Spleetter: a PACT-based Twitter Statistical Analyzer

Matteo Lissandrini
University of Trento
ml@disi.unitn.eu

Davide Mottin
University of Trento
mottin@disi.unitn.eu

ABSTRACT

Twitter is the biggest micro-blogging system and users therein produce every day a 175 million of short messages , namely tweets. The ability to perform analyses fast on the tweets is not only useful, but also vital to retrieve news and real-time statistics. However, since the number of different statistics to compute on the same data is big (e.g., the hashtag trend over the time, the number of tweets per user), there is the need of performing several operations at the same time. On one hand we can consider a Map-Reduce system in order to split the operations into different machines running several map and reduce steps. On the other hand, a map and reduce paradigm may not suffice. We propose to solve this problem with a PACT flow, that is a semantically reacher map-reduce like-system. Our solution is both simple and highly modular, being composed by simple operations that can be easily parallelised. We show how to compute a set of interesting statistics out of 22 million tweets downloaded in a period of two weeks, showing the polarity of each tweet and filtering the data to produce interesting analyses.

Keywords

Twitter, Stratosphere, Data Analysis, Sentiment Analysis

1. INTRODUCTION

The ability to perform various statistical analyses with Twitter has attracted much interest and research in the last few years [2, 3] and used to influence politics [4] and advertising [1]. With 140 millions of users, Twitter is the leading social network for the real time sharing and re-sharing of information, having become the most used communication media in the spreading of socio-cultural events across the world [?]. Moreover, Twitter has an enormous advantage over other social networks like Facebook in one key area: while people on Facebook tend to be friend their friends, users on Twitter subscribe to other users if they match their own interests[1]. In [?, ?, ?] authors introduce different techniques to identify peak of activities around particular keywords, *hashtags* or users. They describe topic discovery techniques to mine informative summarization of events, and to visualize user-friendly streams of postings. All these previous approaches aim to do real time identification of data peaks, smart labeling or even post-mortem analysis of events.

We would like to mine a real time stream of posts and to identify trends and patterns that allow us to forecast which topics, keywords or events will became popular in the near future. For this reason, we need to gather social media timelines and to collect relevant statistics and insights on how events propagates in a social network. However, the huge amount of information per day cannot be processed by a single machine.

To this end, we propose a schema based on the PACT paradigm implemented in Stratosphere, that is an enhanced map-reduce system able to mix second order functions. Our goal is to conduct an analysis of a big dataset, automatically downloaded using the Twitter streaming APIs. We propose a PACT flow or program[2] that takes in input a set of tuples, having tweets and user ids, a set of users and a vocabulary and produces several statistics used by data analysts in order to perform in-depth data analysis and event discovery. We now describe the basic structure of the Twitter system and a highlight of the proposed solution discussed in Section 2.

1.1 Twitter structure

Twitter is a micro-blogging system designed to allow users to send short messages having a maximum of 140 characters, called *tweets*. In the text, users are allowed to specify *hashtags* that are sequence of characters usually describing an argument and marked by the character '#'. Users can also reference other users with '@user' notation. A particular kind of reference is a *retweet* which is a tweet preceded by the tag “RT” and the user name that first posted the message. A user is *retweeting* the tweet from another user when she thinks that it is an interesting piece of text to be shared her followers. The last information in the tweets are the urls, that are usually shortened using available online services.

1.2 Proposed solution

We aim to find information regarding topic trends, analyzing user post trends, polarity of the tweets with respect to the creation time, and with respect also to hashtags contained in them. The set of operations we propose is the following:

- **Tweet Cleansing**: we take in input the tuples containing the tweets and a dictionary of english words and we filter out hashtags, user mentions and tweets having a number of english words less than a threshold. The cleaned data are then used throughout the rest of the flow.

- **Polarity extraction**: we use a well-known library for sentiment analysis to extract the general polarity

[1] http://dcurt.is/twitters-graph

[2] here flow or program are used interchangeably

of each tweet. The polarity is defined as value in the range $[-5, 5]$, where a positive number means that the user is talking about something in a favorable manner. Conversely, if a text has a negative polarity the text is written in a dissenting manner. We adopt the SentiStrength classifier (as in [**?**]) that was originally built to perform sentiment detection in short informal texts. It combines a lexicon-based approach with more sophisticated linguistic rules.

- **Hashtag analysis**: we produce a various statistics about hashtags. that take into account the time in which the hashtag appeared and disappeared (i.e. is not used anymore), the maximum and minimum number of mentions per hashtag with the timestamp of the peaks,
- **Topic Analysis**: we perform an analysis of positive and negative per-topic trends, aggregating for each hashtag, for each point in time, the average polarity expressed in tweets containing it. We compute also the average Emotional Divergence[**?**] for each tag.

The document is structured as follows. Section 3 describes the dataset we downloaded and used in our experiments. We propose a solution and describe the PACT program in detail in Section 2. We also show the results of the analysis, with increasing size of the dataset in Section 4. Concluding, in Section 5 we describe the problems and the issues we encountered during the development of the solution and we remark our findings.

2. SOLUTION

We describe here the PACT program diagram we implemented to produce the statistics on tweets that we need. Recall that a PACT program is a generalization of the Map-Reduce paradigm, in which sequence of second order functions are issued in parallel or in sequence and combined to execute complex tasks. The programming paradigm defines five second order functions: map, reduce, match, cross and co-group. It allows the user to specify any kind of combination between them, in any order. We propose a flow that uses all the operator except the cross. For ease of explanation we describe the PACT program[3] as composed in two different blocks: data cleaning (Section 2.1) and the computation of the statistics (Section 2.2).

2.1 Data cleaning

In the data cleaning part we take as input a comma separated file having the format $\langle tweet\_id, user\_id, tweet \rangle$. Table 1 shows an excerpt of the tweet tuples.

Note that it is not obvious how to find interesting information from arbitrarily short-texts. Therefore, we propose a first flow that tries to remove useless or uninformative tweets by filtering-out the ones that do not contain a minimum number of english words. Figure 6 depicts the sequence of operations we designed to clean the data in the preliminary phase.

Once we loaded the tweets into tuples we clean them, from hashtags, user-mentions and URLs. Tweets are then used in two separate flows: (1) we split them into words in order to count the english words and (2) we perform a sentiment analysis over the text, in order to evaluate the positive and negative polarity expressed by them, as described in Section 1. Note that SentiStrength library is well suited to analyze informal text, so we apply this evaluation on the whole text, before stemming and further cleaning.

In order to restrict the search space to those tweets we consider relevant, we import a dictionary of english words and we count english words in each tweet. If a tweet has a percentage of english words greater than a threshold σ we keep it, otherwise we drop it.

From the pruned tweets we extract the users, and we assign to the cleaned text the polarities found in the previous step.

2.2 Compute statistics

After having cleaned the raw data and after evaluating text polarities, we compute the statistics using the tuples we have kept in the aforementioned steps. First, we load and match the tuples with the tweet timestamp we get from the database. For each timestamp, we keep the date and time up to the hour, this means that any further analysis is condensed in a time window of 1 hour. Second, we match the hashtags with the polarities in order to understand positive and negative trends of the topics. As Table 1 shows in a tweet we can find more than one hashtag, but we wish to analyze each hashtag separately. This step is performed by a match operation with the sentiment polarities followed by a sequence of reduce operations to compute different statistics.

The first information we want to extract is about the evolution of the popularity of an hashtag. To obtain this information we count hour by hour how many tweets contain each hashtag, and similarly we also keep track of how many different users mention the same hashtag. Similarly, we collect the hourly popularity of each hashtag. We also record for each hashtag the moment in which it reaches his peak in popularity alongside the date of first and last appearance, marking in this way the lifespan of the hashtag. Moreover, we use sentiment analysis results to compute three aggregates, namely the average positive sentiment, the average negative sentiment and the emotional divergence present in each of them.

These statistics can give us some insights on opinions about the topic entailed by the hashtag. In particular, in [**?**] is demonstrated how the overall sentiment (polarity) in one tweet does not influence its probability to be re-shared. For this reason, they propose emotional divergence as a novel measure to identify which tweets have a better chance of being retweeted. Aggregating this information for each hashtag we can gather additional informations in support of one hashtag being relevant and interesting.

3. DATA

We downloaded 33 million of English tweets from the tweet-stream using the streaming APIs[4] in a period of approximately two weeks from December 18, 2012 to January 18, 2013 [5]. A summary of the main characteristic of the dataset is represented in table 2. The number of users is 16 million, with an average of 2 tweets per user. The number of hashtags is one order of magnitude less than the number

[3] `https://github.com/kuzeko/Tweets-Analyser/`

[4] `https://dev.twitter.com/docs/streaming-apis`

[5] We didn't downloaded any data during the week between the 25-12-2012 and the 02-01-2013

tweet id	user id	tweet
292375792485298176	858488612	#KidCudi - #EraseMe - The Whizz Bells : http://t.co/Lp1zABOV via @youtube
292375792481099777	486970282	RT @Kirra__: Today is a day where I need to crawl into my bed and sleep the day away.
292375792481099776	336390437	There are poor people, money is the only thing they got.
292375792476909568	74570186	I can't do anything without listening to music while I do it.

Table 1: A sample of the tweet dataset

of tweets meaning that users often talk about same topics or do not mark their tweets with hashtags. From this dataset we generated three datasets 100k-Tweets, 1M-Tweets, 10M-Tweets with 100 thousands,1 million and 10 million tweets respectively, in order to test performance with various data size. We also downloaded a dictionary of 213k English words to filter the meaningful tweets from those irrelevant or containing only symbols. To perform the sentiment analysis (i.e., to assign a positive or negative polarity to each tweet) we used the SentiStrength library registering both positive and negative polarity carried in each tweet.

Period	2012-12-18/2013-01-18
Tweets	33774428
Users	16099129
Hashtags	1194691
Max tweets per user	2380

Table 2: Characteristics of the dataset used in the experiments

4. EXPERIMENTAL EVALUATION

We tested our solution with Stratosphere 0.2.1[6] running on a GNU/Linux machine with 2Gb RAM DDR2, and a CPU AMD Athlon$^{\text{TM}}$64bit X2 Dual Core Processor 5000+.We used a version of Stratosphere which is not publicly available at the time of writing as still in beta-testing.

All the experiments have been performed on samples of the original database, in order to show time and quality performance. We used Java JRE 1.6.0_26 to program the PACT using the Stratosphere libraries and server, additionally we integrated the SentiStrength java library[7]. We performed preliminary experiments to set the english word threshold and we finally set it to $\sigma = 0.1$, i.e., requiring at least 1 word out of ten being a valid English word. Since we are matching with english words that are not stemmed a larger threshold would have pruned interesting tweets.

4.1 Time performance

In this section we briefly report and comment time performance registered in the various experiments. We performed a total of 4 different experiments testing the flow on different dataset sizes.

In particular we run over the last 100 thousands, 1 million, 10 million and 20 million tweets, in reverse chronological order. Time results are present in table 3.

As expected, we registered time performance rapidly increasing with respect to the number of tweets. Although the dataset is big and the computed statistics are complex we measured a total time of at most 1 hour for 20 millions tweets in the worst case.

Number of Tweets	Time (secs)
100 thousands tweets	50
1 million tweets	183
10 million tweets	1516
20 million tweets	4006

Table 3: Time performances for the computation of the statistics with respect to the size of the dataset analyzed

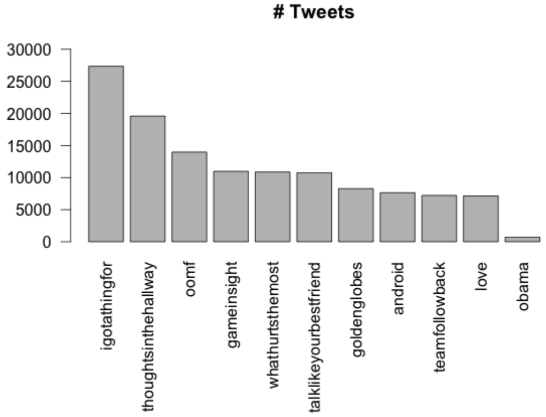

Figure 4: Top10 hashtags for popularities within tweets, plus the hashtag for "obama" vs the number of tweets in which they appear

4.2 Output charts

Starting from our initial assumption, i.e., hashtags are indicators of events or discussion topics, we experimentally show how our PACT flow can find important patterns in our data.

Hashtag	Number of Tweets
obama	1771
obamacare	173
nobama	120
impeachobama	73
obama2012	63

Table 4: Hashtags referring to "Barack Obama" and the number of tweets containing them

In Figure 4 we show the Top10 hashtags for popularity, plus the hashtag for "obama" as a possibly interesting case and we analyze their lifespan in Figure 5. Indeed, identifying that an hashtag has a limited lifespan within the flow of tweets may provide anecdotal evidence of an event happening in that particular time window. Looking instead to the

[6] https://stratosphere.eu/downloads

[7] http://sentistrength.wlv.ac.uk

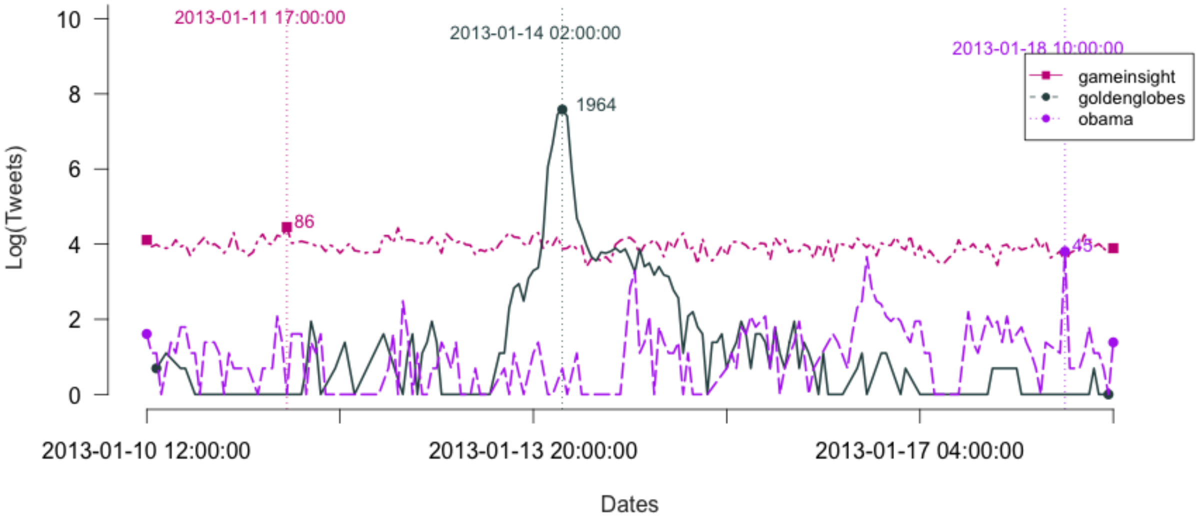

Figure 1: Popularity over time of three hashtags: gameinsight, goldenglobes and obama. Number of tweets have been normalized through the *log* function

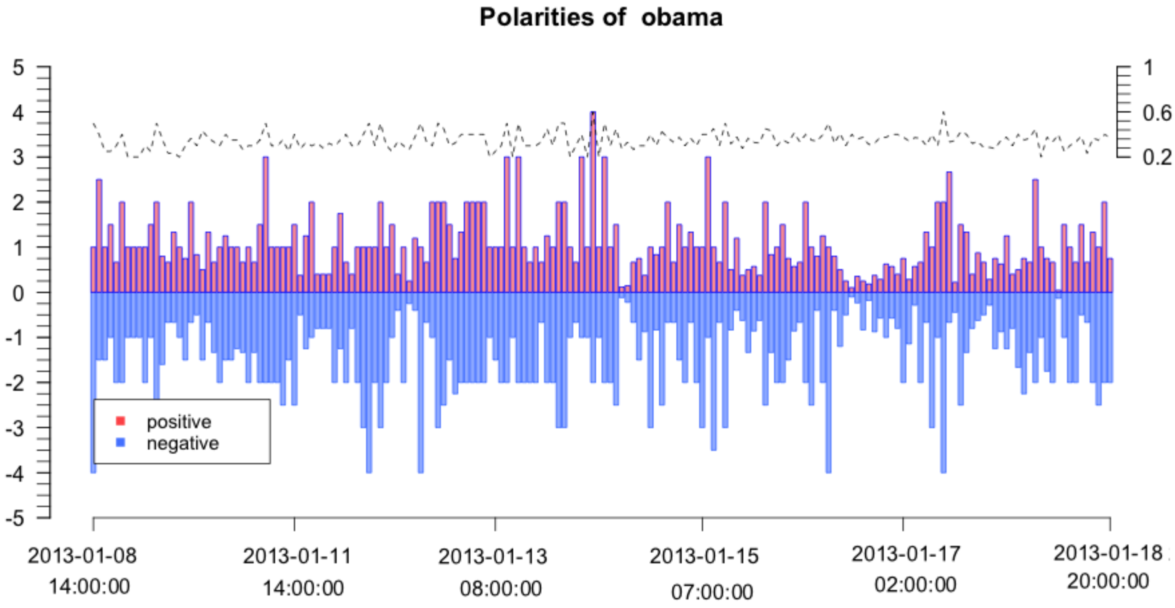

Figure 2: Sentiment polarity expressed through tweets containing the hashtag "obama". The dashed line on the top, represent in each point in time the average emotional divergence.

hourly evolution of the popularity of each tweet we can identify interesting trends, points of peeks and lows. In Figure 1 we can see how the popularity of some hashtags (expressed as hourly number of tweets), can have very different evolutions. In that figure the hashtag "gameinsight" refers to the name of one developer and publisher of mobile games and social games, they and their followers use this hashtag when referring to some of these games. We can see that during the time of the analysis the number of tweets mentioning this tag each our is pretty stable. Instead, the other two hashtags have some highly unstable behavior. The one referring to Barack Obama is not so much popular, even considering all the possible hashtags talking about president Obama, as we shown in Figure 4. In contrast to the previous hashtag, the hourly number of tweets containing the tag "obama" is more unstable, and we can see that there are time windows

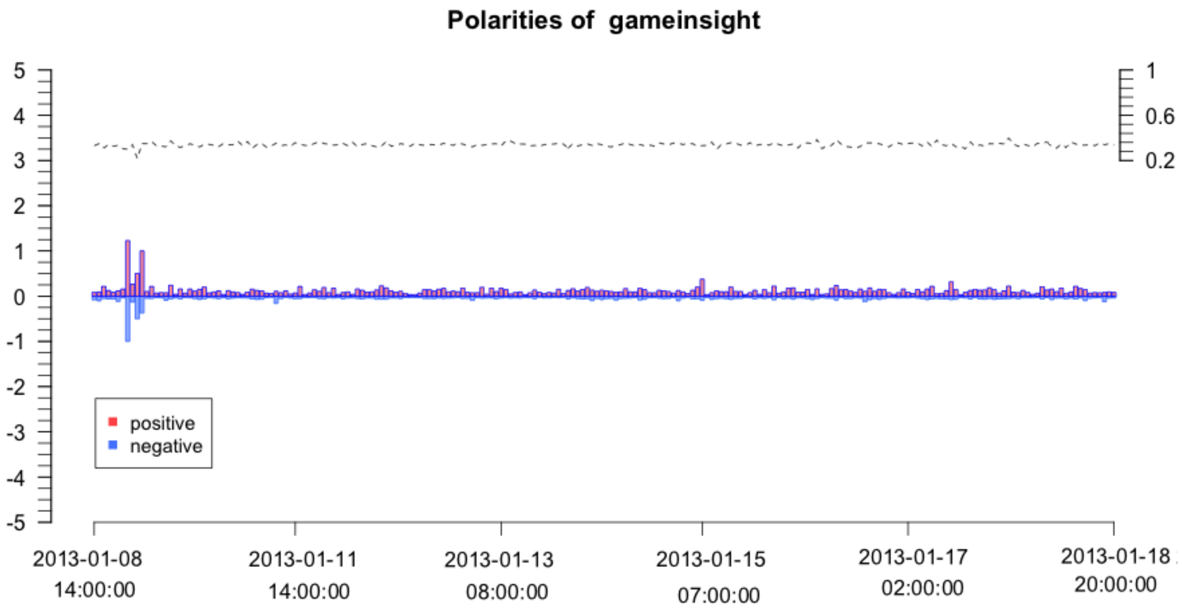

Figure 3: Sentiment polarity expressed through tweets containing the hashtag "gameinsight". The dashed line on the top, represent in each point in time the average emotional divergence.

where this number is actually zero. Finally, in Figure 1 , we analyze also tweets regarding the "goldenglobe" hashtag, referring to the annual Golden Globe Awards, held in 2013 on the night of 13Th. of January. The data-peak we can observe is an indicator of when the award night where held exhibiting a burst in the trend.

Finally, another information to take into account, about the evolution of an event, is the polarity expressed in tweets about it. In Figure 2 and 3 we show average emotional polarities present in tweets in the hourly time-window. These pictures demonstrates how while tweets containing the "gameinsight" keyword are more popular, tweets tagged with the hashtag "obama" have a more strong weight in terms of sentiment polarities. These statistics combined with the evolution of popularity of hashtags are starting points for more complex models and analysis. They can be used to actually identify which hashtags in which time windows are likely to be indicator of events happening. Furthermore, approaches like the ones presented in [**?**] can be applied not only to identify burst in popularity or in sentiment polarities, but also to identify overlapping bursts, i.e. bursts in popularity for different hashtags in the same time window, so that we can actually detected which hashtags may refer to the same event.

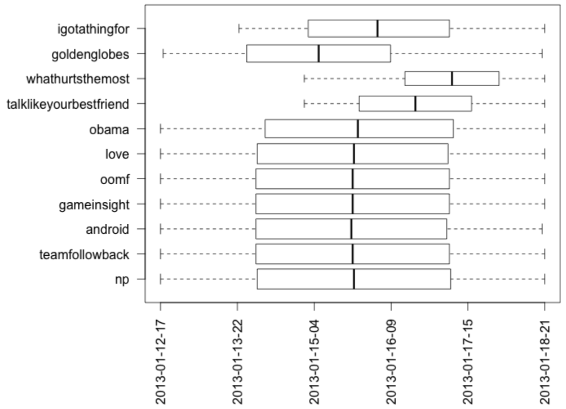

Figure 5: Lifespan of the Top10 hashtags plus "obama" the thicker line delimits where the first and third quartile of tweets for that hashtag lies

In the end of our process we are able to aggregate, for the biggest dataset at our disposal, statistics for more than 480 thousands hashtags, keeping tweets from more than 604 thousands users over 247 time windows. These aggregated statistics can easily be used as grounds for more complex and structured reasoning, but they are already able to highlight popular hashtags.

5. ISSUES AND CONCLUSIONS

During the design and the main development of the project we did not encounter big difficulties or problems with the software provided. We found the PACT model intuitive and flexible for our needs, covering the design and the first implementation without any big issues. The software demonstrated fair performance even on the relative small machine at our disposal, coping with 20 millions tweets and being able to filter and join, for the more loaded component of the flow, more than 90 million tuples.

Problems were encountered during the final testing of the application. First, we reported the following exception:

```
java.lang.RuntimeException:
Data distribution must not be null when
```

```
the ship strategy is range partitioning
```

which was stopping the Nephele runtime, and its web panel, from executing the PACT program submitted.

Second, we noticed some non-coherent results being returned as output of subsequent computations. In this second case, we found that the same PACT program, executed over the same machine with the same input, were actually computing different results in 1 out of 5 subsequent computations on average. Debug of this problem required time due to the current absence of a complete logging environment for the system, in its current version. After extensive analysis we traced back the origin of the problem to a mismatch in attributes used as fields for an equi-join in the early steps of the flow, still we are not able to explain the apparent non-deterministic behavior of the system.

6. REFERENCES

[1] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[2] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 769–778, New York, NY, USA, 2012. ACM.

[3] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 251–260, New York, NY, USA, 2012. ACM.

[4] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the fourth international aaai conference on weblogs and social media*, pages 178–185, 2010.

APPENDIX

A. PACT FLOW

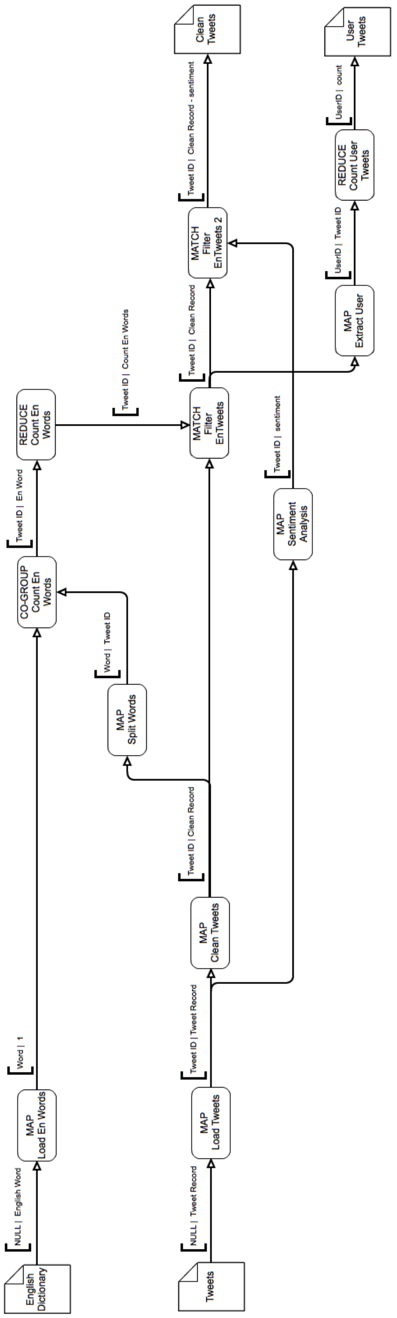

Figure 6: Data cleaning PACT

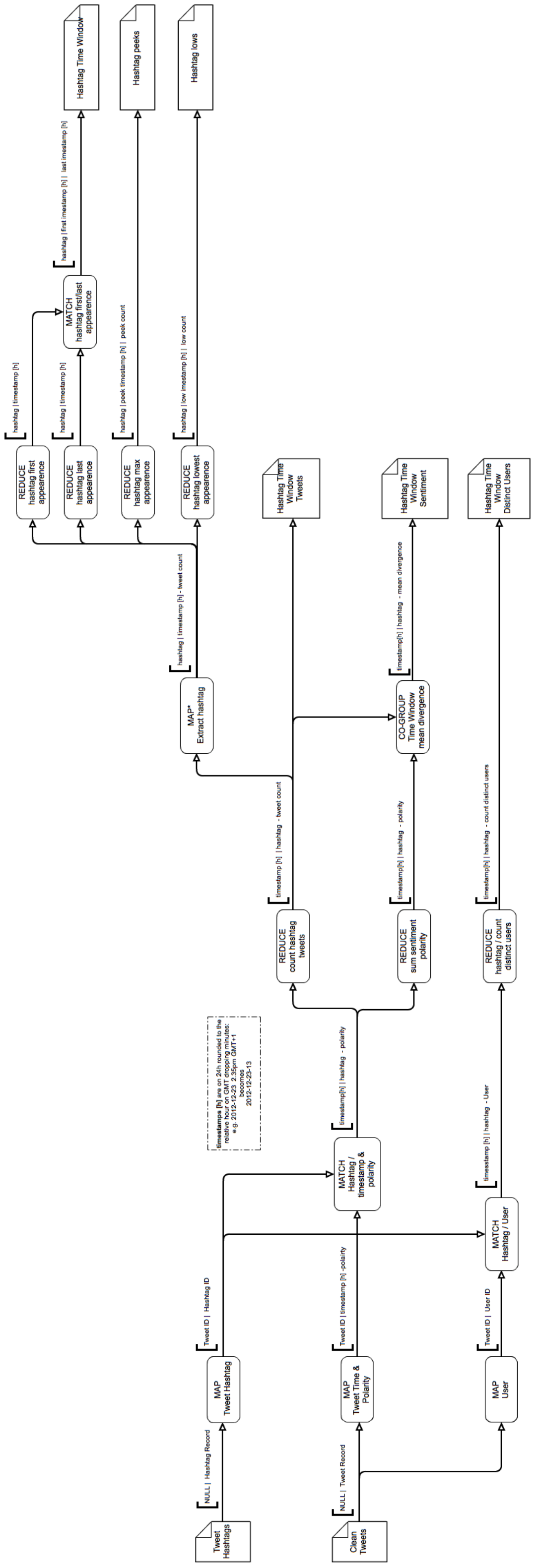
Tweet Hashtags
NULL | Hashtag Record
MAP Tweet Hashtag
Tweet ID | Hashtag ID
Clean Tweets
NULL | Tweet Record
MAP Tweet Time & Polarity
Tweet ID | timestamp [h] -polairty
MAP User
Tweet ID | User ID
MATCH Hashtag / timestamp & polarity
timestamp[h] | hashtag - polarity
MATCH Hashtag / User
timesstamp [h] | hashtag - User
timestamps [h] are on 24h rounded to the relative hour on GMT dropping minutes: e.g. 2012-12-23 2.35pm GMT+1 becomes 2012-12-23-13
REDUCE count hashtag tweets
timestamp [h] | hashtag - tweet count
REDUCE sum sentiment polarity
timestamp[h] | hashtag - polarity
REDUCE hashtag / count distinct users
timestamp[h] | hashtag - count distinct users
MAP* Extract hashtag
hashtag | timestamp [h] - tweet count
CO-GROUP Time Window mean divergence
timestamp[h] | hashtag - mean divergence
Hashtag Time Window Tweets
Hashtag Time Window Sentiment
Hashtag Time Window Distinct Users
REDUCE hashtag first appearence
hashtag | timestamp [h]
REDUCE hashtag last appearence
hashtag | timestamp [h]
REDUCE hashtag max appearence
hashtag | peek timestamp [h] | peek count
REDUCE hashtag lowest appearence
hashtag | low imestamp [h] | low count
MATCH hashtag first/last appearence
hashtag | first imestamp [h] | last imestamp [h]
Hashtag Time Window
Hashtag peeks
Hashtag lows

Figure 7: Compute statistics PACT