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[1], 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 [8] and advertising [1]. With a 140,000,000 subscribers to its service, Twitter is far and away the leading social network for the real time sharing and re-sharing of information, becoming the most used communication media in the spreading of socio-cultural events across the world [2][4]. Moreover, Twitter has an enormous advantage over other social networks like Facebook in one key area: while people on Facebook tend to friend their friends, people on Twitter tend to follow their interests[3]. Previous works as [4], [6] and [5] present different techniques to identify peak of activities around particular keywords, *hashtags* or users. They highlight possibilities for topic discovery techniques for informative summarisation of events, and for user-friendly visualisations of specific streams of postings. All these previous approaches aim to do real time identification of data peaks, smart labelling 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 can allow us to forecast which topics, keywords or events will became popular in the near future. To do this we need to actually mine social media timelines to collect relevant statistics and insights on how events resonates in a social network, but 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[4] that takes in input a set of tuples, having tweets and user ids, a set of users and a vocabulary and produces several statistics. We now describe the basic structure of the Twitter system and an 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 worth spreading such post to a broader audience. The last information in the tweets are the urls, that are usually shortened using some available URL shortener.

1.2 Proposed solution

We aim to find information regarding topic trends, analysing user post trends, polarity of the tweets with respect to the

[1] `http://www.mediabistro.com/alltwitter/twitter-stats_b32050`
[2] `http://www.umpf.co.uk/blog/?p=6830`
[3] http://dcurt.is/twitters-graph
[4] here flow or program are used interchangeably

timing of those tweets, and with respect also to hashtags contained in them. The set of operations we propose to implement using the PACT programming 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 use throughout the rest of the flow.
- **Polarity extraction**: we use a well-known library for sentiment analysis to extract the general polarity of each tweet. The polarity is defined as value between $[-5, 5]$, where a positive number means that the user is talking about something in a favorable manner. Conversely, if a text has a polarity close to -5 the words in the text are dissenting. From [7] we will adopt the SentiStrength classifier, which was built especially to cope with sentiment detection in short informal text. It combines a lexicon-based approach with more sophisticated linguistic rules.
- **Hashtag analysis**: we produce a deep analysis of the hashtags that takes into account the time in which the hashtag appeared and disappeared (not used anymore), the maximum and minimum number of mentions per hashtag with the timestamp,
- **Topic Analysis**: we perform an analysis on positive and negative trends per topic, identifying the topics with the hashtags.

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 want. To recap briefly, a PACT program is a generalisation of the Map-Reduce paradigm, in which sequence of second order functions are issued in parallel and combined to execute complex tasks. The programming paradigm defines 5 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 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 ⟨*tweet_id*,*user_id*,*tweet*⟩. Table 1 shows an excerpt of the tweet tuples. Note that it is not always easy 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 portion of english words. Figure 1 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 the SentiStrength library is well suited to analyse informal text, so we will 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. It is important to notice that in a tweet more than one hashtag can be present, and we want to analyse 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 will count hour by hour how many tweets contain each hashtag, and similarly we also keep track of how many different users tweet about it. In the same way we are keeping track, hour by hour, of how popularity changes for each hashtag. Then we collect for each hashtag the moment in which it reached his peek in popularity alongside the date of first and last appearance, marking in this way the lifespan of the hashtag.

3. DATA

We downloaded 33 milion of US tweets from the tweet-stream using the streaming APIs[5] on a period of two weeks from December 18, 2012 to January 18, 2013. A summary of the main characteristic of the dataset is represented in table 2. The number of users is 16 milion, with an average of 2 tweets per user. The number of hashtags is one order of magnitude less than the number of tweets meaning that users often talk about same topics. From this dataset we generated three datasets 100K-TWEETS, 1M-TWEETS, 10M-TWEETS with 100k,1 milion and 10 milion tweets respectively, in order to test performance with various sizes.

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

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

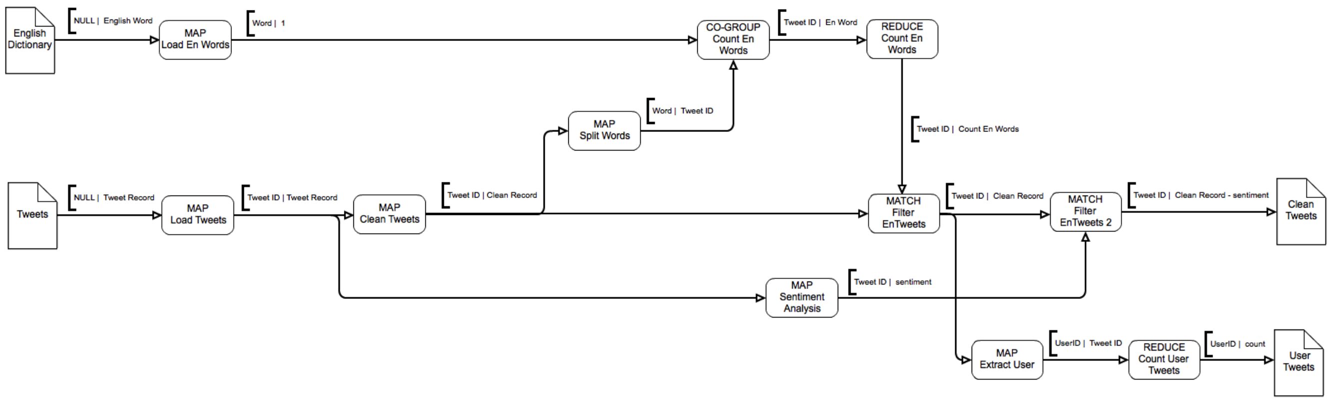

Figure 1: Data cleaning PACT

We also downloaded a dictionary of 213k English words to filter the meanngful tweets from those irrelevant or containing only symbols. To perform the sentiment analysis and (i.e., to assign a positive or negative polarity to the tweets) we used the senti-strength[6] library.

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[7] in Linux machine with 2Gb RAM DDR2, and a CPU AMD Athlon(tm) 64bit X2 Dual Core Processor 5000+ running on Linux kernel 3.0.0-12. 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. We performed preliminary experiments to set the english word threshold and we finally set it to 0.1 in order not to prune to much. Since we are matching with english words that are not stemmed a larger threshold removes interesting tweets.

4.1 Time performance

4.2 Output charts

5. ISSUES AND CONCLUSIONS

6. REFERENCES

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

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

[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] K. Lerman and R. Ghosh. Information contagion: an empirical study of the spread of news on digg and twitter social networks. *CoRR*, abs/1003.2664, 2010.

[5] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *CHI*, pages 227–236, 2011.

[6] M. Mathioudakis, N. Koudas, and P. Marbach. Early online identification of attention gathering items in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM ’10, pages 301–310, New York, NY, USA, 2010. ACM.

[7] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, Dec. 2010.

[8] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In

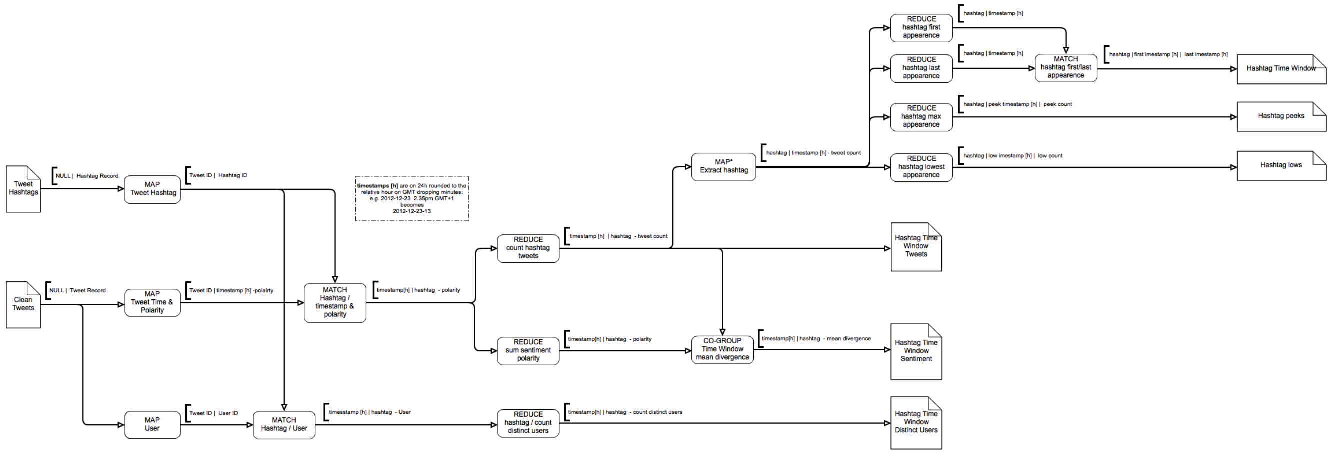

Figure 2: Compute statistics PACT

Proceedings of the fourth international aaai conference on weblogs and social media, pages 178–185, 2010.