high likes/favorites should not be given higher value when compared with a share /retweet count.
- A like has comparatively lower value than a share.

The formula we used to calculate the score of each post is as follows:

$W(\text{tweet/post}) = \{ p * R_{count}(\text{tweet/post}) + q * F_{count}(\text{tweet/post}) \} * T$

where
W(tweet/post) = Weight of a tweet in Twitter or the post from Facebook
R_{count} = Retweet count in Twitter or Share count in Facebook
F_{count} = Favorite count in Twitter or Like count in Facebook
p = The weightage factor for retweet/share (We used p=0.7)
q = The weightage factor for favorite/like (We used q=0.3)
T = Time decay function = e^{-t}
where
t = 1 + Hours (current time – tweet/post time) * K (We used K=0.01)

Considering the above mentioned factors, we deliberately ignored one another main factor after careful thought. The post or tweet from a famous sportsperson has high popularity when compared to a post from the ordinary people, because of the fact that the sportsperson has a lot of friends/followers. So, it is generally assumed that the popularity score of the tweet/post is directly dependent on the popularity of the person tweeting/posting it.

We present the following reasons for our assumption.

- Since the popular user has a lot of followers in his list, it is more likely that his post will have high circulation among the social network. So, no need to add additional weightage factor based on the person.
- Any post from a popular personality may not be liked/shared by all.
- If the post really has good content in it, it will be liked and shared accordingly. Same will be the case with a post/tweet from a normal user too.

We used different values for k(like 0.1,0.5,0.05) with each test run and finalized on 0.1 which gives the best results.

5. Code description and Technical Specs:

5.1 Code Description:

SWMController: This is a controller which initiates the application.

SWMProcessor: SWMProcessor has all the logic. The methods in it will fetch the data from Twitter using Twitter API and then calculates the popularity score, clusters them and stores in the database.

```
public void processData()
public List<Cluster> getClusters()
```

SWMDAO: Interface which contains methods to fetch and store Twitter and Facebook data.

```
public void storeTweets(List<Tweet> tweets) throws Exception;
public List<Post> getPosts() throws Exception;
public void storeClusters(List<Cluster> clusters) throws Exception;
```

TwitterDAO: Interface which contains methods to fetch the Twitter data.

```
public ResponseList<Status> get1DayTweets() throws TwitterException;
```

SWMDAOImpl: Implementation of *SWMDAO* interface.

TwitterDAOImpl: Implementation of *TwitterDAO* interface

Cluster: Model class which contains fields related to a Cluster.

```
private String key;
private double totalPs;
private double maxPs;
private String category;
private Map<String, Object> data;
```

Feature: Model class which contains the data of a Feature.

```
private String name;
private String category;
private List<String> alias;
```

Post: Model class which has data of a Post (It may be a Facebook post or Twitter tweet).

```
private long id;
private String source;
private Object data;
private Date createdAt;
private List<String> tags;
private long likes;
private long spread;
private double ps;
```

Tweet: Model class which holds a Tweet data

```
private Status status;
private double score;
```

TweetUtil: Utility class which contains logic to how to cluster, how to get Popularity score etc.

```
public static List<Tweet> convertStatusResponse(List<Status> list, Map<String,
Feature> featureSetMap)
public static Map<String, Cluster> convertPostToCluster(List<Post> posts, Map<String,
Feature> featureSet)
public static List<String> getHashTagsUserMents(Tweet tweet)
```

5.2 Technical Specifications:

The list of technologies used for the implementation is as given below:

Back end technologies:

Java JDK – Version 1.8

External Java Libraries used –

- Twitter4j – version 4.0.3
- Mongo Java Driver – version 2.13
- JSTL – Version 1.2
- Java Servlet – Version 3.0

Front end technologies:

HTML5
Javascript
CSS3
Bootstrap
Jquery

Database used:

MongoDB – Version 3.0.2

Web server used:

Tomcat – Version 7.0

6. Test Results:

Crawlers:

We created Twitter crawler using Java and Facebook crawler using Node.js and FB Graph module which is used to execute FB graph API queries.

We created a Twitter account and followed all the players, teams & groups related to Cricket. We use this account's to crawl the APIs for the posts and tweets. This way we were able to access and extract all the relevant data needed for classifying, clustering and predicting the popularity of the posts.

Similar work has been done for the Facebook data extraction. After extracting the data from the twitter and Facebook APIs separately, we integrate the data from the two sources into a common database.

Database Details:

We used MongoDB as the database to store our datasets because of the following reasons:

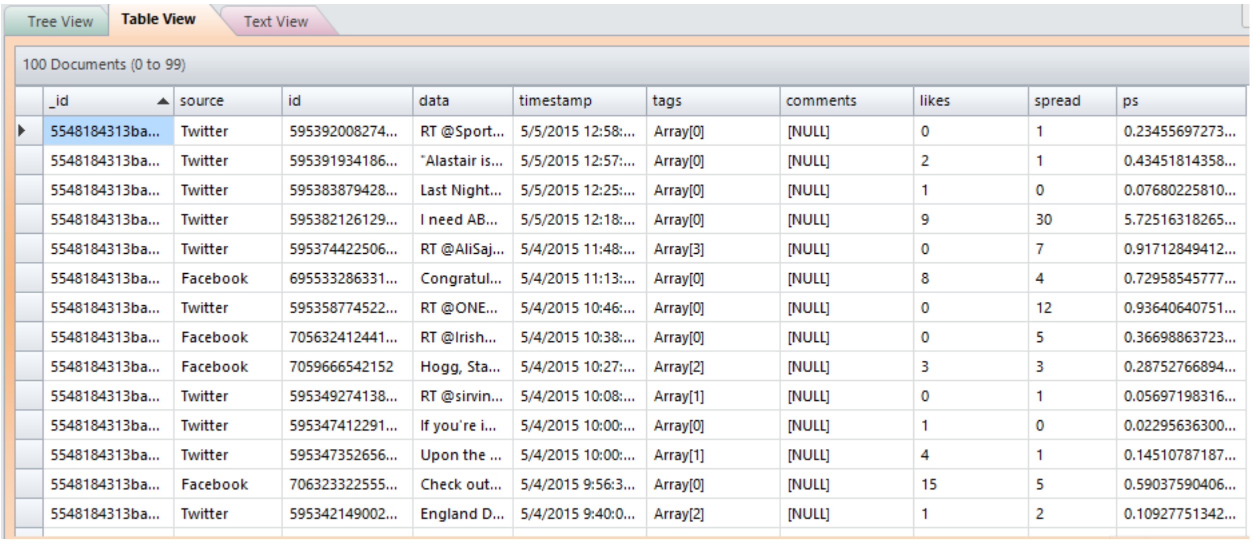
- MongoDB is easy to use.
- We can store unstructured data in the collection. Since all tweets and posts do not contain all the attributes, using a RDBMS database needs usage of NULL values to the missing data fields.
- It's easy to store and access the array values in the collections provided by the MongoDB.
- MongoDB provides faster access to the data.

We have defined a total of 3 collections [DataSchema, Feature, Cluster] in the MongoDB.

1. The Key-Value pairs in the collection **DataSchema** are:

Source **:** The field mentioning the source of the post. It will be either Facebook or Twitter
Id **:** The id of the post.
Data **:** The text of the post.
Timestamp **:** The time of creation of the post.
Tags **:** The calculated Clusters that this post belongs to.
Comments **:** The array of comments of a post
Likes **:** The number of likes/favorites of a post.
Spread **:** The number of retweets/shares of a post.
PS **:** The calculated Popularity score of each post.

Source	Id	Data	Timestamp	Tags	Comments	Likes	Spread	Ps



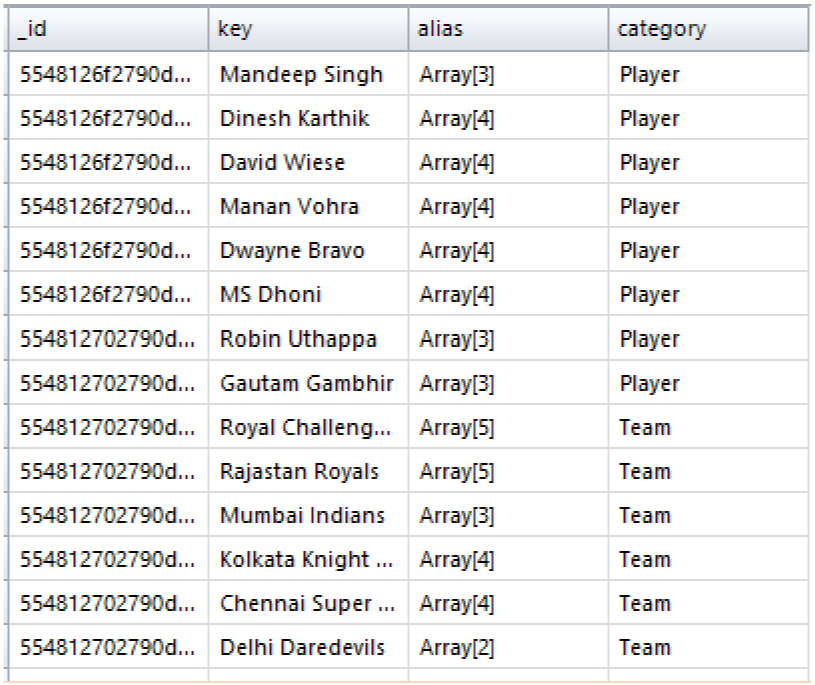
100 Documents (0 to 99)

_id	source	id	data	timestamp	tags	comments	likes	spread	ps
5548184313ba...	Twitter	595392008274...	RT @Sport...	5/5/2015 12:58:...	Array[0]	[NULL]	0	1	0.23455697273...
5548184313ba...	Twitter	595391934186...	"Alastair is...	5/5/2015 12:57:...	Array[0]	[NULL]	2	1	0.43451814358...
5548184313ba...	Twitter	595383879428...	Last Night...	5/5/2015 12:25:...	Array[0]	[NULL]	1	0	0.07680225810...
5548184313ba...	Twitter	595382126129...	I need AB...	5/5/2015 12:18:...	Array[0]	[NULL]	9	30	5.72516318265...
5548184313ba...	Twitter	595374422506...	RT @AliSaj...	5/4/2015 11:48:...	Array[3]	[NULL]	0	7	0.91712849412...
5548184313ba...	Facebook	695533286331...	Congratul...	5/4/2015 11:13:...	Array[0]	[NULL]	8	4	0.72958545777...
5548184313ba...	Twitter	595358774522...	RT @ONE...	5/4/2015 10:46:...	Array[0]	[NULL]	0	12	0.93640640751...
5548184313ba...	Facebook	705632412441...	RT @Irish...	5/4/2015 10:38:...	Array[0]	[NULL]	0	5	0.36698863723...
5548184313ba...	Facebook	7059666542152	Hogg, Sta...	5/4/2015 10:27:...	Array[2]	[NULL]	3	3	0.28752766894...
5548184313ba...	Twitter	595349274138...	RT @sirvin...	5/4/2015 10:08:...	Array[1]	[NULL]	0	1	0.05697198316...
5548184313ba...	Twitter	595347412291...	If you're i...	5/4/2015 10:00:...	Array[0]	[NULL]	1	0	0.02295636300...
5548184313ba...	Twitter	595347352656...	Upon the ...	5/4/2015 10:00:...	Array[1]	[NULL]	4	1	0.14510787187...
5548184313ba...	Facebook	706323322555...	Check out...	5/4/2015 9:56:3...	Array[0]	[NULL]	15	5	0.59037590406...
5548184313ba...	Twitter	595342149002...	England D...	5/4/2015 9:40:0...	Array[2]	[NULL]	1	2	0.10927751342...

2. The Key-Value pairs in the collection **Feature** are:

Key : The field mentioning the unique value of each feature (either a player, a team or a tournament- a sub-clusters of the original clusters)
Alias : The array of alias values which can be used to mention a feature in the post.
Category : The category of each cluster (team, player, tournament).

Key	Alias	Category



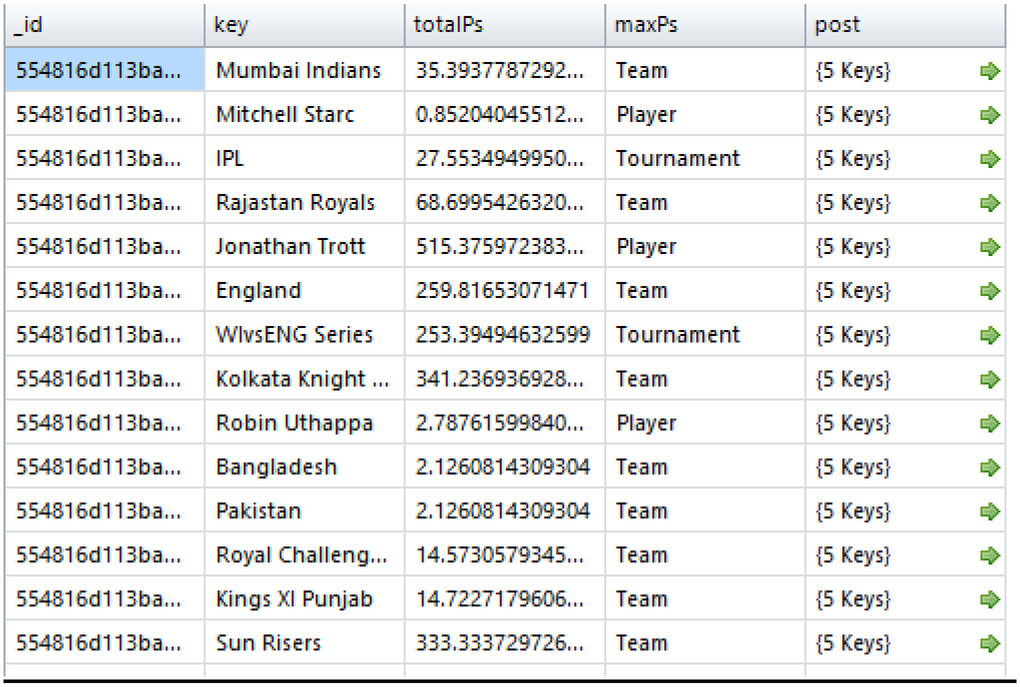
_id	key	alias	category
5548126f2790d...	Mandeep Singh	Array[3]	Player
5548126f2790d...	Dinesh Karthik	Array[4]	Player
5548126f2790d...	David Wiese	Array[4]	Player
5548126f2790d...	Manan Vohra	Array[4]	Player
5548126f2790d...	Dwayne Bravo	Array[4]	Player
5548126f2790d...	MS Dhoni	Array[4]	Player
554812702790d...	Robin Uthappa	Array[3]	Player
554812702790d...	Gautam Gambhir	Array[3]	Player
554812702790d...	Royal Challeng...	Array[5]	Team
554812702790d...	Rajastan Royals	Array[5]	Team
554812702790d...	Mumbai Indians	Array[3]	Team
554812702790d...	Kolkata Knight ...	Array[4]	Team
554812702790d...	Chennai Super ...	Array[4]	Team
554812702790d...	Delhi Daredevils	Array[2]	Team

3. The Key-Value pairs in the collection **Cluster** are:

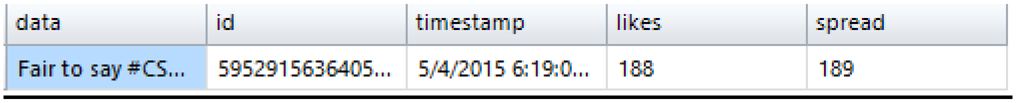
Key : The field mentioning the unique feature
Totalps : The total summed popularity score of the feature
Maxps : The top score of a particular post related to that feature.
post : A 5-tuple key value pair consisting of statistics of the post that earned the top score in each feature.

The 5-tuples are as follows:

Data : The text of the post
Id : The id of the post
Timestamp : The time of creation of the post.
Likes : The number of likes/favorites of a post
Spread : The number of retweets/tweets of a post



_id	key	totalPs	maxPs	post
554816d113ba...	Mumbai Indians	35.3937787292...	Team	{5 Keys} ➡
554816d113ba...	Mitchell Starc	0.85204045512...	Player	{5 Keys} ➡
554816d113ba...	IPL	27.5534949950...	Tournament	{5 Keys} ➡
554816d113ba...	Rajastan Royals	68.6995426320...	Team	{5 Keys} ➡
554816d113ba...	Jonathan Trott	515.375972383...	Player	{5 Keys} ➡
554816d113ba...	England	259.81653071471	Team	{5 Keys} ➡
554816d113ba...	WIvsENG Series	253.39494632599	Tournament	{5 Keys} ➡
554816d113ba...	Kolkata Knight ...	341.236936928...	Team	{5 Keys} ➡
554816d113ba...	Robin Uthappa	2.78761599840...	Player	{5 Keys} ➡
554816d113ba...	Bangladesh	2.1260814309304	Team	{5 Keys} ➡
554816d113ba...	Pakistan	2.1260814309304	Team	{5 Keys} ➡
554816d113ba...	Royal Challeng...	14.5730579345...	Team	{5 Keys} ➡
554816d113ba...	Kings XI Punjab	14.7227179606...	Team	{5 Keys} ➡
554816d113ba...	Sun Risers	333.333729726...	Team	{5 Keys} ➡



data	id	timestamp	likes	spread
Fair to say #CS...	5952915636405...	5/4/2015 6:19:0...	188	189

Sentiment calculations:

Due to insufficient availability of our classified data for training the model for Sentiment Analysis we did not display the partial results in the final output. But we tested our partially trained model on Facebook graph with live feeds using the Facebook API.

The snippet of latest test run:

```
Test Results: 5/4/2015, 10:05AM
69553328633_10152887444638634 2015-05-04T15:44:05+0000 true
69553328633_10152887236018634 2015-05-04T13:00:12+0000 true
69553328633_10152887209008634 2015-05-04T12:31:26+0000 true
```

```
69553328633_10152887161893634 2015-05-04T11:44:30+0000 true
69553328633_10152887076728634 2015-05-04T10:34:16+0000 true
69553328633_10152886941643634 2015-05-04T07:38:55+0000 true
69553328633_10152886836823634 2015-05-04T06:30:00+0000 true
69553328633_10152886772853634 2015-05-04T05:45:00+0000 true
69553328633_10152886753628634 2015-05-04T05:10:00+0000 true
69553328633_10152886715488634 2015-05-04T04:30:00+0000 true
69553328633_10152886709678634 2015-05-04T04:00:01+0000 false
69553328633_10152886705878634 2015-05-04T03:36:16+0000 false
69553328633_10152885974098634 2015-05-03T18:27:01+0000 false
69553328633_10152885931078634 2015-05-03T17:53:52+0000 false
69553328633_10152885530958634 2015-05-03T12:20:52+0000 false
69553328633_10152885398763634 2015-05-03T09:39:49+0000 false
69553328633_10152885296573634 2015-05-03T07:40:56+0000 false
69553328633_10152885208053634 2015-05-03T06:45:00+0000 false
69553328633_10152885199478634 2015-05-03T06:15:01+0000 false
69553328633_10152885171033634 2015-05-03T05:45:00+0000 false
69553328633_10152885163083634 2015-05-03T04:54:45+0000 false
69553328633_10152884487958634 2015-05-02T17:53:41+0000 false
69553328633_10152884417658634 2015-05-02T17:12:57+0000 false
69553328633_10152884272953634 2015-05-02T15:28:37+0000 false
69553328633_10152884178368634 2015-05-02T14:22:11+0000 false
trace ended
Total Post Found: 10
Total comments crawled: 3238
Post Id: 69553328633_10152887236018634
Sentiment score 0
Post Id: 69553328633_10152887444638634
Sentiment score 0
Post Id: 69553328633_10152887209008634
Sentiment score 0.6
Post Id: 69553328633_10152887161893634
Sentiment score 0.6
Post Id: 69553328633_10152886772853634
Sentiment score 1.7999999999999998
Post Id: 69553328633_10152886715488634
Sentiment score 0
Post Id: 69553328633_10152887076728634
Sentiment score -0.6
Post Id: 69553328633_10152886941643634
Sentiment score 0
Post Id: 69553328633_10152886753628634
Sentiment score 0.6
```

Results generated from Label My Data:

```
{'comment' : 'man of da series in da upcoming world cup.', 'value' : 'pos'}
{'comment' : 'amazing', 'value' : 'pos'}
{'comment' : 'Ham Pakistani bolta nai pehla karka dakhata hn!', 'value' : 'neutral'}
{'comment' : 'Ban bouncer in all form of cricket immediately otherwise we should sue
on ICC....', 'value' : 'neg'}
{'comment' : 'still remember this.. <3', 'value' : 'pos'}
{'comment' : 'Alex Hales as England finally get an aggressive opening batsman to make
most of first batting power play', 'value' : 'pos'}
```

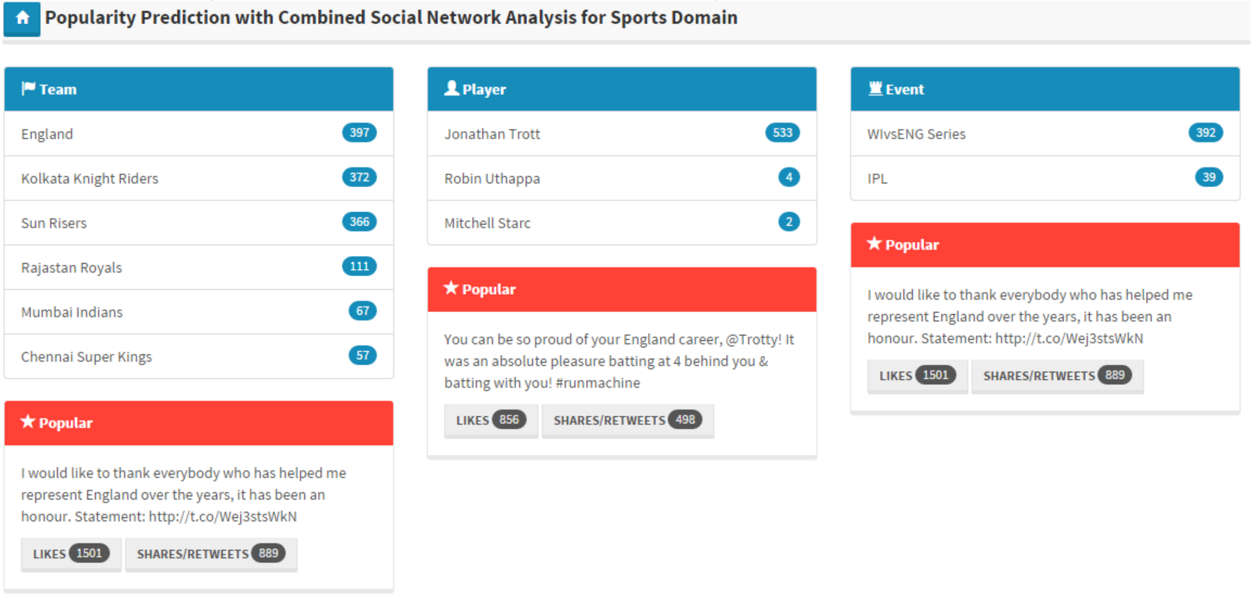
```
{'comment' : 'Kane stuart williamson, because soon 22 is going to replace 10', 'value'
: 'neutral'}
{'comment' : 'Inshallah Pakistan will lift the trophy', 'value' : 'pos'}
```

Prediction calculations:

```
Post [score=1.0474871717376804, status=RT @richardmousley: @bablakecricket So proud
of Alex, Tom & Dan having been selected for @CricketingBears summer squads
#relief http://t.co…, likes=0, spread=8, tags=[]]

Post [score=0.39265494940788653, status=RT @Attockcc: Practice for the big hit
challenge tomorrow at 6pm @CricketingBears @cteccahttp://t.co/T8tmEVi6qQ, likes=0,
spread=3, tags=[]]

Post [score=20.14959777251325, status=.@iamsrk Thanks! And thanks for the hospitality
in Kolkata. Hope to meet your team again this year. :) #OrangeArmy #SRH, likes=99,
spread=112, tags=[Rajastan Royals, Sun Risers]]

Post [score=2.9706233803416935, status=Peter Moores frustrated by England
performances but pleads for patience:http://t.co/L8G1mEmSXy http://t.co/hVO5oSuaxb,
likes=16, spread=16, tags=[England, WIvsENG Series]]

Post [score=6.750431795640988, status=Missing this little poppet.. and you to jubey.
Grace is growing so quickly ! #niece #sister #lovehttps://t.co/ONklS0RILm, likes=89,
spread=14, tags=[]]
```

6.1 Trail run Results

The results using the value with of K=0.05(In Popularity prediction value formula)



Popularity Prediction with Combined Social Network Analysis for Sports Domain

Team	
England	397
Kolkata Knight Riders	372
Sun Risers	366
Rajastan Royals	111
Mumbai Indians	67
Chennai Super Kings	57

★ Popular

I would like to thank everybody who has helped me represent England over the years, it has been an honour. Statement: http://t.co/Wej3stsWkN

LIKES 1501 SHARES/RETWEETS 889

Player	
Jonathan Trott	533
Robin Uthappa	4
Mitchell Starc	2

★ Popular

You can be so proud of your England career, @Trotty! It was an absolute pleasure batting at 4 behind you & batting with you! #runmachine

LIKES 856 SHARES/RETWEETS 498

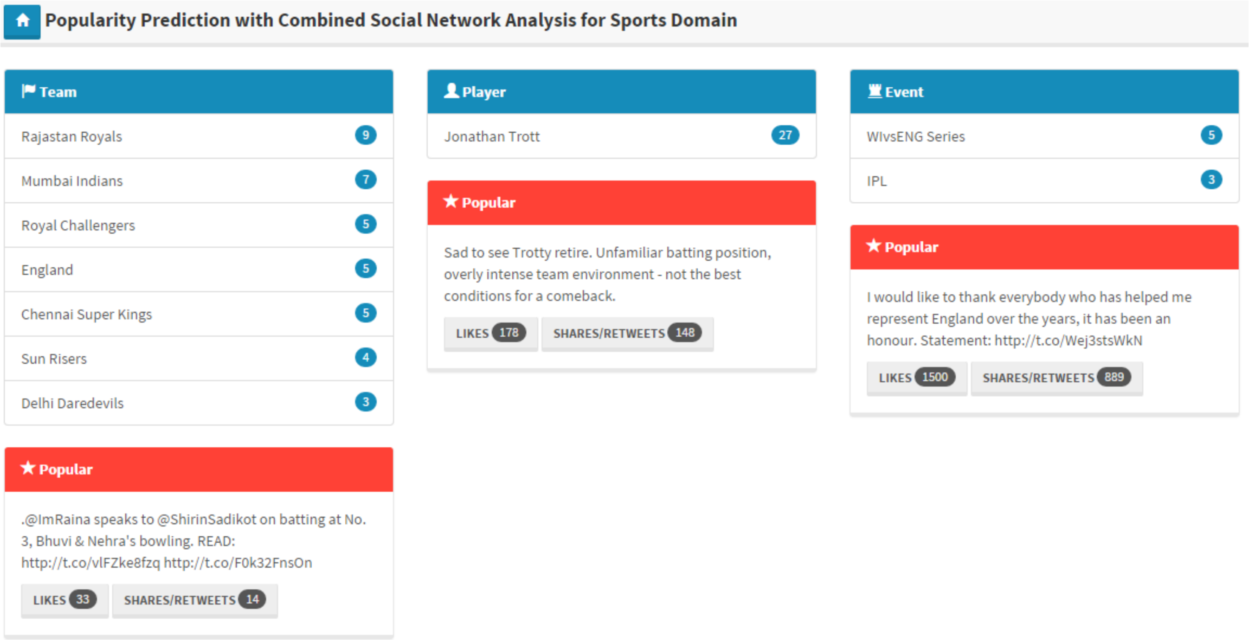
Event	
WIvsENG Series	392
IPL	39

★ Popular

I would like to thank everybody who has helped me represent England over the years, it has been an honour. Statement: http://t.co/Wej3stsWkN

LIKES 1501 SHARES/RETWEETS 889

The results using the value with of K=0.5(In Popularity prediction value formula)



Popularity Prediction with Combined Social Network Analysis for Sports Domain

Team	
Rajastan Royals	9
Mumbai Indians	7
Royal Challengers	5
England	5
Chennai Super Kings	5
Sun Risers	4
Delhi Daredevils	3

★ Popular

.@ImRaina speaks to @ShirinSadikot on batting at No. 3, Bhuvi & Nehra's bowling. READ: http://t.co/vlFZke8fzq http://t.co/F0k32FnsOn

LIKES 33 SHARES/RETWEETS 14

Player	
Jonathan Trott	27

★ Popular

Sad to see Trotty retire. Unfamiliar batting position, overly intense team environment - not the best conditions for a comeback.

LIKES 178 SHARES/RETWEETS 148

Event	
WIvsENG Series	5
IPL	3

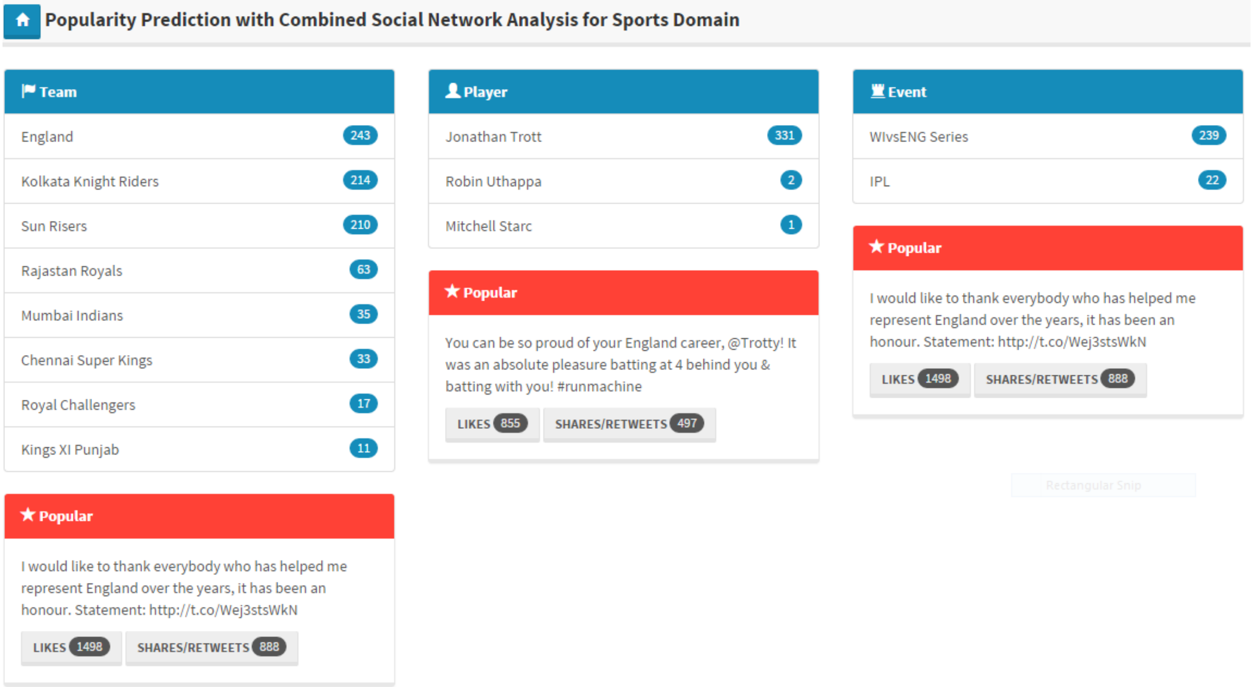
★ Popular

I would like to thank everybody who has helped me represent England over the years, it has been an honour. Statement: http://t.co/Wej3stsWkN

LIKES 1500 SHARES/RETWEETS 889

6.2. Final Results

The results using the value with of K=0.1(In Popularity prediction value formula)



Popularity Prediction with Combined Social Network Analysis for Sports Domain

Team	
England	243
Kolkata Knight Riders	214
Sun Risers	210
Rajastan Royals	63
Mumbai Indians	35
Chennai Super Kings	33
Royal Challengers	17
Kings XI Punjab	11

★ Popular

I would like to thank everybody who has helped me represent England over the years, it has been an honour. Statement: http://t.co/Wej3stsWkN

LIKES 1498 SHARES/RETWEETS 888

Player	
Jonathan Trott	331
Robin Uthappa	2
Mitchell Starc	1

★ Popular

You can be so proud of your England career, @Trotty! It was an absolute pleasure batting at 4 behind you & batting with you! #runmachine

LIKES 855 SHARES/RETWEETS 497

Event	
WIvsENG Series	239
IPL	22

★ Popular

I would like to thank everybody who has helped me represent England over the years, it has been an honour. Statement: http://t.co/Wej3stsWkN

LIKES 1498 SHARES/RETWEETS 888

7. Conclusion and future work

We can have a wide array of extensions to our project implementation.

1. Using the Stanford NER tool, we can train our own models to extract the entities related to the sports domain. With this we can automate the task of the feature extraction with more accuracy.

2. We can extend this application into a web application which can be embedded in other social network sites as an add-on application requiring user permissions to execute.

3. We can extent the same concepts in different domains. Some real life examples we think possible by extending our application to the domain of online shopping goes like this: If we can extend our technique to list out the popular items of a particular user of the shopping application, this information can be used to display the list of favorite items as per the user likings when he logins in next time. This will enhance the user's chance of buying the products thereby bringing profits to the seller.

4. Sentiment Analysis is a topic of wide research and different techniques of sentiment analysis can be used to get more accurate results, for example lexicons used in the bag of words collection can be assigned a different value based on the intensity of their sentiment value. Awesome is more intense as compared to good.

5. Label My Data -We created a website to get an unbiased classification of the datasets. Also the importance of training data for a supervised model is crucial and sometimes acquiring and generating these datasets is difficult. We plan to lend a hand to make this job bit easy with the Label My Data, by asking general public to classify the data and making already classified data sets public for use under GNU license.

8. References:

[1] http://www.iprospect.com/en/ca/blog/10-sentiment-analysis-tools-track-social-marketing-success/
[2] https://cloud.google.com/prediction/docs/sentiment_analysis
[3] https://www.repustate.com/sentiment-analysis/?tour=1
[4] Murphy, David (April 13, 2014). "44 Percent of Twitter Accounts Have Never Tweeted" PC Magazine.
[5] http://en.wikipedia.org/wiki/List_of_most_popular_websites
[6] http://sporteology.com/top-10-popular-sports-world/
[7] https://dev.twitter.com/ rest/public
[8] https://dev.twitter.com/overview/documentation
[9] https://developers.facebook.com/tools/explorer/
[10] https://developers.facebook.com/docs/graph-api
[11] http://arxiv.org/pdf/1111.4503v1.pdf[The Anatomy of the Facebook Social Graph
[12] http://nlp.stanford.edu/software/CRF-NER.shtml
[13] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar