Pääkkönen and Jokitulppo *J Big Data* (2017) 4:6
DOI 10.1186/s40537-017-0066-7

Journal of Big Data

SHORT REPORT Open Access

Quality management architecture for social media data

Pekka Pääkkönen[1*] and Juha Jokitulppo[2]

*Correspondence:
pekka.paakkonen@vtt.fi
[1] VTT Technical Research Centre of Finland, Oulu, Finland
Full list of author information is available at the end of the article

Abstract

Social media data has provided various insights into the behaviour of consumers and businesses. However, extracted data may be erroneous, or could have originated from a malicious source. Thus, quality of social media should be managed. Also, it should be understood how data quality can be managed across a big data pipeline, which may consist of several processing and analysis phases. The contribution of this paper is evaluation of data quality management architecture for social media data. The theoretical concepts based on previous work have been implemented for data quality evaluation of Twitter-based data sets. Particularly, reference architecture for quality management in social media data has been extended and evaluated based on the implementation architecture. Experiments indicate that 150–800 tweets/s can be evaluated with two cloud nodes depending on the configuration.

Keywords: Quality attribute, Quality metric, Quality policy, Spark, Cassandra, Word2Vec

Background

Social media data can be analysed for getting insights into the behaviour of consumers [1, 2] or businesses [3]. However, data to be analysed may have originated from a spam campaign [4] or contain false information [5]. Therefore, filtering may be needed before meaningful insights can be created. Machine learning methods have been utilised for analysing and predicting credibility of social media data [6–8]. Also, frameworks have been developed for data quality management in social media [9, 10]. However, reference architecture (RA) design for data quality management in social media data has only been addressed recently [11]. The RA is aimed for facilitating design of quality management aspects into implementation architectures of big data systems.

This research extends an earlier work [11], which focused primarily on social media data quality evaluation for decision making, and initial validation of metadata management architecture for big data systems. This work focuses on validation and extension of the earlier RA [11]. Particularly, the proposed architecture has been implemented into a business context, where a company utilised tweet sentiment analysis in product development. A tool has been developed for data quality management, which enables evaluation, filtering, and querying of tweet-related quality information based on user-defined

Springer Open

© The Author(s) 2017. This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

rules. Performance results indicated that 150–800 tweets/s can be evaluated on two cloud nodes depending on the configuration.

The document is structured as follows. First, related work is reviewed. Then, research question and research method are presented. Next, design of the implemented architecture is illustrated with unified modelling language (UML) views. RA for data quality management in big data systems has been enhanced based on the implementation architecture. The RA is evaluated based on technology selections and performance. Finally, main lessons learnt and future work is discussed. The "Appendix" includes a detailed data view of quality rules, and metadata.

Literature review

Reference architecture has been designed for big data systems based on realised implementation architectures of big data companies (Twitter, LinkedIn etc.) [12]. The RA consists of functionalities (rectangles), data stores (circles), and data flows (arrows) between them. In a big data system, data is typically extracted from various sources, and loaded/transferred into a temporary or raw data store. New information may be extracted from the data. Data may also be cleaned, combined or replicated. Processed data is typically analysed, transformed, and served for visualisation with different applications. A new architectural layer (Fig. 1) has been introduced to the RA [11], which covers management of metadata including quality aspects of social media data. Metadata is "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource" [13]. Metadata management covers extraction of metadata from social media data, and providing access to it [11]. Quality management refers to assigning values to quality attributes of related social media data sets [11].

Additionally, important theoretical concepts have been defined in earlier work for data quality management [11]. Quality of data can be understood with associated quality attributes. A quality attribute represents a single aspect of quality (e.g. timeliness) [11, 14]. Quality metrics [11] are used for measuring properties of a quality attribute. An

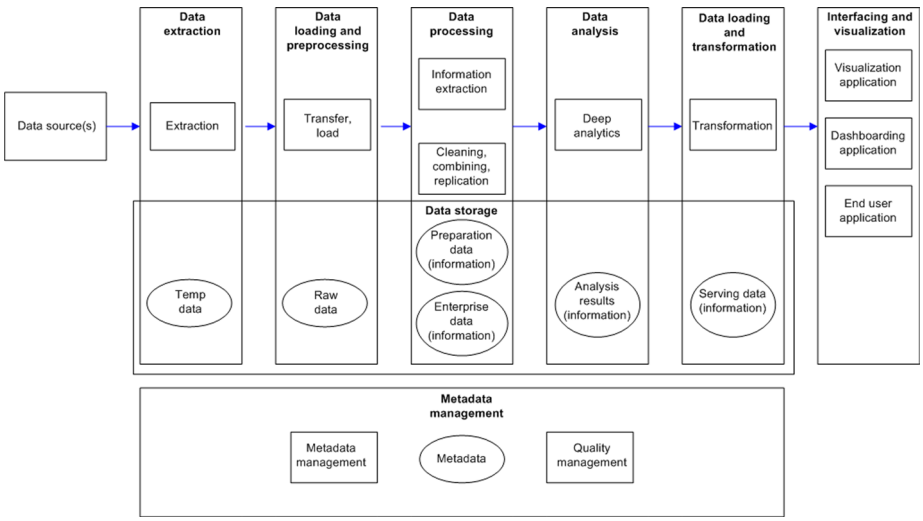

Fig. 1 Reference architecture for managing quality aspects of social media data [11]

example of a quality metric is a model, which evaluates timeliness based on the timestamp of a data item. Quality evaluation refers to evaluating the quality of information (data with meaning [11]), where context and intended use of information is taken into account [11]. Quality evaluation can be performed by utilizing quality policies [15]. The purpose of quality policies is to enable the organisation to define contexts for different situations, where social media data is utilised for decision making. An organizational quality policy defines acceptable data sources, and context of the task [11]. A filtering policy is used for evaluating only relevant quality dimensions to satisfy the requirements of a data consumer [11].

Also, other solutions have been developed for quality management of social media data. Social-QAS is a tailorable quality management service for social media content [16]. The tool enables quality assessment based on metadata, content, classification of messages, and scientific methods. End users can search for information with different quality parameters in a Facebook-application, which focuses on emergency services [17]. A dynamic quality evaluation concept has been implemented for supporting emergency situations [18]. Specifically, search of social media data can be filtered by weighting different quality parameters (e.g. timeliness, author reputation). A quality framework has been developed for supporting data quality profile selection and adaptation [9]. Particularly, the framework has been implemented for pre-processing of medical data. A hybrid approach has been proposed for quality evaluation across the big data value chain [10]. Quality of medical datasets has been evaluated for detection of cardiovascular risks, and sleep-disordered breathing.

Social media data analysis has facilitated decision making within an enterprise. For example, the role of Twitter-based analytics for supply chain management (SCM) has been studied [19]. Particularly, a framework was proposed for extracting intelligence from Twitter based on descriptive, content, and network analytics. Tweets have been analysed for gaining competitive intelligence in retail [3]. Particularly, sentiment, mentions, and topics were analysed in product categories. Social media channels (Facebook, Twitter) have also been integrated with processing of customer orders with an enterprise resource planning (ERP) system [20]. Several tools exist for validation of data from enterprise systems [21]. The tools typically evaluate uniqueness, conformance, completeness, accuracy, and consistency of data. The impact of data quality metadata for decision making purposes has been studied [22]. Results suggested that the use of data quality metadata may be enhanced with related training.

Methods and algorithms for social media data analysis can be categorized into network analytics, community detection algorithms, text analytics, information diffusion models and methods, and information fusion [23]. Text analytics covers natural language processing (NLP), information extraction, data mining, and machine learning approaches [23]. Information extraction refers to techniques for extracting entities, and their relationships from text. Information extraction has been utilised for entity extraction [24–26], and making tweet hashtag recommendations [27]. Neural network-based language models have been developed for estimation of word representations [28]. Particularly, high-quality word similarity vectors can be learned from billions of words based on continuous bag-of-words or skip-gram model. Google has also published pre-trained models as part of their Word2Vec-implementation [29]. Hashtag recommendation has been

proposed by learning of word representations with Word2Vec's skip-gram model, and a trained neural network [30]. Another application based on Word2Vec is restaurant recommendations based on similarities between tweets, and specified keywords related to foodborne disease symptoms [1].

Machine learning approaches can be categorized to supervised and unsupervised approaches [31]. Unsupervised learning approaches create a predictive model based on input data with a clustering method. Unsupervised approaches have been used for topical clustering of tweets [32, 33], and tweet hashtag recommendations [34]. Supervised methods utilize training data (input/output about the phenomenon) for classification or regression type of algorithms [31]. Supervised machine learning approach has been used for sentiment analysis of tweets [35]. Sentiment analysis has also been used as a factor in a regression model for predicting box-office revenues for movies [2].

More importantly, quality aspects of social media data have been analysed mainly with supervised machine learning. Automatic methods have been developed for assessing credibility of Twitter-based data sets [6]. Classifiers trained based on a similar approach have been developed for detecting newsworthiness and credibility of tweets [7]. TweetCred is a real time system for assigning a credibility score to a Twitter user's time line [36]. The usefulness of comments in Flickr and YouTube has been studied [8]. The results indicated that a few straightforward features can be used for detecting usefulness. A model has been trained based on user and tweet characteristics using supervised learning for identifying misinformation in Twitter (accuracy ~77%) [37]. CREDBANK is a corpus of tweets, topics, events, and associated credibility scores comprising more than 1 billion tweets [38]. Annotations for the corpus have been crowdsourced with Amazon Mechanical Turk, which focused on event detection and credibility detection of the events (a set of tweets).

The review indicates that many applications exist, where social media data analysis has facilitated decision making within an enterprise [3, 19, 20]. While several methods and algorithms can be utilised for social media data analysis [23], typically supervised learning based methods have been used for evaluating quality aspects of social media data [6, 36, 38]. In order to understand quality aspects of social media data, different approaches have been proposed for management of quality in social media data [9–11, 17, 18]. However, only some the approaches [10, 11] focus on development of architecture for managing the quality of data in a big data system. The contribution of this paper is validation of RA design for management of quality in social media data. Particularly, a proof-of-concept implementation of the RA has been created, which has been validated in product development context focusing on sentiment analysis of tweets.

Research question and method

This research was performed in collaboration with a company, which focuses on sentiment analysis of Twitter data. The company needs to understand and manage quality of tweets, which are extracted for analysis. Sentiment analysis itself should provide various insights into social media. Based on the collaboration the following research question was created:

- RQ1: How is quality management architecture constructed for evaluating the quality of social media data?

The research method follows the framework proposed for analysis and design of empirically grounded RAs [39, 40]. The approach consists of deciding a type for the RA (step 1), selection of a design strategy (step 2), empirical acquisition of data (step 3), construction of the RA (step 4), enabling of variability (step 5), and evaluation (step 6). In previous work, RA has been developed for big data systems [12]. Facilitation type of RA was selected (step 1), which should provide guidelines for the design of similar big data systems. The design strategy was selected as practise-driven (step 2). The RA was created based on realised implementation architectures gathered from publications and blogs (steps 3–4). In another earlier work, the developed RA was initially implemented and evaluated (step 6) for data quality management in social media sources, and a new metadata management layer was designed as part of the RA [11].

This research aims to extend the earlier RA [11]. First, the existing architecture implementation has been extended for acquisition of new empirical data (step 3). The implementation has been deployed into Eucalyptus cloud computing environment [41]. Particularly, a tool has been implemented, which enables management of quality aspects in Twitter-based data sets. Then, the existing RA [11] has been extended (step 4) and evaluated (step 6) based on new empirical data gathered from the implemented architecture (step 3). Variability aspect (step 5) has not been focused in the study.

Architecture design

This chapter presents design of the architecture implementation. The design has been presented from the point of view of utilizing the developed tool for quality management of Twitter based data sets. First, a use case view illustrates how a company (Invenco) can manage quality of social media data. Then, a high level data view of the system is presented. Deployment view illustrates how different components of the tool are executed in the target development environment. Finally, component and sequence views demonstrate how rules are created for data quality management, and how data quality is evaluated, and queried from the developed system.

Use case view

The use case view (Fig. 2) illustrates how the developed tool has been utilized for evaluating the quality of Tweet-based sentiment analysis. Invenco extracts social media data from Twitter, and provides the extracted tweets for data quality evaluation (by VTT). Additionally, it should be identified, which social media channels are utilized, and what dimensions of quality are valuable for decision making purposes. In our case, timeliness, relevancy, and popularity were selected as the most interesting quality attributes related to tweets. The information was specified by product/sales manager of Invenco. Subsequently, an organisational quality policy [11] was defined, which specified acceptable quality attributes for a social media data source (Twitter). Additionally, a profile may be utilized for definition of multiple organisational quality policies for a decision point. The provided information was utilized by VTT for data quality management.

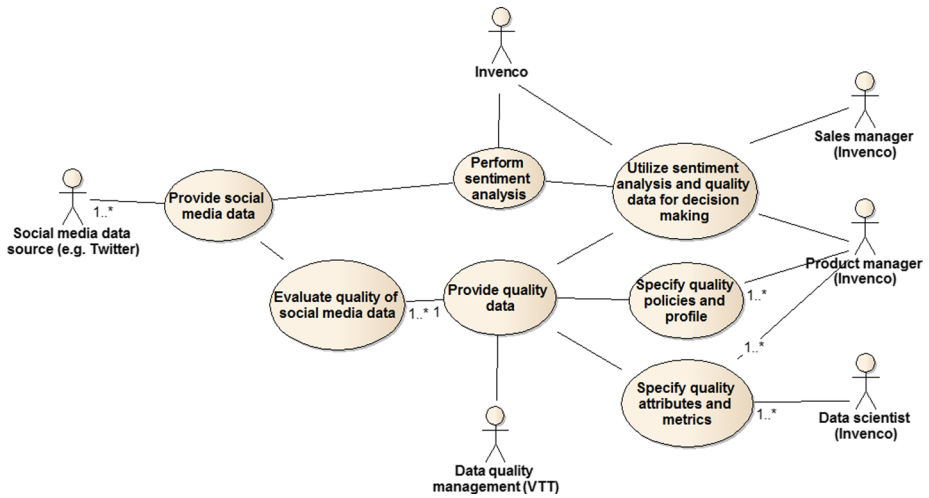

Fig. 2 Use case view to data quality management

Next, it should also be identified what level of quality is acceptable for an organisation. For this purpose, a filtering policy was defined, which specified an acceptable level for each quality attribute. The role of the filtering policy is to discard data with unacceptable quality from decision making. Also, quality metrics should be defined by a person, who understands basics of data science (e.g. a data scientist). Quality metrics specify how quality attributes are evaluated based on social media data under study.

For utilization of data for decision making, a search filtering policy was defined. The policy specified acceptable quality level for tweets to be returned in queries.

High level data view

Figure 3 presents a high level view of the most important architectural concepts. A data set refers to a group of data items (e.g. a set of tweets), which is extracted in the big data system from a social media data source. Each data set is associated with metadata, which contains structured information of the data set. Metadata dimensions include quality, navigational, process, descriptive, and administrative aspects [11] (Fig. 15). End user(s) of an organisation define quality rules for data quality management. Quality rules include organisational quality policy, filtering policy, profile, and search filtering policy. Additionally, quality attributes and quality metrics have to be specified. A detailed data view of all implemented concepts has been provided in the "Appendix".

Deployment view

Deployment view (Fig. 4) illustrates the main SW components, which are needed for data quality management. Front-end node provides services for management of data quality for end users, whereas Back-end node executes required processing for data quality evaluation. A node has been allocated specifically for relevancy evaluation of tweets.

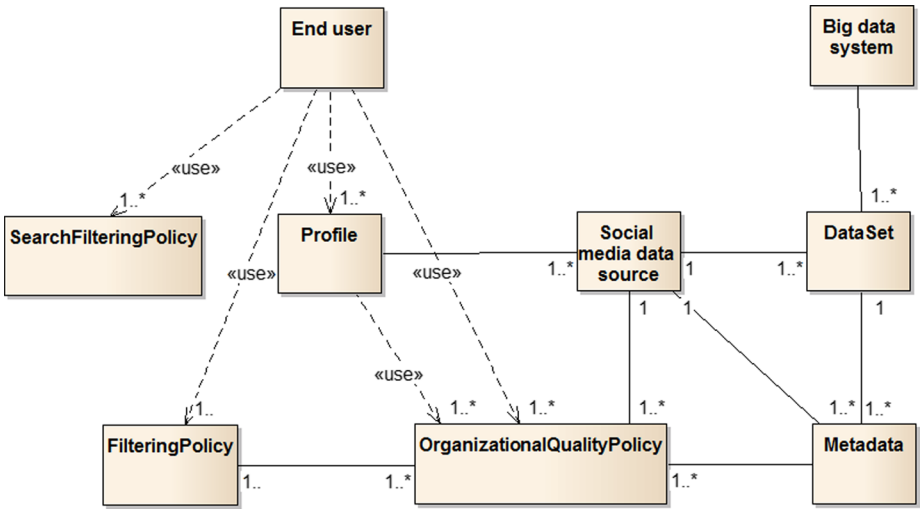

Fig. 3 High level data view of architectural concepts

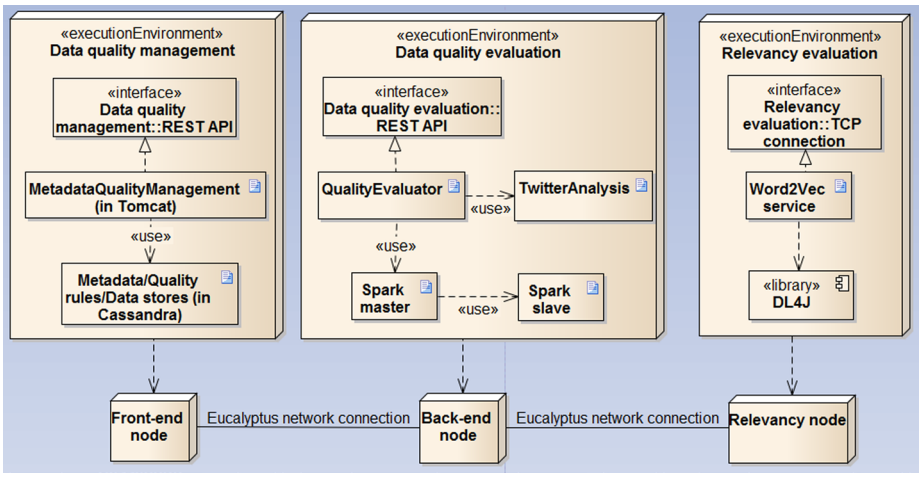

Fig. 4 Deployment view of the system

Quality rules-store contains data structures from the data view (Figs. 3, 14). Metadata is stored in metadata store. Data store contains social media data, which has been evaluated and validated successfully with a filtering policy.

The stores have been implemented to Cassandra, which is deployed in the Front-end node. The Front-end node also provides a representational state transfer (REST) API for interacting with end users (provided by MetadataQualityManagement). The Back-end node performs data quality evaluation of tweets. QualityEvaluator provides a REST API for execution of quality evaluation operations based on requests received from the Front-end node. TwitterAnalysis component is executed in Spark Streaming [42] environment. Especially, QualityEvaluator deploys TwitterAnalysis process to a Spark cluster. The Relevancy node performs relevancy evaluation of tweets. Word2Vec-service

provides a TCP socket interface for relevancy evaluation, which is utilized by the TwitterAnalysis-component.

Detailed component view: Front-end node

Figure 5 presents a detailed view of MetadataQualityManagement-component, which is executed in the Front-end node. Most of the components follow earlier design [Immonen], which has been extended in this work.

Responsibilities of revised and new components (in Fig. 5) have described in Table 1.

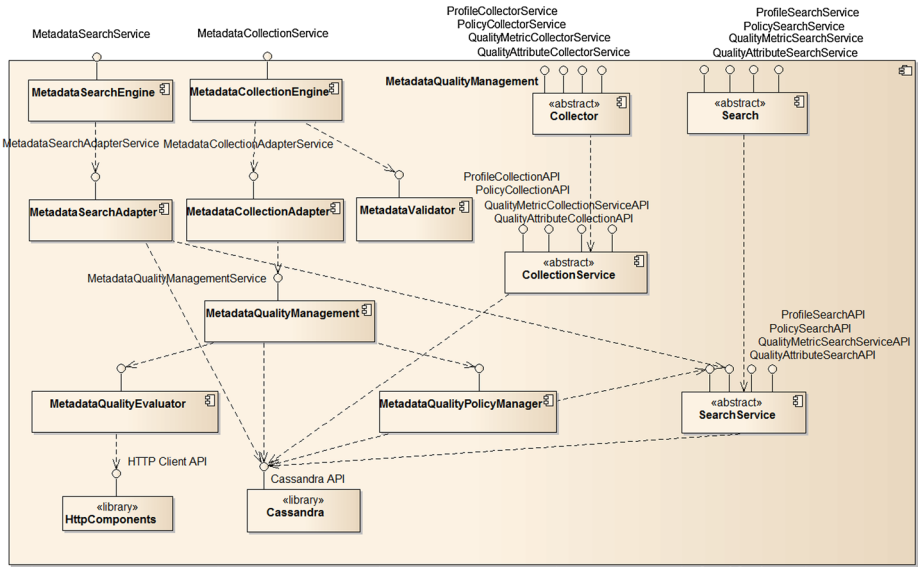

Fig. 5 Component view of MetadataQualityManagement, which is executed in the Front-end node

Table 1 Responsibilities of components for metadata quality management

Component	Responsibility
MetadataSearchEngine (revised)	Provides REST API for searching of metadata based on end user queries
MetadataCollectionEngine (revised)	Provides REST API for creation of metadata
MetadataSearchAdapter, MetadataCollectionAdapter (revised)	Adapters for interaction between REST API and lower SW components
MetadataQualityManagement (revised)	Manages quality aspects of metadata for data sets
MetadataQualityEvaluator (revised)	Manages quality evaluation for quality attributes
MatadataQualityPolicyManager (revised)	Manages quality policies for quality evaluation
MetadataValidator (revised)	Validates metadata received from MetadataCollectionInterface
Collector (new)	Components provide REST APIs for collection of quality policies, profiles, quality metrics, and quality attributes
Search (new)	Components provide REST APIs for searching of quality policies, profiles, quality metrics, and quality attributes
CollectionInterface (new)	Internal interfaces for saving of quality policies, profiles, quality metrics, and quality attributes to a database
SearchInterface (new)	Internal interfaces for searching for quality policies, profiles, quality metrics, and quality attributes from a database

Sequence view: creation of quality rules

Figure 6 presents how profile, organisational quality policy, quality metrics, and quality attributes are created by end users of an organisation (as described in Fig. 2):

Steps 1–2: Product manager creates a new profile for decision making, and defines a decision point ('Sentiment analysis')

Steps 3–4: The product manager identifies timeliness as an important quality attribute, which is communicated to the data scientist. A new quality attribute is created for timeliness. The data scientist includes reference to a data processing tool, which is capable of evaluating the quality attribute (timeliness)

Steps 5–6: A quality metric is created for timeliness, which is specified as JEngineRules [43]. The rules specify how tweet metadata is used for calculating a value for timeliness

Steps 7–8: The product manager creates a filtering policy, and specifies an acceptable quality level for tweets regarding timeliness

Steps 9–10: An organisational quality policy is created by the product manager. The policy defines applicability of timeliness for the data source (Twitter). Also, an associated filtering policy is referred to. Optionally, an associated identifier of metadata may be included to the policy

Sequence view: searching of data quality information for supporting decision making

Figure 7 presents how data quality information can be retrieved from the metadata store:

Steps 1–3: End user/application creates a search filtering policy. Quality attributes and associated ranges are provided. It is also indicated, if social media data should be included to the response. Optionally, metadata identifier and time range may be provided. The policy is saved into the quality rules store

Step 4: The end user sends a HTTP GET to MetadataSearchEngine, and provides identifier of the search filtering policy

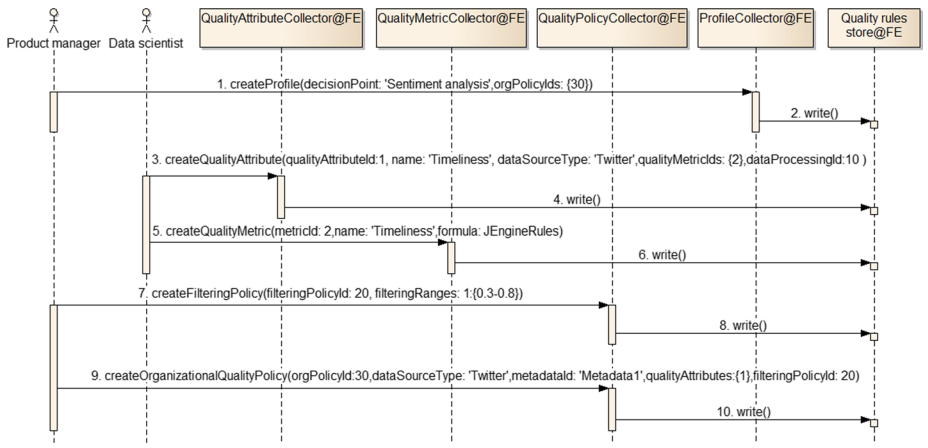

Fig. 6 Creation of quality rules (profile, quality metrics, quality attribute, and organisational quality policy)

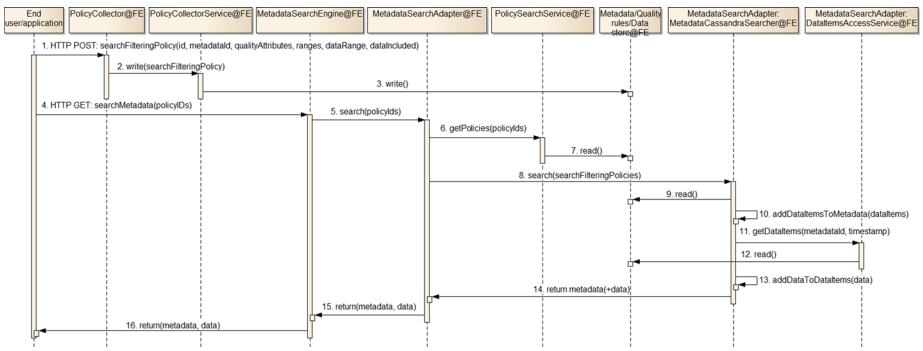

Fig. 7 Searching of data quality information from the metadata store

Steps 5–7: Metadata is searched from the metadata store based on the input parameters

Steps 8–10: If matching metadata is found, related metadata for data items (Fig. 15) is read from the metadata store

Steps 11–13: Social media data is read from the Data store, if inclusion of data has been specified in the search filtering policy (Fig. 14)

Steps 14–16: Metadata and related data are returned to the end user/application for supporting decision making

Component view: Back-end node

Figure 8 provides a detailed component view to data quality evaluation in the Back-end node. The role of QualityEvaluator is to deploy data quality evaluation tasks based on requests received from the MetadataQualityManagement-component (Fig. 4). NetworkService provides a REST API by using Jetty as a web server. The purpose of QualityEvaluationService is to deploy new Twitter data analysis tasks in a Spark Streaming cluster based on the received requests. TwitterAnalysis-component encapsulates a Spark Streaming-task, which will be started for each data quality evaluation task.

Responsibilities of sub-components in TwitterAnalysis are described in Table 2.

Sequence view: data quality evaluation (sequence 1)

Figures 9 and 10 present how metadata is created for Twitter-based data sets, and how the data quality evaluation process is started in the Back-end node:

Steps 1–3: The end user (of Invenco's product) searches for data from Twitter APIs, which is extracted, and saved into a temporary data store in the big data system

Step 4: The data extractor (of Invenco's product) transmits metadata about the stored data set to MetadataCollectionEngine by using the REST API (Fig. 5)

Steps 5–6: The metadata is received and validated

Steps 7–9: The metadata is compared against matching profiles and organisational quality policies. Data source type and identifier of metadata is matched against organisational quality policies

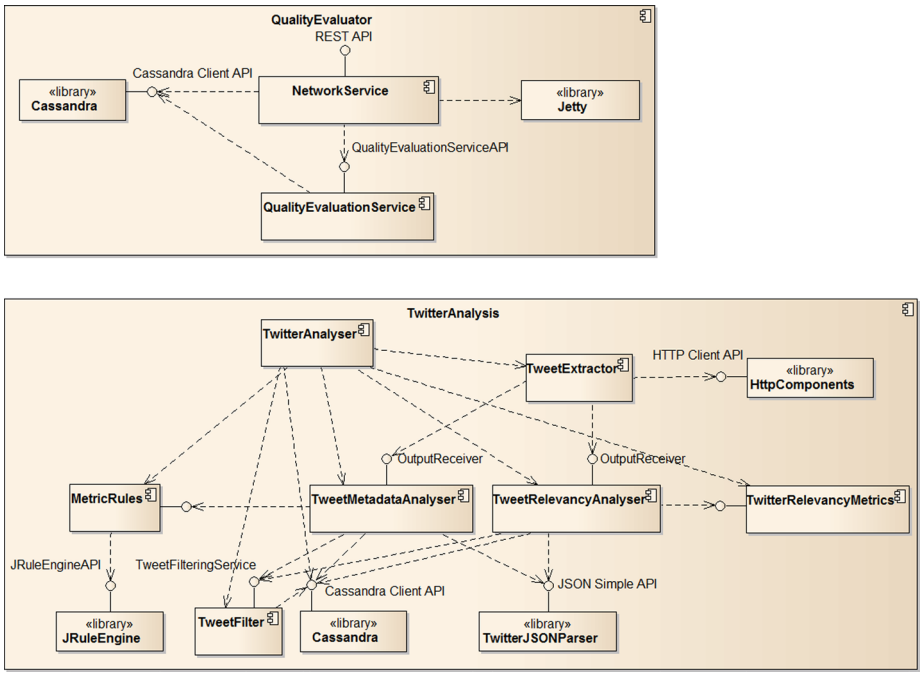

Fig. 8 Component view of QualityEvaluator and TwitterAnalysis, which are executed in the Back-end node

Table 2 Responsibilities of sub-components in TwitterAnalysis

Component	Responsibility
TwitterAnalyser	Coordination of data quality evaluation process for tweets
MetricRules	Rules in JRuleEngine-format for evaluation of timeliness and popularity (quality attributes)
TweetFilter	Filtering of data based on a filtering policy
TweetMetadataAnalyser	Analysis of tweet metadata for evaluation of timeliness and popularity
TweetRelevancyAnalyser	Analysis of tweet content for evaluation of relevancy
TweetExtractor	Extraction of tweets from an external data source
TwitterRelevancyMetrics	Metrics for evaluation of relevancy

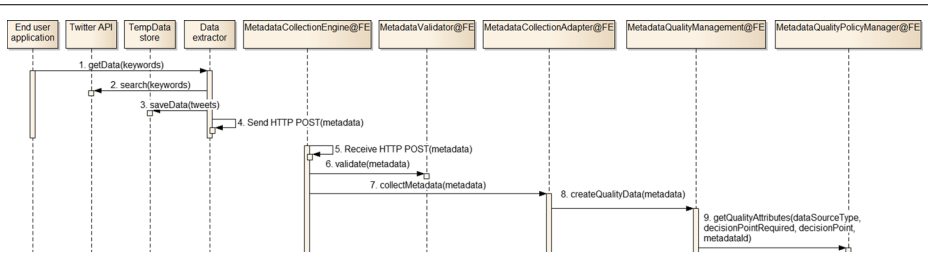

Fig. 9 Data quality evaluation (sequence 1). Front-end node = FE, Back-end node = BE

Steps 10–11: If an applicable policy is found, metadata is accepted for further evaluation, and stored into metadata store

Steps 12–13: Identifier of metadata, quality attributes, and filtering policy are transmitted to NetworkService (in a HTTP POST), which is executed in the Back-end node (Fig. 8)

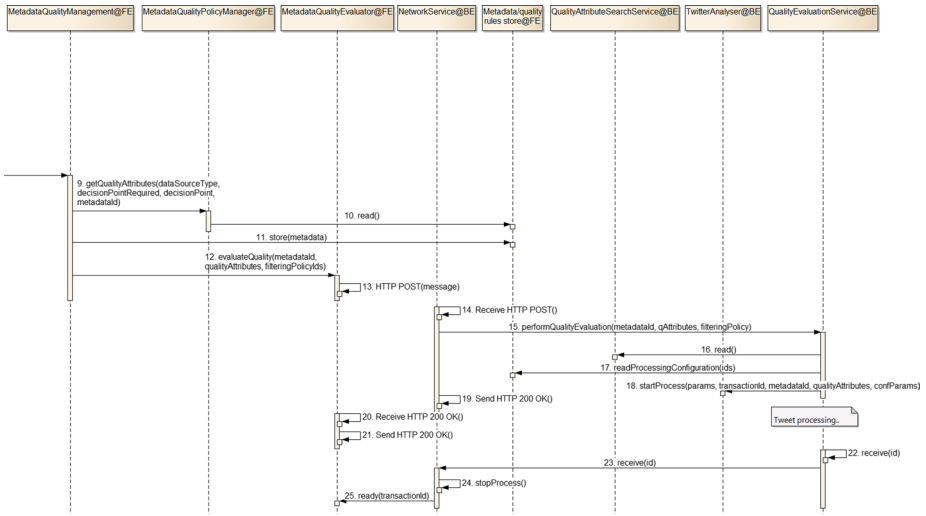

Fig. 10 Data quality evaluation (sequence 1). Front-end node = FE, Back-end node = BE

Steps 14–17: HTTP POST is received. Processing configurations (Fig. 14) are read from the quality rules store. The information contains the data quality evaluation task to be started, and configuration/processing parameters to be used for starting of the analysis process

Step 18: TwitterAnalyser-process is started in Spark Streaming context (Fig. 8). Identifier of transaction and Java process is saved

Steps 19–21: HTTP 150 OK response will be transmitted to MetadataQualityEvaluator (at the Front-end node). HTTP 150 OK will be transmitted to the data extractor in the big data system

Steps 22–25: Transaction identifier is matched against the Java process identifier. When tweet processing has been completed, the Spark Streaming process (based on the Java process identifier) will be stopped

Data quality evaluation (sequence 2)

Figure 11 illustrates data quality evaluation in the Back-end node:

Step 1: TwitterAnalyser is started in the Back-end node

Steps 2–5: Metadata and quality attributes are read from metadata/quality rules store based on input received from the Front-end node. Processing configuration is retrieved based on the reference in the quality attribute

Step 6: Data quality analysis processing to be executed is filtered based on supported quality processing information (Fig. 14). Quality attributes are matched with supported processing based on data source type. Also, target of quality attribute must match with output_data_type-parameter of supported processing (Fig. 14)

Steps 7–9: Quality metrics are read from the quality rules store. Metrics are decoded with appropriate decoders. Format-field of the metric (Fig. 14) indicates how the metric should be decoded

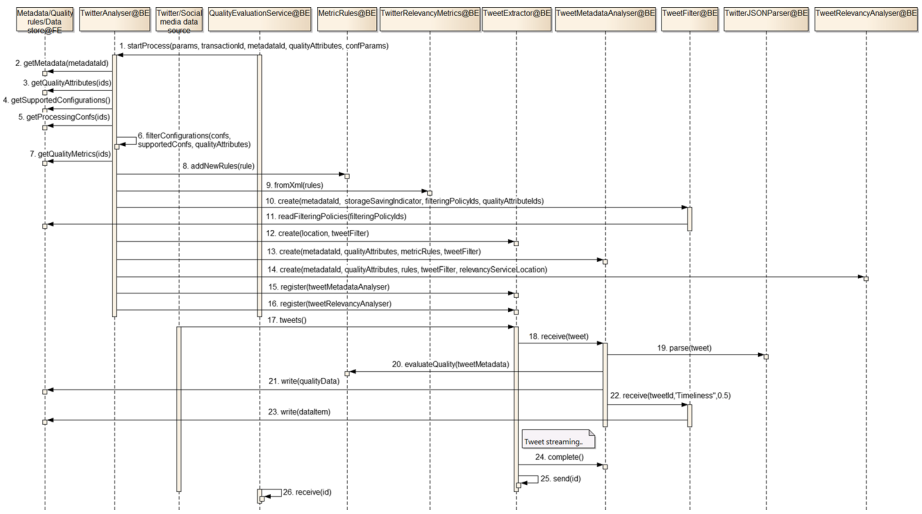

Fig. 11 Data quality evaluation in the Back-end node (sequence 2)

Steps 10–11: A filter is created for tweets. Filtering policy is read from the quality rules store

Steps 12–14: Tweet extractor, metadata analyser, and relevancy analyser are created

Steps 15–16: Metadata and relevancy analysers are registered to the tweet extractor for processing of tweets

Step 17–18: Tweet extractor downloads tweets based on the DataSetLocation-parameter of metadata (Fig. 15). Each retrieved tweet is forwarded for further processing

Steps 19–21: Metadata analyser parses tweet metadata from the tweet. Timeliness and popularity are evaluated based on the associated metric, and tweet metadata. Values of the quality attributes are saved into the metadata store

Steps 22–23: Analysed tweets and quality attributes are filtered. If all quality attributes associated with a tweet satisfy a filtering policy, the tweet will be saved into the data store. Otherwise, the tweet will be discarded

Steps 24–26: When all tweets have been streamed, tweet extractor informs upper layer (QualityEvaluationService in the Back-end node) about completion (by providing a transaction identifier). The indication will be utilized for stopping of the data evaluation process (step 24 in Fig. 10)

Evaluation

Metadata management architecture

Earlier work [11] presented initial validation of the metadata management layer for big data systems (Fig. 1). Based on incremental design and implementation of the prototype system (step 3 of the research method [40]), the metadata management layer has been extended (step 4) and evaluated (step 6) in this work. In the following, elements of the improved architecture (Fig. 12) are presented.

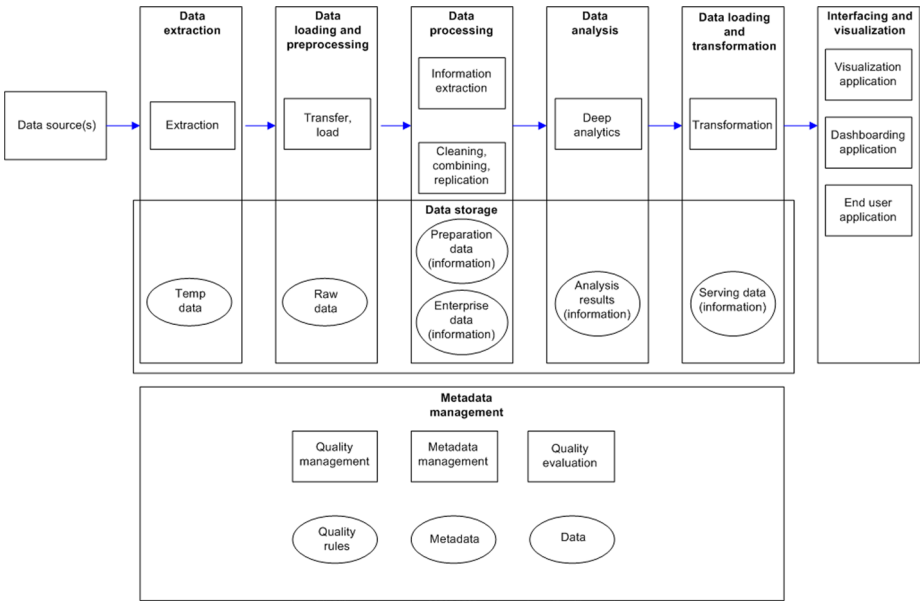

Fig. 12 Improved metadata management architecture for big data systems. A *rectangle* illustrates functionality, and an *ellipsis* describes a data store. Extended based on Immonen's work [11]

Quality rules store include quality attributes, quality policies, quality metrics, and profile (Fig. 14), which are utilised for managing quality of data sets. Metadata contains information about data sets in different dimensions including quality aspects (Fig. 15). Data store contains social media data (tweets in the prototype system), which has been evaluated.

Metadata management refers to creation of metadata, and providing access to it. MetadataCollectionEngine and MetadataSearchEngine components of the Front-end node (Fig. 5) encapsulated functionality of metadata management. Quality management refers to managing quality aspects of data sets with user-defined quality rules. In the prototype system MetadataQualityManagement, MetadataQualityEvaluator, and MetadataQualityPolicyManager of the Front-end node implemented quality management functionality (Fig. 5). Quality evaluation refers to analysing quality of social media data sets based on quality metrics, which have been selected to a context based on quality rules. In the prototype system quality evaluation was comprised of QualityEvaluator and TwitterAnalysis components of the Back-end node (Fig. 8), and Word2Vec-service of the relevancy-node (Fig. 4).

In the big data system, social media data was extracted from Twitter (data source), and stored. The raw data was transferred to the data quality management tool for quality evaluation. Sentiment analysis (deep analytics) was performed, and results were saved (analysis results). Then, sentiment analysis results are going to be filtered (transformation) based on the data quality information. Finally, the filtered information will be visualised to the end users in an application.

Implementation

In the following technology selections for implementation are evaluated in terms of the data quality management architecture:

Data management

Cassandra was used for storing of metadata, quality rules, and data (see Additional file 1). However, each of the conceptual stores could also have been implemented with a different database technology. Data was modelled based on expected queries to be served, which is one of the principles of data modelling for Cassandra [44]. From the end user's point of view, quality policies, quality attributes, and quality metrics have to be defined for data quality management (Fig. 2). The information was stored into the database (see Additional file 1). The result of tweet quality evaluation was stored into metadata store (metadata_dataitems in Additional file 1), and social media data into data store (data_store in Additional file 1). An example on data quality management over REST API has been provided in the Additional file 3. Performance of the data management has been evaluated in the following section.

External and internal APIs

The external APIs were implemented based on REST-based communication paradigm. Interface messages were defined in extensible markup language (XML) format, and implemented with Jersey [45]. REST API of the QualityEvaluator in the Back-end node (Fig. 8) was implemented as an embedded Jetty server [46] (Jersey could also have been used). Word2Vec service's API (Fig. 8) in the relevancy server was implemented as a TCP server, which accepted words of tweets as input, and returned calculated relevancy value as a response.

Social media data extraction for quality evaluation

Social media data was extracted from Twitter APIs, and stored in raw format into a temporary data store in the big data system [12] by Invenco. Stored data sets containing tweets were provided for quality management with HTTP. Location [uniform resource locator (URL)] of a raw tweet data set was included into metadata (Fig. 15). The raw data was transferred with chunked HTTP encoding to data quality evaluation (step 17 in Fig. 11). Apache HTTP components [47] were utilized for the transmission in Tweet-Extractor (Fig. 11). Javascript object notation (JSON) Simple parser [48] was used for extracting of tweet content and related metadata from the raw data (step 19 in Fig. 11).

Quality metrics for quality evaluation

Different metrics were needed for evaluation of the quality attributes (see Additional file 2). Timeliness was evaluated based on timestamp (created_at [49]) of a tweet. Retweet count (retweet_count [49]), number of friends (friends_count [49]), and number of followers (followers_count [49]) were utilised for evaluating popularity of a tweet (see Additional file 3 for an example). The metric for evaluation of quality attributes based on the tweet metadata was implemented as a file, which contained evaluation rules. Java Rule Engine API [43] was utilised for encoding/decoding of the evaluation rules in XML format (steps 8–9 in Fig. 11). Particularly, the XML file contained weighted parameters of tweet metadata for quality evaluation.

Relevancy evaluation relied on Google's Word2Vec-implementation [29]. DeepLearning4 J-library [50] provided a Java API to Google's pre-trained word vector model, which is based on Google News data set. The implementation was executed on a separate

Relevancy node (Fig. 4), due to large memory consumption of the model (zipped model file ~1.6 GB). Word2Vec algorithm was utilised for calculation of word cosine distance between words of a tweet, and context words. A metric file (see Additional file 2) was utilized for adjusting a threshold for word cosine distance for indicating, when a tweet word is relevant. The metric file also contained context words, which were compared to tweet words. Another metric (in JRulesEngine format) was used for calculating the final relevancy score based on the quantity of words, which were found to be relevant in a tweet.

Performance evaluation

Performance of data quality evaluation was studied. In the experiments 176,478 tweets (~798 MB) were transmitted for quality evaluation. The tweets were extracted from the public Twitter API [49]. The average size of a tweet was ~4.6 KB. The experiments were executed within the Eucalyptus cloud computing environment (Fig. 4), where Front-end node had two vCPUs, and 4 GB RAM. Back-end node and relevancy nodes had six vCPUs and 40 GB RAM. Two vCPUs and 4 GB RAM was allocated for Twitter analysis-component (Fig. 8) within the Spark streaming cluster. An additional node (had six vCPUs and 40 GB RAM) was used for simulating a Twitter data source, which utilized Netcat for streaming of tweets using a TCP connection. TCP was used instead of HTTP to minimise protocol overhead in the experiments. The tests were performed five times, and average processing rate of tweets is reported.

In the experiments, quality attributes, Cassandra batching, and saving of social media data were controlled. Each Cassandra batch contained ten quality attribute updates, or five tweet writes. The results (Fig. 13) indicated that relevancy evaluation is slower, when compared to evaluation of timeliness/popularity. Secondly, batching of database writing/updating requests [51] improved performance significantly, when compared to the baseline (no batching). Finally, if social media data (tweets) do not need to be stored, performance can be improved (as expected).

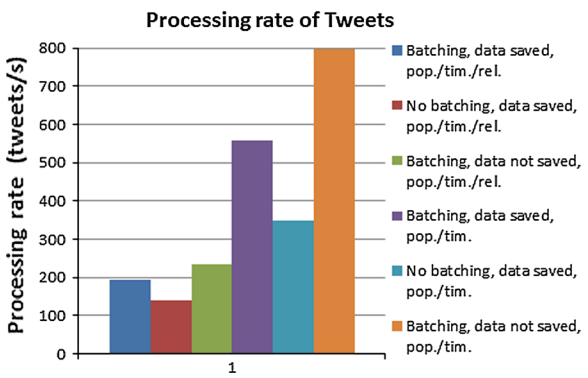

Fig. 13 Performance of data quality evaluation. Different configuration combinations were tested: Cassandra batching/no batching, tweet saving/no saving, and evaluated quality attributes [2 (popularity/timeliness) or 3 (popularity/timeliness/relevancy)]

Discussion

Main lessons learnt

Quality data was stored separately for each tweet (metadata_dataitems in Additional file 1). Therefore, multiple updates/writes to Cassandra were batched, which is known to increase performance, when applied to data residing in the same cluster [51]. Batching increased performance of data quality evaluation, as expected (Fig. 13). Alternatively, quality attributes could have been modelled differently. For example, values of quality attributes could have been stored inside of a Cassandra collection, which provides support for storing up to 2 billion entries [51].

Relevancy evaluation was based on word cosine distance-measure, which is used by the Word2Vec-implementation. Word2Vec model had to be loaded initially into memory, which takes ~4.5 min on a cloud node. Due to the slowness of loading, and large memory consumption, relevancy evaluation was performed in a separate Eucalyptus instance. The experiments indicated that relevancy evaluation was slower, when compared to evaluation of timeliness/popularity (Fig. 13). Word2Vec service in the Relevancy node (Fig. 4) read words of a tweet one line at a time from a single TCP socket. Performance of relevancy evaluation may be improved, if multiple sockets would be utilized for the communication.

The quality metrics were associated with the underlying implementation of quality evaluation. For example, metrics were used for calculating a value for popularity based on specified tweet metadata (friend's count, followers count etc.). However, if a different set of tweet metadata would need to be utilized for quality evaluation, quality metrics and quality evaluation implementation would need to be modified accordingly. An alternative approach would be to utilize ontology-based approach to definition of quality metrics and evaluation, where updated representation of concepts in the metrics would be automatically updated in the system [52].

Several configuration [e.g. location of java archive (JAR)-files] and run-time parameters (e.g. CPU cores, memory, location of Spark master) have to be provided to Spark Streaming in order to control the quality evaluation process. Therefore, new data structures (quality_data_processing and supported_quality_data_processing in Additional file 1) had to be designed to the quality management architecture (Fig. 14). The role of the data structures is to enable run-time configuration of quality evaluation process. Additionally, each data processing tool has a unique identifier, which has to be provided to the end user of the data quality management tool. For example, when quality attributes are created by the data scientist of an organisation (Fig. 6), each quality attribute should refer to the associated data processing tool. In the prototype system, a reference was made to a single processing tool (identifier), because one processing implementation was capable of performing quality evaluation for all of the evaluated quality attributes.

Comparison to literature

This work can be compared to existing framework proposals for quality management in social media (Table 3). Social Haystack assesses quality of citizen-generated content during emergencies [18]. Up-to-datedness/dissemination corresponds to timeliness/popularity quality attributes of our work. Social-QAS has been proposed for quality assessment of social media data [16]. However, neither approach can be utilised for

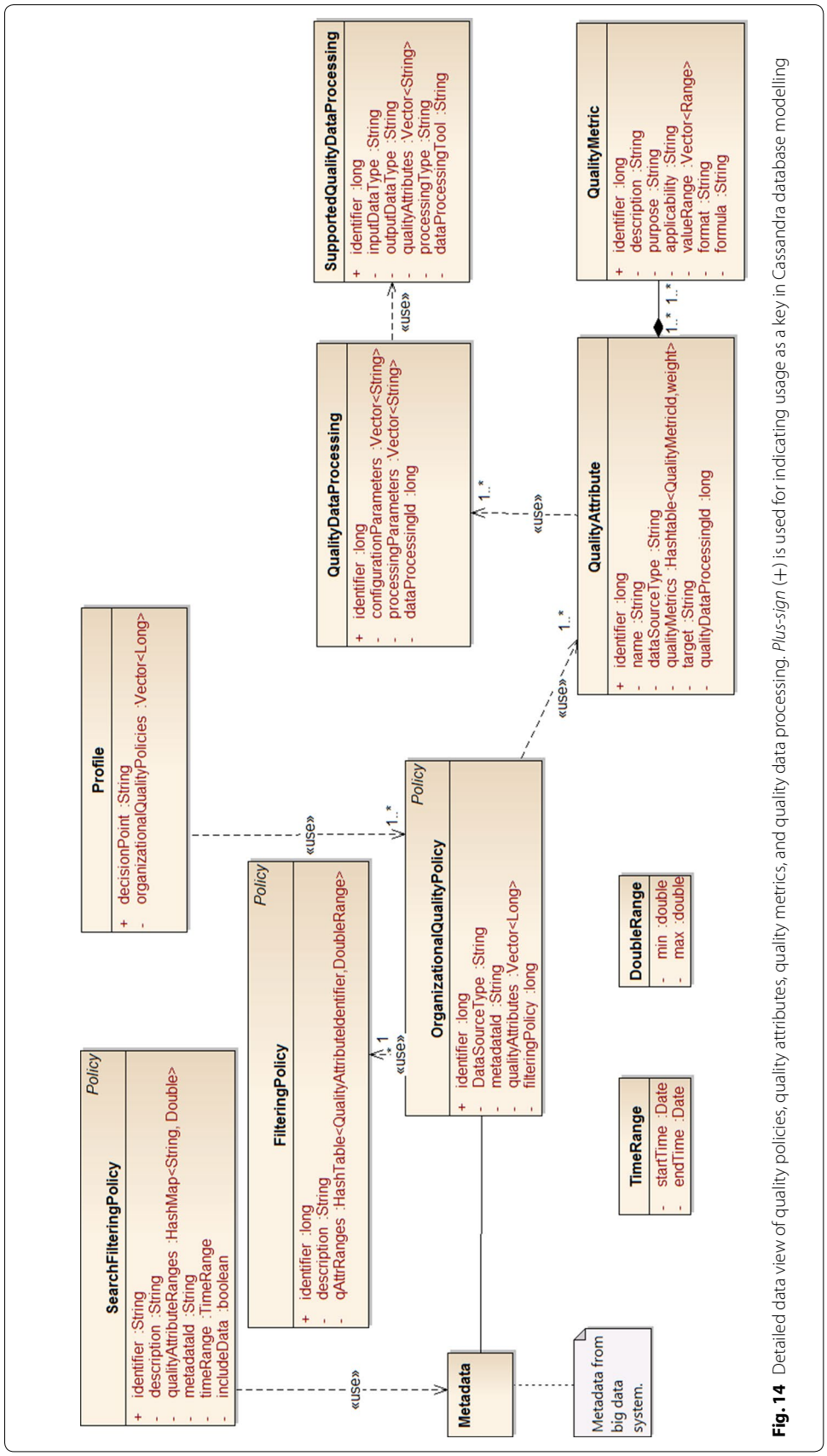

Fig. 14 Detailed data view of quality policies, quality attributes, quality metrics, and quality data processing. *Plus-sign* (+) is used for indicating usage as a key in Cassandra database modelling

Table 3 Comparison to approaches in literature

Paper	Domain (data set)	Quality evaluation	Tools/technologies	Quality level spec.	Quality metric spec.	Quality policy spec.
Ludwig [18]	Social media (Twitter, Facebook, Maps)	Weighted score: link, credibility, up-to-datedness, dissemination, quality of coordinates	OpenSocial format, MSSQL database	Weighting of quality attributes	–	–
Reuter [16]	Social media (Facebook)	Based on metadata, content, message, classification, and scientific methods	Stanford NER, Classifier4J, Open Thesaurus,Gisgraphy Geocoder	Weighting of parameters	–	–
Taleb [9]	Medical (EEG data)	Accuracy, consistency	Hadoop MapReduce	XML file: targeted data quality	XML file: data cleansing algorithm	Data quality profile
Serhani [10]	Medical (SHRS)	Completeness, consistency, accuracy	Talend and Trifacta Wrangler. ML Vagrant	–	–	–
Immonen [11]	Social media (Twitter)	Timeliness	Cassandra	–	–	Static
Our approach	Social media (Twitter)	Timeliness, relevancy, popularity	Spark, Cassandra, Word2Vec	Ranges specified in SearchFilteringPolicy	JEngineRules/XML file	Dynamic quality policies

specifying quality metrics/policies from the end user perspective. Taleb has proposed quality evaluation to the pre-processing phase of a big data system [9]. The main differences to our work are focus on medical data, and focusing on data quality with a data cleansing algorithm. Serhani proposed a hybrid approach for quality evaluation, covering pre-processing, processing and analysis phases of a big data pipeline [10]. The main differences to our work are focus on medical data, offline data processing/analysis with data analytics tools. Also, a different set of quality attributes (accuracy, completeness, and consistency) were evaluated, when compared to this work.

Immonen [11] presented mainly quality evaluation of social media data based on quality policies. Also, an initial validation of RA for quality management of social media data was presented [11]. The RA (Fig. 1) has been extended in this work with implementation of quality evaluation for tweets, evaluation of new quality attributes based on the developed metrics, and adaptable quality rules to a database. The main contribution of this work is presentation of the metadata management layer for big data systems (Fig. 12), which has been validated with an implementation. Additionally, performance of the implementation has been evaluated from data quality evaluation point of view.

Future work

Extension of the RA based on other implementation approaches

Social media data may need to be cleaned or filtered before analysing its quality (Serhani). We didn't focus on the pre-processing phase of the big data pipeline (Fig. 12). When social media data is cleaned/filtered, quality of data may improve, which should be saved into the associated metadata. Social media data may also be processed with offline data processing tools (Sehrani). We aimed at automating the process of data quality evaluation instead of manual processing. If social media data sets would be processed externally by the end user, metadata/quality management part of the architecture would need to be extended.

Validation of the RA with new business cases

The revised RA [11] has been developed originally based on seven published implementation architectures of big data systems [12]. In this work, the RA has been extended, and validated based on a realised architecture in a business case. In order to validate suitability of the RA for various business cases in the context of social media data, the RA should be experimented with several implementation architectures. This work can be considered as a step towards that goal.

Performance of quality attributes and development of quality evaluation algorithms based on machine learning

Quality of social media data was evaluated by utilizing either simple (timeliness, popularity) or trained models (relevancy). In order to evaluate performance (e.g. precision or recall) of the quality attributes, a ground truth based on human observations would be needed for comparison. The ground truth could be established by crowd-sourcing annotations to the evaluated social media data (e.g. with Amazon Mechanical Turk) [6, 7]. Also, if credibility of tweets would need to be evaluated, a model may have to be developed based on supervised machine learning, where annotated data would be utilised for training of the model (e.g. [6, 7, 36]).

Performance optimisation in quality evaluation

Spark streaming was used in the implementation to study feasibility of the technology for quality evaluation. Previously, sentiment analysis of tweets has been simulated in a similar environment [53]. Spark streaming processed lower number of tweets (per second) with one node in this study (~150–800 tweets/s vs. ~1000 tweets/s). Performance may be improved with optimisations related to storage of quality data and quality attribute evaluation (relevancy). Currently, tweets are analysed sequentially in a Spark streaming cluster. Parallel processing of tweets may need to be implemented, in order to achieve better performance.

Streaming support for quality evaluation

In the prototype no feedback is provided to the end user regarding the processing of tweets. As an alternative to REST-based communication pattern, a streaming interface could be developed for providing processing indications to support utilisation of quality information in real time.

Updating of data quality information

Quality evaluation is performed each time metadata is received from the end user. Social media data is saved permanently, if saving has been specified in the data_storage_indicator-parameters of metadata (Fig. 15), and processed data has not been filtered out. Updating of data quality information may be needed, when quality metrics change. In the future HTTP PUT could be utilised for initiating an update procedure for data, which has been stored earlier.

Filtering and visualisation of tweets for facilitating decision making

Currently, quality data has been created based on Twitter data sets, and filtered for supporting sentiment analysis of tweets. Subsequently, quality information of individual tweets will be used for filtering in sentiment analysis. Especially, it should be determined how quality information will be visualised for end users in order to support decision making.

Multiple social media data sources and decision points

When an organisational actor has a need for managing quality of social media data, quality rules are specified. In the prototype a single organisational policy and profile were used for quality management of tweets. Multiple organisational policies would be needed, when different social media data sources are utilised in a business case. Similarly, multiple policies would be needed for different decision points. Especially, it should be considered how data from one or more social media data source(s) can serve decision making in different business contexts.

Evaluation of new quality attributes. New functionality has to be developed for evaluation of additional quality attributes. The added functionality should be updated to the processing configurations (Fig. 14), and communicated to the end users of the tool.

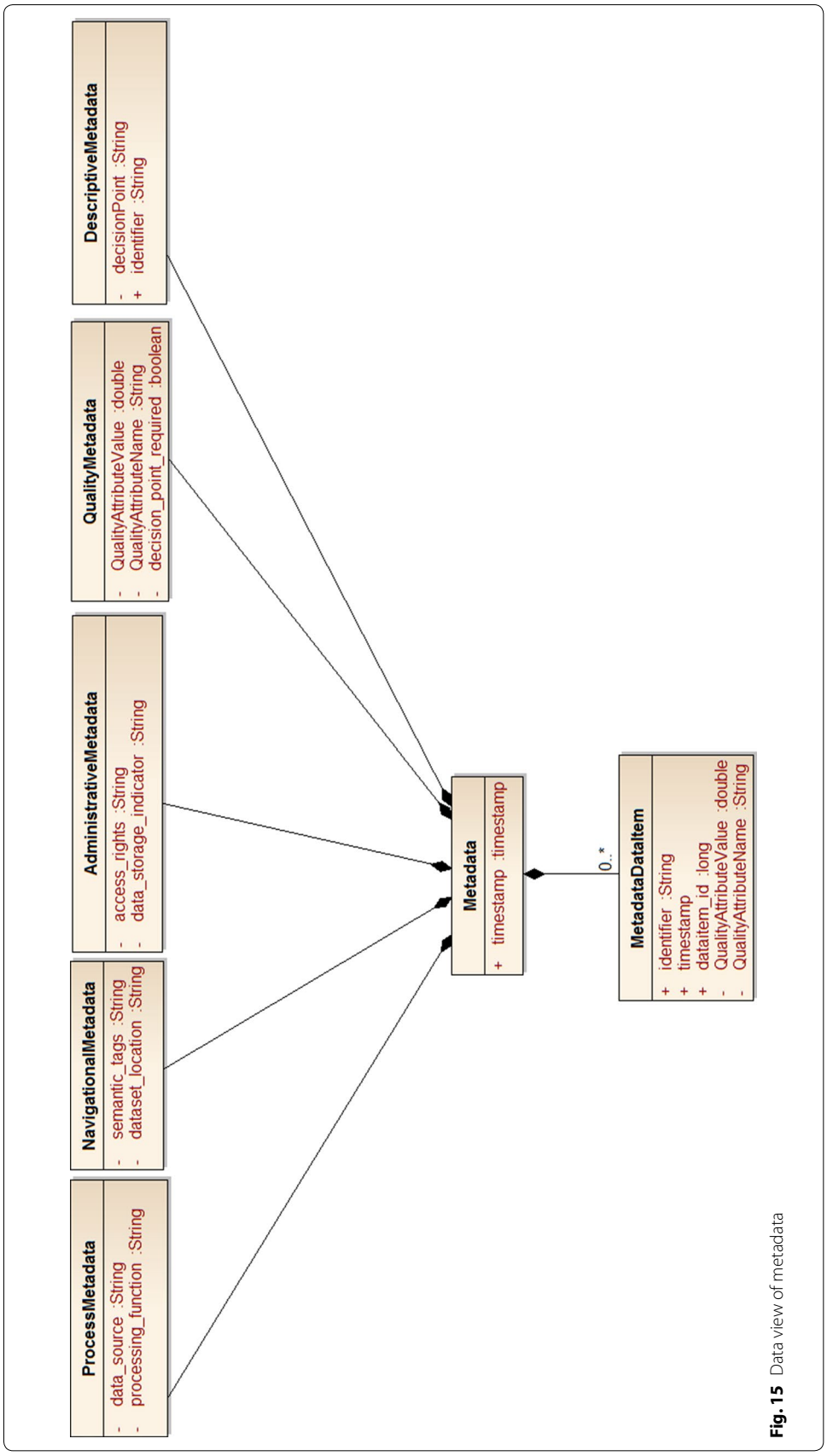

Fig. 15 Data view of metadata

Conclusion

This research was conducted based on the needs of a company for managing quality in social media data. The contribution of the paper is (an extended) RA design of metadata management layer for big data systems, which focuses on data quality management. The RA was implemented for ensuring empirical validity of the design. Feasibility of the developed data quality management-tool was validated in a business context, in which the company utilised data quality information for sentiment analysis of tweets. Quality of tweets was evaluated in terms of timeliness, popularity, and relevancy. The quality information was used for filtering of tweets to facilitate creation of higher quality insights with the company's product.

The research question was "How is quality management architecture constructed for evaluating the quality of social media data?" The data quality management architecture is comprised of a metadata management layer in the RA for big data systems. The metadata management layer consists of quality rules, metadata, and data (data stores). Quality rules provide means for the organization of the company to manage quality of social media data sets for decision making purposes. The main functional elements of the metadata management layer include metadata management, quality management, and quality evaluation. In the prototype system metadata management enabled creation of and access to metadata related to tweets. Quality management was responsible for managing quality aspects of metadata based on user-defined quality rules. Quality evaluation of tweets was performed in a Spark streaming cluster, which indicated that 150–800 tweets/s can be processed with two cloud nodes depending on the configuration.

Additional files

Additional file 1. The file contains modelling of metadata store, quality rules, and data store for Cassandra.
Additional file 2. The file contains quality metrics for evaluation of timeliness, relevancy, and popularity.
Additional file 3. An example for data quality management of tweets. Example code can be compared with the corresponding data views (Figs. 14, 15).

Abbreviations

ERP: enterprise resource planning; JAR: java archive; JSON: javascript object notation; NLP: natural language processing; RA: reference architecture; REST: representational state transfer; SCM: supply chain management; UML: unified modelling language; URL: uniform resource locator; XML: extensible markup language.

Authors' contributions

PP wrote the article. He also designed and implemented the presented architecture, and executed the experiments. JJ contributed to the design of the use case view (in "Architecture design" chapter). Both authors read and approved the final manuscript.

Author details

[1] VTT Technical Research Centre of Finland, Oulu, Finland. [2] Invenco, Oulu, Finland.

Acknowledgements

The author acknowledges Esa Ronkainen and Juha Jokitulppo (from Invenco) for providing business context and related feedback for this research.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The dataset supporting the conclusions of this article is included within the article (and its additional files).

Funding

This research has been carried out in Digile Need for Speed program, and it has been partially funded by Tekes (the Finnish Funding Agency for Technology and Innovation).

Appendix

Detailed data view of quality rules

Figure 14 provides a detailed data view of quality rules. In the following, each concept is described in detail (data modelling with Cassandra in Additional file 1). Profile refers to a decision point within the organisation. It may be associated with multiple organisational quality policies. An organisational quality policy defines an acceptable data source type, and applicable quality attributes [11]. Also, it refers to a filtering policy, which defines acceptable value ranges for specified quality attributes. Metadata identifier associates organizational quality policy with metadata.

A quality attribute specifies source of social media data (e.g. Twitter), and target (e.g. tweets) the attribute refers to. A quality attribute may refer to one or more quality metrics, which are utilized for evaluation of the attribute [11]. Each quality metric may contain a formula/model, which can be encoded with a specified format.

Quality data processing defines configuration and processing parameters, which are utilized in the evaluation process of quality attributes. Examples of configuration parameters include location of binary or configuration files, and environmental execution parameters (e.g. number of CPU cores, size of memory). Processing parameters refer to input data, which should be provided to data processing executables (e.g. Spark streaming). Supported quality data processing defines processing, which can be performed with a specific data processing tool. Especially, it can be specified what kind of quality attributes for a data source can be evaluated with a specific data processing tool.

Finally, a search filtering policy may be specified for querying of quality data. The end user can specify a range for each quality attribute, identifier of searched metadata, and a time range. Also, it can be specified, if social media data should be provided (includeData).

Data view on metadata

Figure 15 presents a data view of metadata (metadata in Additional file 1). Metadata contains navigational, process, descriptive, administrative, and quality aspects of collected data sets [11]. A few parameters have been described for each aspect of metadata. In the prototype system, dataset_location (NavigationalMetadata) contained the location for downloading of tweets. data_storage_indicator (AdministrativeMetadata) was used for specifying, if associated data items were required to be stored permanently. decision_point_required (DescriptiveMetadata) indicated if a matching profile was needed for processing of associated metadata. The results of quality evaluation related to individual tweets were stored to MetadataDataItem. If quality attributes would have been related to an entire data set of tweets, the results of quality evaluation would have been stored to the metadata structure (QualityMetadata).

Received: 13 December 2016 Accepted: 15 March 2017
Published online: 23 March 2017

References

1. Cui W. How to use the social media data in assisting restaurant recommendation. LCNS. 2016;9645:134–41.

2. Asur S, Huberman BA. Predicting the future with social media. In: Paper presented at the IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, Toronto, Canada, 31 August to 3 September, 2010.
3. He W. Gaining competitive intelligence from social media data. Ind Manag Data. 2015;115(9):1622–36.
4. Chen C. 6 million spam tweets: a large ground truth for timely twitter spam detection. In: Paper presented at the communication and information systems security symposium, London, United Kingdom, 8–12 June, 2015.
5. Reuter C, Spielhofer T. Towards social resilience: a quantitative and qualitative survey on citizens' perception of social media in emergencies in Europe. Technol Forecast Soc Change. 2016. doi:10.1016/j.techfore.2016.07.038.
6. Castillo C, Mendoza M, Poblete B. Information credibility on twitter. In: Paper presented at the 20th international world wide web conference, Hyderabad, India, 28 March to 1 April, 2011.
7. Castillo C, Mendoza M, Poblete B. Predicting information credibility in time-sensitive social media. Internet Res. 2013;23(5):560–88.
8. Momeni E, Haslhofer B, Tao K, Houben G. Sifting useful comments from Flickr Commons and YouTube. Int J Digit Libr. 2015;16(2):161–79.
9. Taleb I, Dssouli R, Serhani MA. Big data pre-processing: a quality framework. In: Paper presented at the IEEE international congress on big data. New York, USA, 27 June to 2 July, 2015.
10. Serhani MA, El Kassabi HT, Taleb I, Nujum A. An hybrid approach to quality evaluation across big data value chain. In: Paper presented at the IEEE international congress on big data, San Francisco, USA, 27 June to 02 July, 2016.
11. Immonen A, Pääkkönen P, Ovaska E. Evaluating the quality of social media data in big data architecture. IEEE Access. 2015;3:2028–43.
12. Pääkkönen P, Pakkala D. Reference architecture and classification of technologies, products, and services for big data systems. Big Data Res. 2016;2(4):166–86.
13. National Information Standards Organization. Understanding metadata. 2004. http://www.niso.org/publications/press/UnderstandingMetadata.pdf. Accessed 30 Jan 2017.
14. Wang RY, Strong DM. Beyond accuracy: what data quality means to data consumers. J Manag Inform Syst. 1996;12(4):5–33.
15. W3C. Web services policy 1.5—framework (W3C recommendation). 2007. https://www.w3.org/TR/ws-policy/#Policy_Model. Accessed 05 Dec 2016.
16. Reuter C, Ludwik T, Ritzkatis M, Pipek V. Social-QAS: tailorable quality assessment service for social media content. In: Paper presented at the 5th international symposium on end user development, Madrid, Spain, 26–29 May, 2015.
17. Reuter C, Ludwik T, Kaufhold M, Pipek V. XHELP: Design of a cross-platform social-media application to support volunteer moderators in disasters. In: Paper presented at the CHI crossings, Seoul, Korea, 18–23 April, 2015.
18. Ludwig T, Reuter C, Pipek V. Social haystack: dynamic quality assessment of citizen-generated content during emergencies. ACM Trans Comput Hum Interact. 2015;22(4):17.
19. Chae B. Insights from hashtag# supplychain and Twitter analytics: considering Twitter and Twitter data for supply chain practice and research. Int J Prod Econ. 2015;165:247–59.
20. Shankararaman V, Lum EK. Integration of social media technologies with ERP: a prototype implementation. In: Paper presented at the 19th Americas conference on information systems, Chicago, Illinois, USA, 15–17 August, 2013.
21. Gao J, Xie C, Tao C. Big data validation and quality assurance—issues, challenges, and needs. In: Paper presented at the IEEE 2016 symposium on service-oriented system engineering, Oxford, United Kingdom, 29 March to 2 April, 2016.
22. Moges H, Vlasselaer VV, Lemahieu W, Baesens B. Determining the use of data quality metadata (DQM) for decision making purposes and its impact for decision outcomes—an exploratory study. Decis Support Syst. 2016;83:32–46.
23. Bello-Orgaz G, Jung JJ, Camacho D. Social big data: recent achievements and new challenges. Inf Fusion. 2016;28:45–59.
24. Bontcheva K. TwitIE: an open-source information extraction pipeline for microblog text. In: Paper presented at the recent advances in natural language processing, Hissar, Bulgaria, 7–13 September, 2013.
25. Derczynski L. Analysis of named entity recognition and linking for tweets. Inf Process Manag. 2015;51:32–49.
26. Ritter A, Clark S, Etzioni M, Etzioni O. Named entity recognition in tweets: an experimental study. In: Paper presented at the conference on empirical methods in natural language processing, Edinburgh, Scotland, UK, 27–31 July, 2011.
27. Zangerle E, Gassler W, Specht G. Recommending #-tags in Twitter. In: Paper presented at the workshop on semantic adaptive social web, Girona, Spain, 15 July, 2011.
28. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Paper presented at the international conference on learning representations, Scottsdale, Arizona, USA, 2–4 May, 2013.
29. Google Code Archive. Word2Vec. https://code.google.com/archive/p/word2vec/. Accessed 08 Nov 2016.
30. Tomar A. Towards twitter hashtag recommendations using distributed word representations and a deep feed forward neural network. In: Paper presented at the international conference on advances in computing, communications, and informatics, Delhi, India, 24–27 September, 2014.
31. Batrinca B, Treleaven PC. Social media analytics: a survey of techniques, tools and platforms. AI Soc. 2015;30(1):89–116.
32. Rosa KD. Topical clustering of tweets. In: Paper presented at the social web search and mining, Beijing, China, July 28, 2010.
33. Ferrara E. Clustering memes in social media. In: Paper presented at the IEEE/ACM international conference on advances in social networks analysis and mining, Niagara, Ontario, Canada, 25–29 August, 2013.
34. Godin F. Using topic models for twitter hashtag recommendation. In: Paper presented at the 22nd international conference on world wide web, Rio de Janeiro, Brazil, 13–17 May, 2013.
35. Le B, Nguyen H. Twitter sentiment analysis using machine learning techniques. Adv Intell Syst Comput. 2015;358:279–89.
36. Gupta A, Kumaraguru P, Castillo C, Meier P. TweetCred: real-time credibility assessment of content on twitter. In: Paper presented at the 6th international conference on social informatics, Barcelona, Spain, 11–13 November, 2014.

37. Antoniadis S, Litou I, Kalogeraki V. A model for identifying misinformation in online social networks. LNCS. 2015;9415:473–82.
38. Mitra T, Gilbert E. CREDBANK: a large-scale social media corpus with associated credibility annotations. In: Paper presented at the 9th international AAAI conference on web and social media, Oxford, UK, 26–29 May, 2015.
39. Angelov S, Grefen P, Greefhorst D. A framework for analysis and design of software reference architectures. Inf Softw Technol. 2011;54(4):417–31.
40. Galster M, Avgeriou P. Empirically-grounded reference architectures: a proposal. In: Paper presented at the joint ACM SIGSOFT conference on quality of software architectures and ACM SIGSOFT conference on quality of software architectures and ACM sigsoft symposium on architecting critical systems, ACM, Boulder, Colorado, USA, 20–24 June, 2011.
41. Pääkkönen P, Pakkala D. The implications of disk-based RAID and virtualisation for write-intensive services. In: Paper presented at the 30th annual ACM symposium on applied computing, Salamanca, Spain, 13–17 April, 2015.
42. Zaharia M, Das T, Li H, Hunter T, Shenker S, Stoica I. Discretized streams: fault-tolerant streaming computation at scale. In: Paper presented at the 24th ACM symposium on operating systems principles, Farmington, Pennsylvania, USA, 3–6 November 2013.
43. SourceForge. JRulesEngine. 2016. http://jruleengine.sourceforge.net. Accessed 08 Nov 2016.
44. Hobbs T. Basic rules of Cassandra data modeling. 2015. http://www.datastax.com/dev/blog/basic-rules-of-cassandra-data-modeling. Accessed 08 Nov 2016.
45. Jersey. 2016. https://jersey.java.net/. Accessed 08 Nov 2016.
46. Eclipse. Jetty. 2016. http://www.eclipse.org/jetty/. Accessed 08 Nov 2016.
47. The Apache Software Foundation. Apache HttpComponents. 2016. https://hc.apache.org/. Accessed 08 Nov 2016.
48. Google code archive. JSON simple. 2016. https://code.google.com/archive/p/json-simple/. Accessed 08 Nov 2016.
49. Twitter developer documentation. https://dev.twitter.com/overview/api/. Accessed 08 Nov 2016.
50. DeepLearning4J. https://deeplearning4j.org/. Accessed 08 Nov 2016.
51. Datastax. CQL for Apache Cassandra. 2016. http://docs.datastax.com/en/cql/3.1/cql/cql_using/useBatch.html. Accessed 08 Nov 2016.
52. Pantsar-Syväniemi S. Situation-based and self-adaptive applications for the smart environment. J Ambient Intell Smart Environ. 2012;4(6):491–516.
53. Pääkkönen P. Feasibility analysis of AsterixDB and Spark streaming with Cassandra for stream-based processing. J Big Data. 2016;3:6.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Immediate publication on acceptance
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com