Manuscript Preparation Template: ***IBM Journal of Research and Development***

Title: A Scalable Architecture for Real-Time Analysis of Microblogging Data

Authors: P. Cavalin[1], M. Gatti[1], T. Moraes[2], F. Oliveira[2], C. Pinhanez[1], A. Rademaker[1] and R. de Paula[1]

[1] IBM Research
[2] IBM Software Group

Abstract: As events take place in the real world, e.g. sports games and marketing campaigns, people react and interact on Online Social Networks (OSNs), especially microblog services like Twitter, generating a large stream of data. Analyzing this data presents an opportunity for researchers and companies to better understand human behavior on the network, and supposedly in their real lives, during the event's lifespan. Designing automated systems to conduct these analyses in fractions of minutes (or even seconds) is subjected to many challenges: the volume of data is large, the number of posts in future events cannot be predicted, and the system need to be always available and running smoothly to avoid information loss and delays on delivering the analytics results. In this paper, we present a scalable architecture for real-time analysis of microblogging data with the ability to deal with large volumes of posts, by considering modular parallel workflows. This architecture, which has been implemented on the IBM InfoSphere Streams platform, was tested on a real-world use case to conduct sentiment analysis of Twitter posts during the games of the 2013 Fédération Internationale de Football Association (FIFA) Confederations Cup, and has successfully coped with the challenges of this task.

1 Introduction

Messages on microblogs generally consist of short texts that people post to provide either status updates on their activities or observations and/or interesting content about some subject. Such messages can be directed or not to other people [1]. Presently, the most famous online social network for microblogs is Twitter, which reports more than 255 million monthly active users posting about 500 million tweets every day on average (about 347,220 tweets per minute) [2].

The increasing number of users and messages on microblog networks has been drawing the attention of the research community. Although in many cases the contents of these messages (status updates for example) might be useful only for friends and relatives of the user, in many other cases they can also be related to real-world events, for instance sports games, new products' releases, elections, and so forth. Given this relation between the events and what people post on microblogs, processing the contents of event-related messages can be a valuable way for companies to better understand their existing or potential customers/clients.

In this context, opinion mining can provide near-to-real-time event-related costumer insights, given that social media users can post messages to express their opinions about a given subject. Suppose the subject is related to a real-world event, by making of use of a sentiment analysis algorithm, which can classify the text of messages into desired sentiment classes (positive or

negative sentiments for example), a better understanding of what people think about an event can be quickly captured. This information can then be used as insight for decision-making purposes.

For certain types of events, for instance TV shows and political debates, the time required to get the insights need to be very short so that decisions can be made in a way to significantly change how an event has been unfolding. However, for the decisions to be effective they must be based on accurate analytics reports. A good balance between speed and accuracy is something that can make the use of real-time analytics systems successful in real-world applications. Specifically for analyzing streams of posts from microblogs, a big challenge is to accommodate complex algorithms (advanced Machine Learning (ML) and Natural Language Processing (NLP) techniques) in an architecture that scales up to large amounts of data and provides high-confidence analytics reports very quickly. Moreover, system availability is also a matter of concern and can affect both speed and accuracy. Whenever the system is down, either valuable information can be lost or delays in delivering the results can occur. Furthermore, the exact volume of data is unpredictable. There is no way to tell exactly how many users will send messages during the events. In addition, the posting of a high number of messages can be triggered whenever something usual[1] happens during the event. Given these challenges for this type of application, it is very desirable a system that does not slow down (in terms of providing the desired outputs) if to many users post messages or if a high number of messages are posted for some reason.

In the light of this, the main goal of this paper is to describe the development and validation of a system that is able to provide accurate analytics results from event-related microblog data, and is able to scale up to large amounts of data and does not suffer performance and system availability issues. Various systems have been published in the literature for this type of application [3, 4, 5, 6, 7]. Nonetheless, the aforementioned challenges are not addressed by those systems either because the data they process is only a sampling of the total of messages posted, or coping with these challenges is out of their scope. In this paper, we present a general architecture for a system to address these challenges, making use of modular and parallel workflows, allowing the application to scale up to large streams of messages. This scaling up processing, and also the system availability requirement, is aided by implementing the system on the IBM InfoSphere Streams (or simply Streams) platform [8], which provides the appropriate middleware and development language for this task. To validate the architecture and the implemented system, we present results from a real-world use case that consisted of conducting ML-based sentiment analysis of Twitter posts related to soccer games of the 2013 FIFA Confederations Cup. We demonstrate that the analytics results of the system are closely related to the events of the games, demonstrating that the system is suitable to be used as supporting tool for further decision making. In addition, we demonstrate that the system is able to successfully handle throughput peaks of about 120 thousand messages per a five-minute time window, during an event associated to a large volume of messages.

2 Background and Related Work

In this section, we present an overview of current challenges and research on the processing of microblogging data and sentiment analysis on texts. In addition, we discuss current work on real

[1] Something unexpected, very exciting, or polemical, for instance

time support for this task, and describe Streams, the baseline middleware for stream processing that greatly helped the implementation of our proposed architecture (described later).

2.1 Processing Microblog Data

Microblog social networks have been proposed with one main purpose: to provide a fast way for people to write messages (or tweets) and express their opinions. For this reason, the size of messages is limited to a certain number of characters (140 characters in the case of Twitter). On one hand, this is one advantage of the platform for people to write concise messages quickly. On the other hand, the limited length for the messages surely poses new challenges to text mining.

With less space available for writing, people tend to find alternative and faster ways to write the same word, for instance the English words ‘straight’ and ‘str8’, or ‘together’, ‘2gthr’ and ‘2getter’ [9]. Furthermore, microblogs are prone to typing errors, owing to either messages written in a hurry or the typing interface used, such as mobile devices [10]. Additionally, given that people have less space to explain what they write, implicit knowledge is necessary for a proper understanding of a given message. Dealing with these challenges is fundamental to achieve high accuracy on this type of data.

More recently, increasing attention is being paid to sentiment analysis on large social networks such as Twitter [11, 12]. Sentiment analysis consists in evaluating whether a message express some sentiment about a specific topic. The sentiment might be related to specific classes, such as positive, negative, and neutral, or to different degrees of likeness, like the popular five-star approach of review websites [13, 14, 15, 16]. Traditional approaches to sentiment analysis employ ML techniques to classify texts represented with features such as bags of n-grams and Part-of-Speech tags. Many of the proposed approaches have been applied to problems such as review-related websites and business intelligence [17, 18], where the language tends to be more formal and less prone to mistakes since people take enough time to write these texts. However, on microblogs, additional challenges are imposed by the noise, volume and velocity of data. These characteristics make sentiment analysis on OSNs such as Twitter very challenging, since there is the need to cope not only with several difficulty factors in terms of NLP but also with large volumes of data. As a consequence, more sophisticated algorithms and system architectures are required.

We observe that there is a clear and increasing interest in real-time analysis of microblogs. Systems such as TwitterViz[3], Geo and Temportal Association Creator (GTAC) [4], among others [5, 6, 7], have been proposed as means to visually understand what people post on OSNs. Nonetheless, most of the current solutions focus on better methods and visualizations to for this task without taking into account the large volumes of data that may occur during certain events. Samplings can be used reduce the volume of data to be processed by a system. Although in many cases a sample of the data can represent well all the remaining data, this is not guaranteed as observed with the sampling provided by Twitter API [19]. Thus, a system that is able to process all the microblogging data to which it is subjected consists of a way to assure the accuracy of the analytics reports to be provided.

2.2 IBM InfoSphere Streams Platform

Systems for real-time processing of large volumes of data must be highly available, reliable and fault-tolerant, as well as scalable to higher demand. In this context, Streams is a software platform that enables the development and execution of applications that process information in data streams. Streams has been developed by IBM to enable continuous and fast analysis of massive volumes of moving data with low latency. This makes the platform suitable to handle the large volume of posts on social media streams like Twitter.

Streams has been designed to aid developers of parallel and high performance stream processing systems with the capability of scaling over a range of hardware environments. In addition, the platform offers automated deployment of stream processing applications, incremental deployment without restarting to extend stream processing applications, and a secure and auditable execution environment.

For the development of applications for Streams, the Streams Processing Language (SPL) [20] provides a programming language and runtime framework to support streaming applications and it has integration with several edge adapters that can connect to external sources or applications. SPL is based on the idea of creating workflows of operators. An operator represents a class of manipulations of tuples from one or more input streams to produce tuples on one or more output streams. An operator invocation is a specific use of an operator with specific assigned input and output streams with locally specified parameters, logic, etc. Many operators have one input port and one output port; others have: a) zero input ports: source adapters, e.g., TCPSource; b) zero output ports: sink adapters, e.g., FileSink; c) multiple output ports, e.g., Split; or d) multiple input ports, e.g., Join. A composite operator is a collection of operators. i.e. an encapsulation of a subgraph of primitive operators (non-composite) or composite operators. Even though entire applications can be developed in SPL, programmers have also the possibility to create native operators in C++ or Java.

3 Proposed Architecture

In this section, we present the proposed architecture for real-time analysis of microblogging messages. This architecture is designed to process a stream of messages provided by a microblog service such as Twitter and, after a pre-defined interval of time has elapsed, to output analytical results regarding the past time frame. Hereafter, we refer to these results as interval's statistics.

Given the real-time nature of this application, system availability and scalability are two important factors to avoid issues with hardware failure and volume unpredictability. For this reason, the proposed architecture follows a modular pipeline structure to make it easier the deployment on parallel environments and take advantage of a distributed architecture. This architecture can then be extended with parallel workflows to deal with large volumes of data. Then, an appropriate middleware such as Streams can handle the availability requirement.

We describe the overall architecture in a more high-level way in the remainder of this section. In addition, we discuss some insights about employing it in practice, mainly to deal with high performance requirements.

3.1 Main Workflow

The proposed architecture is shown in Figure 1. This architecture consists of one pipelined main workflow divided into the following modules:

1. **Relevant text selection:** consists of evaluating if the text in the current message is relevant or not for the application, considering domain knowledge. Unrelated messages are discarded to save computing resources.
2. **Tokenizer:** this process consists of dividing the text into a list of sub-strings, i.e. tokens, representing meaningful pieces for the given domain.
3. **Preprocessing:** the tokenized text is processed for variability reduction, according to language and domain.
4. **Topic classification:** this step consists of finding topics of interest. All topics found are added to the topics output list. If no topic is found the tweet is discarded.
5. **Sentiment classification**: an algorithm is used to classify the sentiment of the text, considering a pre-defined set of classes, e.g. positive, negative and neutral.
6. **Postprocessing**: the main goal of this module is to prepare the previously computed tokens for the subsequent statistical analysis.
7. **Statistics computation**: in this module we compute the main statistics for the interval. These statistics are related to topics, terms, users, and so forth, and the sentiment of the texts to which they are related. After the predefined duration of the interval is over, the results are sent to the output.

Different methods can be used to implement each of the aforementioned modules, according to the application, domain and language.

The input, i.e. the text stream, can be provided by a microblogging service such as Twitter. Nonetheless, these messages must be sorted in terms of posting time, i.e. the messages must be in the same order that they have been posted.

Notice that this architecture depends on domain and language knowledge in some of its modules. For instance, a list of general-domain words is useful for semantic and lexical disambiguation. Some domain-specific synonyms can also help the system, for instance in the soccer domain the nickname of a player can be used to normalize to a unique token the different variations of tokens referring to the same player. The use of such knowledge is detailed in the next section.

3.2 Distributed Modules to Deal With High Volumes of Data

High performance computing is crucial for the proposed architecture to provide near-real-time responses for the analysis of OSNs, especially when high volumes of data become available. Given that parallel and distributed infrastructures are very common nowadays, the proposed structure was designed to take advantage of parallel computing in the following ways.

The first way to optimize the processing of the input stream is to implement the main workflow in a pipelined fashion. That is, each module is available for processing a new message even though subsequent modules might still be processing the current tweet. The proposed

architecture itself allows for this since no module needs to wait until the processing of the subsequent modules ends. Nonetheless, some middleware for distributed computing, such as Streams, is necessary for realizing this in practice.

Moreover, the main workflow can be decoupled to process several messages at once. In Figure 1, the modules marked with * (modules 1 to 7 described above) can run in parallel, provided that their outputs are synchronized in the end of the split. Nonetheless, different splits can be used for each module if necessary.

4 A Case Study on Soccer Events: Architecture Implementation

In this section, we describe both the implementation and test of a system that implements the architecture proposed in the previous section. This system was used to conduct real-time analysis of Twitter posts from Brazilian soccer fans during the 2013 FIFA Confederations Cup.

4.1 Use case description

The main goal of the proposed system was to analyze, in real time, the sentiment of tweets about the Brazilian team during each game of the 2013 FIFA Confederations Cup. The FIFA Confederations Cup consists of an 8-team soccer competition that is organized one year before the FIFA World Cup Finals, both hosted in same country. The 8 teams comprise each of FIFA's six continental champions, the host nation and last World Cup's champion. The 2013 edition, hosted in Brazil, included the following national teams: Brazil, Italy, Japan, Mexico, Nigeria, Spain, Tahiti, and Uruguay. Italy represented the European continent since Spain has won both the last World Cup and the Euro Cup, and Italy was runners-up in the latter.

The expected main outcome of the monitoring system for this specific use case was a sentiment-aware summary of what the users posted related to the team after pre-defined intervals. This includes both an open set of the most frequent terms (and co-occurrences of terms) and a closed set of entities (herein referred as topics) along with the number of posts related to each category of sentiment. For this real-time monitoring task, only tweets in Portuguese and supposedly from Brazilian users, i.e. fans of the Brazilian team, needed to be taken into account. The monitoring timeframe consisted of tweets from about one hour before the start until about one hour after the end of game.

It is worth mentioning that this edition of the confederations cup presented an interesting context to social media analysis. The competition took place just one year prior to the 2014 FIFA World Cup Finals, which is a very special edition of the World Cup to Brazilian people for two reasons: a) the country was going to be hosting again the most important soccer tournament in the world after 64 years; and b) they are very passionate about soccer. As a consequence, a high number of posts from Brazilian users was expected on social media during the event, even though the exact volume was unknown.

4.2 System Design

In this section, we explain how the proposed architecture was implemented for this use case, where additional details on the implementation of each module shown in Figure 1 are provided. Later, we also present details on how to increase the application performance.

4.2.1 *Relevant text selection*

In order to find only the tweets related to soccer, a set of dictionaries containing domain-specific knowledge was considered, e.g. player names and common soccer moves, and look for these entries in the text of each tweet.

Five different dictionaries were considered. The first one contains words that present very strong evidence of the application domain, for instance soccer players' Twitter accounts. Only by containing one of these words the tweet is considered as relevant. The second dictionary was composed of more common words that should appear together at least twice, for instance 'penalty' and 'kick'. The third one includes words that are useful to know that tweet is not relevant, even though it contains some words that might be associated to soccer but to other domains as well. The fourth and fifth dictionaries contain adjectives (positive and negative) and common names of players, respectively, to add context to the second or fourth dictionary. These dictionaries are evaluated in this sequence using an Annotation Query Language (AQL) query [21]. In the end, a tweet is selected based on the count of words in these dictionaries. Only the tweets that contain a word from the third dictionary, or that do not any word from the remaining ones, are discarded.

4.2.2 *Tokenizer*

In this module, the tweets are decomposed into words, punctuations, numbers, URLs, usernames, and hashtags. Blank characters, such as spaces, tabulations and new lines, define delimiters for the tokenizer except when punctuations appear together with another type of token. In this case, the punctuation and the other token are decoupled.

Domain knowledge is used to find proper nouns composed of more than one token, for instance 'Mario Balotelli'. In this case, after the tokens 'Mario' and 'Balotelli' are found, a dictionary containing composed words is used to convert them to a single token, i.e. 'Mario Balotelli'.

Moreover, if some punctuation occurs at least twice in a row, these occurrences are converted to a single token representing that punctuation but with more intensity, which might indicate an evidence for sentiment analysis (for instance the sequence '!!!!!!!' is converted to the token '!!').

4.2.3 *Preprocessing*

The tokenized text is then processed in the following way. First, all URLs and usernames are converted to special tokens representing all URLs and usernames, in this case '__URL__' and '__USERNAME__'. Next, the tokens containing non-alphanumeric characters are removed, except if they represent some special token such as a punctuation. Then, the tokens that contain several repeated occurrences of the same letters are normalized, if possible, to the correct word with the help a language dictionary. Afterwards, all words in a pre-specified black list are removed, e.g. stop-words [22]. Finally, semantic and lexical word disambiguation is conducted on words that yield similar meaning or that are mistyped (either on purpose or by mistake), i.e. with the aid of a domain dictionary, we check all tokens that may represent the same entity or expression and convert them to a single token that represent the same concept.

4.2.4 *Topic classification*

For this task, a lexicon-based topic categorizer is used. By considering a list containing all desired topics, in our case the players, the coach, the team itself and the referees, we compare all current tokens with the entries in this list. The output of this module consists of a list containing all topics that have been found in the tokens. If the output list is empty, the tweet is discarded.

4.2.5 *Sentiment classification*

A Naïve Bayes classifier performs sentiment classification of the tweets into three distinct classes: positive, negative, and neutral. Even though domain knowledge is not explicitly used in this module, domain knowledge is implicitly used by means of the previous pre-processing modules and the training of classifier, which benefits from the use of domain-specific training data.

The classifier has been trained with a corpus comprising tweets posted during four friendly matches played by the Brazilian team in 2013, before the Confederations Cup, against the following countries: Italy, England, Bolivia and Chile. About 1 million of tweets were gathered from these games, and a total of 2,092 tweets (523 per game) have been labeled with the corresponding sentiment. By considering a cross-validation scheme, recognition rates of about 65% were achieved on the test set using the combination of unigrams and bi-grams for the feature set.

4.2.6 *Postprocessing*

The goal of this module is to conduct stemming/lemmatization [22] to reduce the number of different words that the system might output, and to remove other unwanted tokens that were not removed in the preprocessing because they were useful for sentiment analysis (bad words for instance) but are not useful for the interval's statistics.

For lemmatization, a two-step procedure is considered. First, a part-of-speech tagger is used to find the verbs in the current list of tokens, and then these verbs are stemmed. For removing the remaining unwanted tokens, a manually defined black list is considered.

4.2.7 *Statistics computation*

In this module, the statistics for each interval are computed. Such statistics take into account the topics, terms, co-occurrences of terms and co-occurrences of topics with terms that appeared in the interval. That is, the goal is to count the number of occurrences of each of these types of statistics, for each sentiment, and to output those that occur the most during the given time window.

The process of finding terms is pretty straightforward: each token corresponds to a term except when it is in the topics list. In that case, the token is a topic. Pairs of terms are computed by taking into account the co-occurrence of terms up to distance D, which is also pre-defined (we set D to 5 in this work). Pairs of topic/term are similar to the latter, but it is considered the co-occurrence of a topic with a term. All occurrences of these types of information are stored in a map containing the topic, term, pair of term, or pair of topic/term (a different map is used for each type) as the key, and the value corresponds to the number of tweets observed for each

sentiment class in the current interval, i.e. a single dimension array containing one position for positive, one for negative and another for neutral. After this is computed, the map is updated. If a key is in the map already, the value corresponding to the current sentiment of the tweet is incremented by one. Otherwise, a new key is inserted with value 1 for the current sentiment and 0 for all the others.

When the interval is over, i.e. a tweet from the next interval has been observed in the input, the maps are sorted in a descending way, by considering the sum of all sentiments. Then, these sorted lists are saved as the statistics for interval *i*, and the maps are reset to starting computing the statistics for the new interval, i.e. interval *i +1*.

4.2.8 *Implementation on Streams*

This system has been implemented on the InfoSphere Streams platform, using the SPL language. By designing each module as a different SPL operator, the platform manages a pipeline that can process several messages at the same time, according to the availability of each module. This greatly helps the system to deal with a large volume of messages in the input.

In addition, special operators allowed us to implement the parallel workflows mentioned in the previous section. To increase even further the performance, some modules have been decomposed into sub-modules, containing intermediary steps of the corresponding algorithms, to take better advantage of a distributed computing environment.

It is worth mentioning that Streams is also the middleware that allows for transparently handling system availability by means of a cluster of computers. That is, the platform manages the reallocation of processing elements whenever a node of the cluster becomes unavailable.

5 Architecture Validation

The implemented system was tested during several games of local Brazilian soccer competitions and other friendly matches of the Brazilian national team. The system was put in production during the FIFA 2013 Confederations Cup during all five games of the Brazilian team, against the following opponents: Japan, Mexico, Italy, Uruguay and Spain.

5.1 Analytics Results

Although the system was designed to handle variations of user activity, our experience running it during these games was beyond the expectations. Game followers react extremely quickly and in large numbers to the main events of a game.

Figure 2 shows the number of tweets (grouped in blocks of 5 minutes) during the final of the tournament, the game Brazil 3x0 Spain (Brazil versus Spain, where Brazil won the match scoring 3 goals against none from Spain). We started collecting about one hour before the game started (at 6:00 PM, considering Brazil's local time zone) and finished the collection about half an hour after the end of the game (at 9:30 PM). The graph depicts only the tweets selected for analysis, and the proportion between tweets classified as positive, neutral, and negative in each 5-minute block.

The number of tweets presents in Figure 2 follows closely the main events of the game. In the first half, an impressive defense of a Brazilian defender almost increased 5-fold the number of tweets produced by the fans (interval 18). Following that move, the goals scored by the Brazilian team resulted in a large number of messages, generally more than 60 thousands messages in the five-minute time windows (intervals 20, 23 and 28). Other events, such as a penalty kick defended by the Brazilian goalkeeper in the second half, were slightly less pronounced but a great deal of messages can be observed as well (interval 29). We also see a lot of chat after the game ends, in a pattern of decay taking about 30 minutes that we consistently observed in the other games (after interval 33, marked with 90').

The focus and polarity of the comments also change wildly as the game unfolds, following the performance and events of individual players. Figure 3 shows the progression of positive tweets for 5 players of the Brazilian national team during the same game. There is a strong positive showing for the defender David Luiz right after he performed an amazing defense for his team, when the ball was almost crossing the goal line. It is worth noting that this was the most praised move of the entire competition. Notice also that the peak of positive sentiment happens some minutes after the event, since it takes some time for the audience to react to it. Similarly, there is a peak for a striker, Neymar, just after David Luiz's defense when he scored the first goal of the Brazilian team and of the match.

We have observed that not all major conversations are directly tied to game plays. We have observed that the negative sentiment towards the Brazilian forward Hulk is strong throughout the game, sometimes peaking following other events of the game not shown in the graph. Brazilian fans have expressed, consistently during all five games during the cup, that they do not want him in the team. But even popular players like Oscar can drive negative sentiment when they do not play well, as it was observed in other games of the tournament.

Finally, we have also observed in this and other games that there are occasions in which a parallel conversation emerges during the game, triggered by an event of the game. For example, in the Brazil 2x1 Uruguay game, one of the semi-finals, a player from Brazil provoked another player by sending kisses when he was substituted in the second half. Comments about this event evolved into a more complex conversation about kissing and irony, persisting well beyond the end of game, generating at least 7,000 tweets.

As this analysis shows, the volume of tweets during a large event such as the ones we have done varies wildly and is largely unpredictable. As noticed, game occurrences are not always direct good predictors of tweet volume. In other words, predicting demand in this kind of task is very difficult, so to adequately manage this kind of workload we need flexible but powerful architectures such as the one proposed in this paper.

5.2 Performance Analysis

Figure 4 shows the performance comparison for four different implementations of the proposed architecture. In this case we compare the performance of a standalone application, i.e. a single-host executable, with that of the distributed implementation running on Streams. The main difference is that in the former the programmer takes care of managing the distribution of the processing modules is done by the programmer, while in the latter the platform manages such

distribution. In addition, for each of them we also compare sequential (single) versus parallel (multiple) workflows to observe the benefits of this parallelization. These results consider the Brazil versus Spain game, which presented the largest volume of posts, and were run on a single host with four processors. Notice that, in this case, performance corresponds to the time, in seconds, to process each block of data in a five-minute interval. Since each experiment has been repeated five times, the average time is presented.

It is clear in Figure 4 that the use of Streams presents a significant boost in performance against the standalone code. A better view of this superior performance can be observed with the speed up factor shown in Figure 5, where the implementations running on Streams presented an average speed up of 2.11 and 2.55, for the sequential and parallel workflows, respectively. We observe a significant boost in performance owing only to the use of Streams, which shows how well designed are its algorithms to distribute processing on parallel architectures and deal with sequential processing flows. Moreover, we observe a clear increase in performance of the parallel workflows, especially when the volume of data is larger. This shows that the proposed architecture can scale up to a higher number of messages, and the processing time of each interval can be made constant with the addition of more processing power, i.e. more hosts and/or processors (in this work we considered a single host for the sake of simplicity).

6 Conclusions and Future Work

In this paper, we presented an architecture for real-time sentiment analysis of social networks, aiming at keeping track of what users post during a given event and the sentiment of their tweets about it. The pipelined and parallel architecture was implemented on the IBM InfoSphere Streams for high system availability and scalability, to conduct machine learning-based sentiment analysis of Twitter posts.

The system was tested during the 2013 FIFA Confederations Cups, to analyze the messages posted by Brazilian users (in Portuguese) during each game of the Brazilian national team. The architecture was robust enough to cope with the volume of data presented during these events, and we observed that the statistics computed by the system are strongly related with the facts of the games.

As future work, the evaluation of the architecture on other types of events, for instance marketing campaigns, may help evaluating how the system can be adapted and used in other domains. Consequently, faster methods to conduct these adaptations would be very helpful, such as unsupervised methods to find abbreviations, synonyms, etc. In addition, it would be important to define some metric to define the capacity of the system given the resources available, e.g. number of processors, number of hosts, etc.

References

[1] K. Ehrlich and N. S. Shami, “Microblogging inside and outside the workplace”, in *Proc. of the 4th Intenational AAAI Conferences on Weblogs and Social Media (ICWSM)*, Washington, DC, USA, 2010.

[2] Twitter Inc., About [Online], Available: https://about.twitter.com/company.

[3] E. Kandogan, D. Soroker, S. Rohall, P. Bak, F. Van Ham, J. Lu, H.-J. Ship, C.-F. Wang and J. Lai, "Architetural Patterns For Real-Time Visual Analytics on Streaming Data", in *Proc. of the IEEE Symposium on Large Data Analysis and Visualization 2013*, Atlanta, GA, USA, 2013.

[4] T. Kraft, D. X. Wang, J. Delawder, D. Wendwen, Y. Li and W. Ribarsky, "Less After-the-Fact: Investigative Visual Analysis of Events from Streaming Twitter", in *Proc. of the IEEE Symposium on Large Data Analysis and Visualization 2013*, Atlanta, GA, USA, pp. 95-103, 2013.

[5] J. Chae, D. Thom, Y. Juan, S. Kim, T. Ertl and D. S. Ebert. "Public behavior response analysis in disaster events utilizing visual analytics of microblog data", *Computers & Graphics*, vol. 38, pp. 51-60, 2014.

[6] C. A. Guille, H. Hacid and D. Zighed. "An open source platform for social dynamics mining and analysis", in *Proc. of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, pp. 1005-1008, 2013.

[7] T. Shelton, A. Poorthuis, M. Graham and M. Zook. "Mapping the data shadows of hurricane sandy: Uncovering the sociospatial dimension of 'big data'". *Geoforum*, vol. 52, pp. 167-179, 2014.

[8] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos and C. Moran, "IBM InfoSphere Streams for Scalable, Real-time, Intelligent Transportation Services", in *Proc. of the 2010 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, pp. 1093-1104, 2010.

[9] F. Liu, F. Weng and X. A. Jiang, "A broad-coverage normalization system for social media language", in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, Stroudsburg, PA, USA, 2012, vol. 1, pp. 1035-1044.

[10] A. P. Felt and D. Wagner, "Phishing on mobile device", in *Proc. of Web 2.0 Security & Privacy 2011*, Oakland, CA, USA, 2011.

[11] A. Go, L. Huang and R. Bhayani, "Twitter sentiment classification using distant supervision", Tech. Rep. No. CS224N, Stanford University, 2009.

[12] G. Paltoglou and M. Thelwall, "Twitter, myspace, digg: Unsupervised sentiment analysis in social media", *ACM Transactions on Intelligent Systems and Technology*, vol. 4, pp. 1-66, 2012.

[13] A. Celikylmaz, D. Hakkani-Tur and J. Feng, "Probabilistic model-based sentiment analysis of twitter messages", in *Proc. of Spoken Language Technology Workshop (SLT)*, Berkley, CA, USA, 2010, pp. 79-84.

[14] A. Bakliwal, P. Arora, S. Madhappan, N. Kapre, M. Singh and V. Varma, "Mining sentiments from tweets", in *Proc. of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, Jeju, Korea, 2012.

[15] L. Li, Y. Xia and P. Zhang, "An unsupervised approach to sentiment word extraction in complex sentiment analysis", *Int. J. of Knowledge and Language Processing*, vol. 2, no. 1, pp. 40-52, 2011.

[16] A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. Jong and U. Kaymak, "Exploiting emoticons in sentiment analysis", in *Proc. of the 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal, 2013, pp. 703-701.

[17] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment analysis of Twitter data", in *Proc. of the Workshop on Languages in Social Media - LMS'11*, Stroudsburg, PA, USA, 2011, pp. 30-38.

[18] B. Pang and L. Lee, "Opinion mining and sentiment analysis". *Foundations and Trends in Information Retrieval*, vol. 1, no. 2, pp. 1-135, 2008.

[19] F. Morstatter, J. Pfeffer, H. Liu and K. M. Carley, "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose", *Proc. of the 5th Intenational AAAI Conferences on Weblogs and Social Media (ICWSM)*, 2013.

[20] M. Hirzel, H. Andrade, B. Gedik, G. Jacques-Silva, R. Khandekar, V. Kumar, M. Mendell, H. Nasgaard, S. Schneider, R. Soule and K.-L. Wu, "IBM Streams Processing Language: Analyzing Big Data in motion", *IBM J. Res. & Dev.*, vol. 57, no. 3/4, pp. 7:1-7:11, 2013.

[21] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan and H. Zhu, "SystemT: a system for declarative information extraction", *ACM SIGMOD Record*, vol. 37, no. 4, pp. 7-13, 2009.

[22] S. M. Weiss, N. Indurkhya and T. Zhang, *Fundamentals of Predictive Text Mining*, London, UK: Springer-Velag, 2010.

Paulo R. Cavalin *IBM Research – Brazil, Av. Pasteur 138&146, Rio de Janeiro, RJ, Brazil (pcavalin@br.ibm.com).* Dr. Cavalin is currently a Research Staff Member in the Social Data Analytics Group of IBM Research – Brazil. He received a Ph. D. degree in Automated Production engineering from École de Technologie Supérieure (ÉTS) - Université du Québec in 2011, a M.Sc. Degree in Applied Informatics from Pontifícia Universidade Católica do Paraná (PUCPR), in 2005, and a B.Sc. degree in Informatics from Universidade Estadual de Ponta Grossa (UEPG), in 2002. He joined IBM in 2012, conducting both theoretical and applied research in Pattern Recognition, Machine Learning and Computer Vision.

Maíra A. C. Gatti *IBM Research – Brazil, Av. Pasteur 138&146, Rio de Janeiro, RJ, Brazil (mairacg@br.ibm.com).* Dr. Gatti is a Research Staff Member (RSM) in the Social Data Analytics group at IBM Research, Brazil. She joined the Lab in February 2011, when it just started the operations in Rio de Janeiro. She works with Big Data Analytics and her main area of expertise lies in the Computer Science field and specific areas ranges from Distributed Systems to Multi-Agent-based Simulation. Dr. Gatti holds a Ph.D. (2009) and a M.Sc. (2006) degree in Software Engineering from Informatics Department of the PUC-Rio, Pontifical Catholic University of Rio de Janeiro, Brazil. She has been a research fellow at the Agent and Intelligent Systems of Kings College London, England (2009), and prior to that a visiting student at the University of Waterloo, Canada (2008).

Tiago G. P. Moraes *IBM Software Group, Av. Tutóia 1157, São Paulo, SP, Brazil (tgomesm@br.ibm.com).* Mr. Moraes obtained a B.Sc. degree in Computer Science from Pontifícia Universidade Católica de São Paulo (PUCSP), in 2002, and has about twelve years of experience in the Information Technology market, being six of them with solutions for Business Intelligence and Data Warehouse. He joined IBM in 2011 and currently works in presales to model customer needs with IBM solutions, in special IBM Big Data solutions since 2012.

Fabio S. Oliveira *IBM Software Group, Av. Pasteur 138&146, Rio de Janeiro, RJ, Brazil (fsolive@br.ibm.com).* Mr. Oliveira holds a B.Sc. degree in Computer Science from Universidade Católica de Petrópolis, since 2001, and has about fifteen years of experience in large Brazilian companies, in projects involving integrating of data governance and quality. In 2005 he joined IBM to work in projects related to business intelligence outsourcing, and in 2007 he moved to the Software Group to become an expert in Big Data and data governance.

Claudio S. Pinhanez *IBM Research – Brazil, Av. Tutóia 1157, São Paulo, SP, Brazil (csantosp@br.ibm.com).* Dr. Pinhanez is a researcher, professor, and media artist. He is the leader of the Social Data Analytics group of IBM Research-Brazil, where he has been a research scientist since 2009, working on Social and Human Data Analytics, Service Science, Ubiquitous Computing, and Human-Computer Interfaces. He was one of the founding members of IBM Research - Brazil and previously was a researcher at the Watson Research Center of IBM Research from 1999 to 2009. Dr. Pinhanez got his PhD. in 1999 from the MIT Media Laboratory; and Master and Bachelor's degrees from the University of São Paulo. From 1987 to 1997 he was a faculty member of the department of Computer Science of the University of Sao Paulo. He has also been a visiting researcher at the ATR-MIC Laboratory in Kyoto, Japan in 1996, and at the Sony Computer Science Laboratory in Tokyo, Japan in 1998.

Alexandre Rademaker *IBM Research – Brazil, Av. Pasteur 138&146, Rio de Janeiro, RJ, Brazil (alexrad@br.ibm.com)*. Dr. Rademaker holds a B.Sc. degree from Universidade Federal do Rio de Janeiro (UFRJ), a M.Sc. degree from Universidade Federal Fluminense (UFF), and a Ph.D. degree from Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), all in Computer Science, obtained in 2001, 2005 and 2010, respectively. He has experience in computability, computational models, knowledge representation and reasoning, acting in the following subjects: description logic, proof theory, ontology, functional programming, and category theory.

Rogério A. de Paula *IBM Research – Brazil, Av. Tutóia 1157, São Paulo, SP, Brazil (ropaula@br.ibm.com)*. Dr. de Paula is a research manager at IBM Research – Brazil, leading the Social Enterprise Technologies Group. He is also member of the Center for Social Business at IBM Research. He has more than 10 years of experience conducting empirical qualitative research in the design, use, and adoption of collaborative technologies. He is particularly interested in models and patterns of social interactions in people's everyday life and work. In that, he studies the meanings, values, and practices that are created as people carry out their daily affairs through and around technology. At IBM, his research focuses on understanding the human aspects of large-scale service practices in order to devise new service models, technologies, and theories to shape and improve IBM's social business solutions. Currently, he studies the emergence of 'work' networks--particular form of complex, heterogeneous social networks--that emerge from everyday interactions within and across organizations and social groups at work.

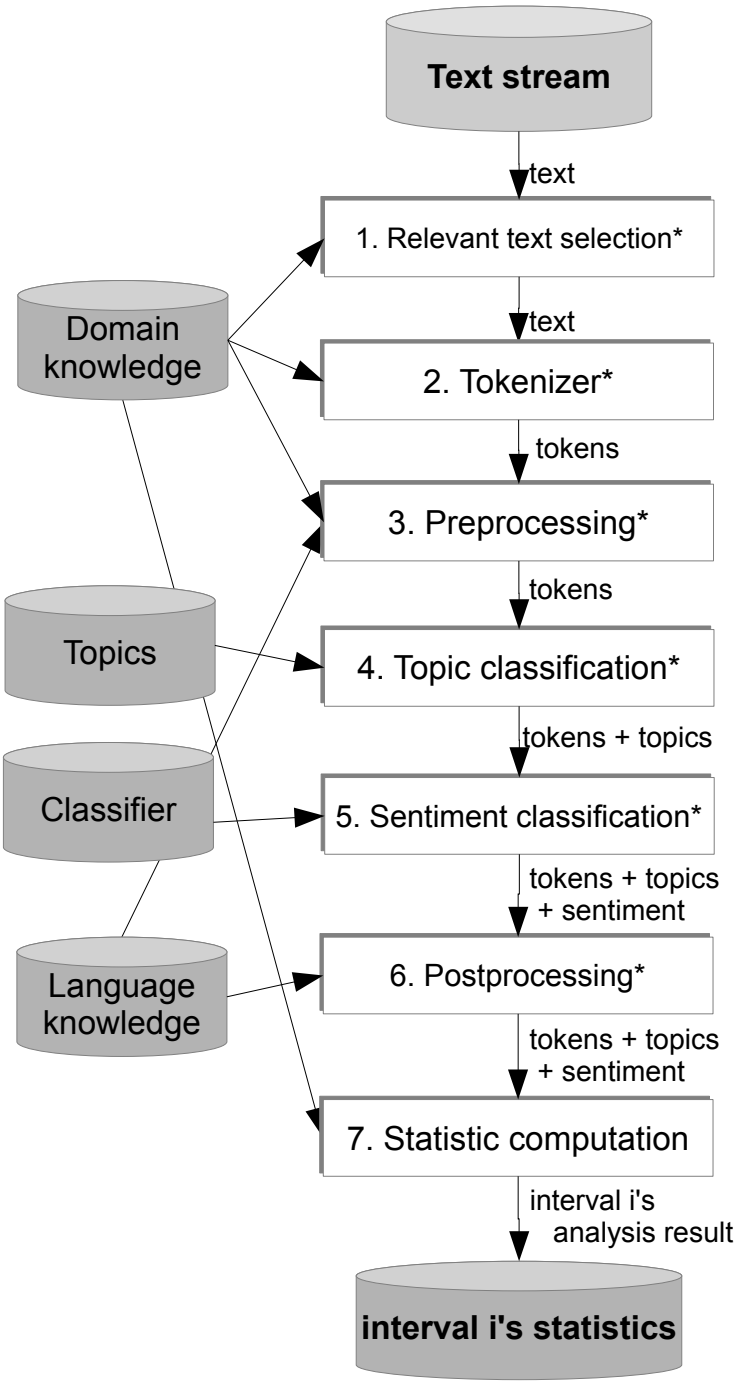

Figure 1: Main workflow. Modules marked with * can be split into parallel workflows.

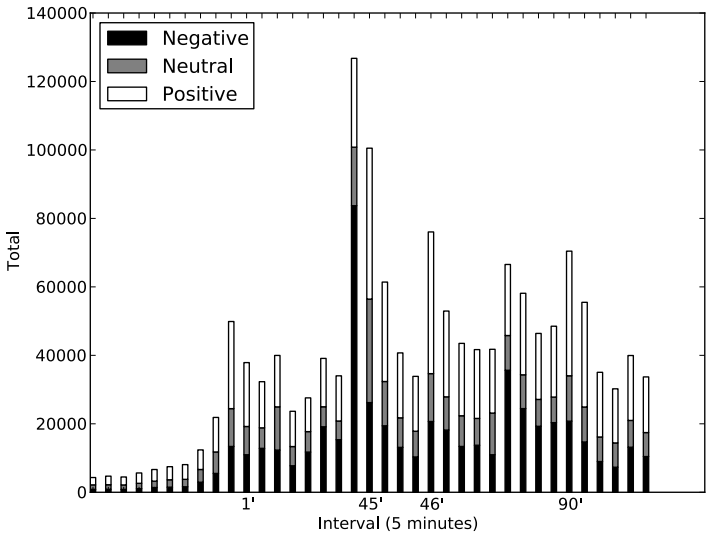

Figure 2: Total number of tweets posted during the Brazil vs Spain game, separated by sentiment. The first half of the game started at the interval marked with 1', and ended at 45'. Correspondingly, 46' and 90' relate to the start and end of the second half.

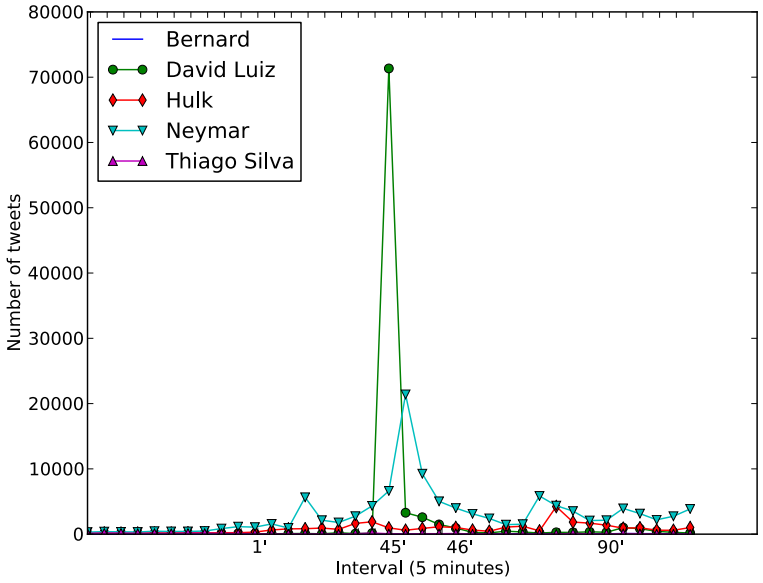

Figure 3: Progression of positive tweets for 5 selected players of the Brazilian national team (Brazil vs Spain game). The intervals marked with 1', 45', 46' and 90' are analogous to what is described in Figure 2.

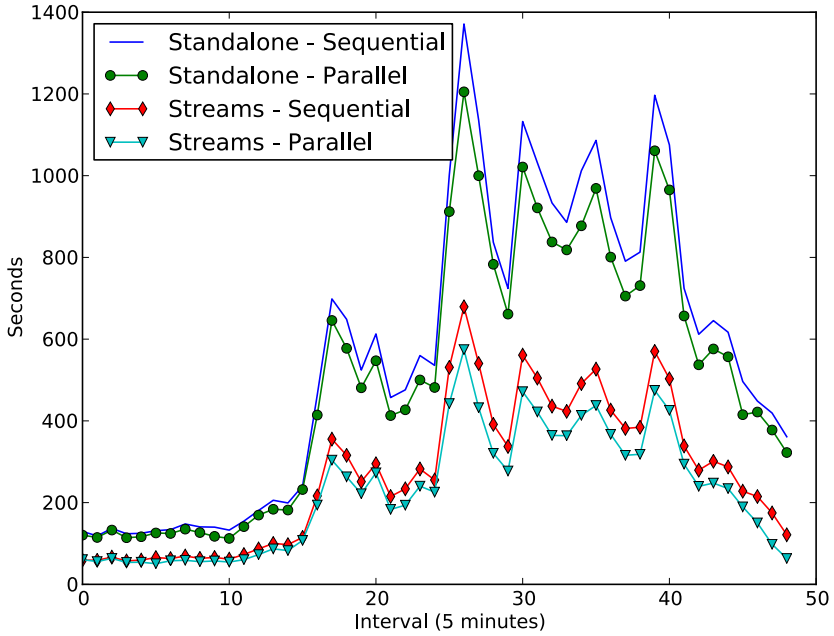

Figure 4: Overall performance of different implementations of the proposed architecture

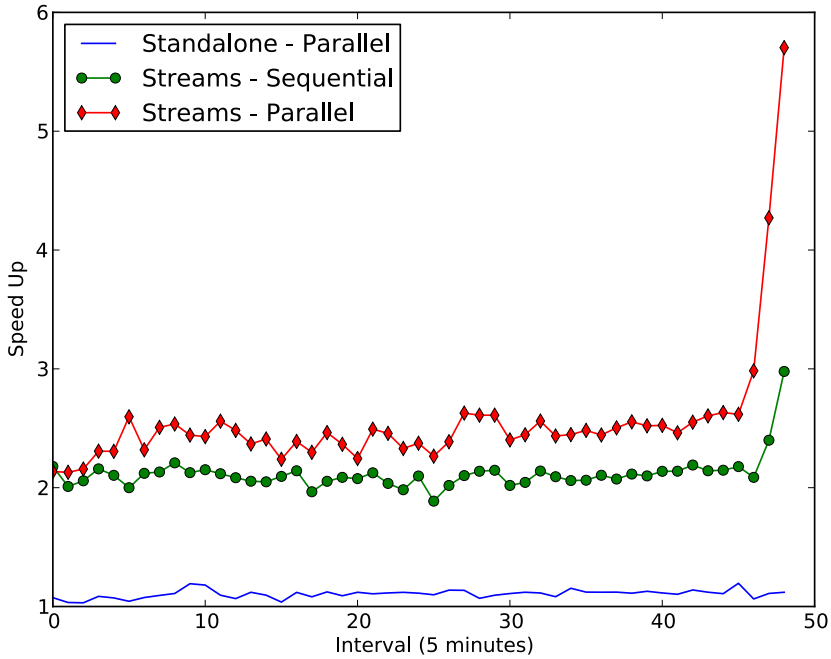

Figure 5: speed up factor of distributed and sequential implementations on InfoSphere Streams compared with the Standalone – Sequential implementation.