Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/cosrev

Survey

Understandable Big Data: A survey

*Cheikh Kacfah Emani, Nadine Cullot, Christophe Nicolle**

LE2I UMR6306, CNRS, ENSAM, Univ. Bourgogne Franche-Comté, F-21000 Dijon, France

ARTICLE INFO

Article history:
Received 2 February 2014
Received in revised form
1 May 2015
Accepted 4 May 2015

Keywords:
Big data
Hadoop
Reasoning
Coreference resolution
Entity linking
Information extraction
Ontology alignment

ABSTRACT

This survey presents the concept of *Big Data*. Firstly, a definition and the features of *Big Data* are given. Secondly, the different steps for *Big Data* data processing and the main problems encountered in big data management are described. Next, a general overview of an architecture for handling it is depicted. Then, the problem of merging Big Data architecture in an already existing information system is discussed. Finally this survey tackles semantics (reasoning, coreference resolution, entity linking, information extraction, consolidation, paraphrase resolution, ontology alignment) in the *Big Data* context.

Contents

* *Corresponding author.*
E-mail addresses: cheikh.emani@checksem.fr (C. Kacfah Emani), nadine.cullot@u-bourgogne.fr (N. Cullot), cnicolle@u-bourgogne.fr (C. Nicolle).

http://dx.doi.org/10.1016/j.cosrev.2015.05.002
1574-0137/

1. Introduction

Today, people and systems overload the web with an exponential generation of huge amount of data. The amount of data on the web is measured in exabytes (10^{18}) and zettabytes (10^{21}). By 2025, the forecast is that the Internet will exceed the brain capacity of everyone living in the whole world [1]. This fast growth of data is due to advances in digital sensors, communications, computation, and storage that have created huge collections of data.[1] The term *Big Data* had been coined, by Roger Magoulas (according to [2]), to describe this phenomenon.

Seven recent papers (including [3] and [4]) have aimed to extract Big Data trends, challenges and opportunities. [5] provide a survey on scalable database management: updating of heavy application, analytics and decision support. Likewise, [6] study analytics in *Big Data* with a focus on data warehouse. These two papers have different goals comparatively to [7]. In a more rigorous way, M. Pospiech and C. Felden [7] have selected relevant and recent papers which tackle different aspects of *Big Data* and have clustered them in four domains: *Technical data provisioning* (acquisition, storage, processing), *Technical data utilization* (computation and time complexity), *Functional data provisioning* (information life cycle management, lean information management, value oriented information management, etc.) and *Functional data utilization* (realms where big data is used). At the end of their clustering, [7] note that a lot of papers (87%) are technical and that there is not any paper on functional data provisioning. More closed (compared to the three previous works) to our target, semantics in the age of *Big Data*, [8] focus on knowledge discovery and management in *Big Data* era (flooding of data on the web). As our paper they zoom on gathering relational facts, information extraction, emergence of structure, etc. But a deep circonscription of the concept of *Big Data* is not in the scope of their article like some other key themes of this paper like reasoning on large and uncertain OWL triples, coreference resolution, ontology alignment. The last paper has been authored by [9]. They present *Big Data* integration in a easy-understandable-way. *Schema alignment*, *record linkage* and *data fusion* are presented w.r.t to *Big Data* characteristics (volume, velocity and variety). Knowing the high value carried by data in general and thus by *Big Data*, it is not surprising therefore that Chief Information Officers (CIOs) are interested in it analytics as technological. If initially web pages and traditional databases were the raw materials respectively for search engine companies and other businesses, now it has been mixed with large sets of miscellaneous, heterogeneous and unstructured data. It implies that tools and techniques have to be designed to disambiguate it before putting it together to master and manage data of organizations. Our work is similar to [9] in the approach. We discuss challenges and opportunities of semantics in the age of *Big Data* and present the supply chain to handle it. Therefore, this article defines *Big Data* (Section 2), briefly discusses its management (Section 3) and finally tackles *Big Data* and semantics challenges and opportunities (Section 4).

2. What is big data?

Manyika et al. [10, page 1] define *Big Data* as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze". Likewise, Davis and Patterson [1, page 4] say "Big data is data too big to be handled and analyzed by traditional database protocols such as SQL"; and the same opinion is shared by [11,3,4], etc. Both groups of authors previously mentioned go beyond the only size aspects of data when defining Big Data! Edd Dumbill in [12, page 3] explicitly conveys the multi-dimensionality of Big Data when adding that "the data is too big, moves too fast, or doesn't fit the strictures of your database architectures". This quotation allows us to see that extra characteristics should be added to large datasets to be considered as *Big Data*, or *big size data* as often found throughout the literature [2].

Now it is assumed that *size* is not the only feature of *Big Data*. Many authors [1,12,11,9,13,4] explicitly use the **Three V's** (*Volume*, *Variety* and *Velocity*) to characterize *Big Data*. If the three V's are largely found in the literature, many authors [10,13] and institutes like IEEE focus on Big Data *Value*, *Veracity* and *Visualization*. This last "V" to notice how important it is to provide good tools to figure out data and analysis' results.[2]

Volume (Data in rest). The benefit gained from the ability to process large amounts of information is the main attraction of big data analytics. Having more data beats having better models [12]. The consequence is that it is a trend for many companies to store vast amount of various sorts of data: social networks data, health care data, financial data, biochemistry and genetic data, astronomical data, etc.

Variety (Data in many forms). These data do not have a fixed structure and rarely present themselves in a perfectly ordered form and ready for processing [12]. Indeed,

[1] http://www.cra.org/ccc/docs/init/Big_Data.pdf.

[2] http://www.esg-global.com/blogs/the-6-vs-the-bianalytics-game-changes-so-microsoft-changes-excel/.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

such data can be highly structured (data from relational databases), semi-structured (web logs, social media feeds, raw feed directly from a sensor source, email, etc.) or unstructured (video, still images, audio, clicks) [12]. Another "V", for **Variability**, can be added to *variety* to emphasize on semantics, or the variability of meaning in language and communication protocols.

Velocity (Data in motion). Velocity involves streams of data, structured records creation, and availability for access and delivery.[3] Indeed it is not just the velocity of the incoming data that is the issue: it is possible to stream fast-moving data into bulk storage for later batch processing, for example. The importance lies in the speed of the feedback loop, taking data from input through to decision [12].

Value (Data in highlight). This feature is the purpose of *Big Data* technology. This view is well expressed by the `International Data Corporation`[4] when saying that *Big Data* architectures are: *"designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis"*. This value falls into two categories: analytical use (replacing/supporting human decision, discovering needs, segmenting populations to customize actions) and enabling new business models, products and services [12,10].

Veracity (Data in doubt). Veracity is what is conform with truth or fact, or in short, Accuracy, Certainty, Precision. Uncertainty can be caused by inconsistencies, model approximations, ambiguities, deception, fraud, duplication, incompleteness, spam and latency. Due to veracity, results derived from Big data cannot be proven; but they can be assigned a probability.

To conclude, dealing effectively with *Big Data* requires one to create *value* against the *volume*, *variety* and *veracity* of data while it is still in motion (*velocity*), not just after it is at rest [11]. And at the end, as recommended by [13], scientists must jointly tackle *Big Data* with all its features.

3. Big data management

Basically, data processing is seen as the gathering, processing, management of data for producing "new" information for end users [3]. Over time, key challenges are related to *storage*, *transportation* and *processing* of high throughput data. It is different from *Big Data* challenges to which we have to add *ambiguity, uncertainty* and *variety* [3]. Consequently, these requirements imply an additional step where data are cleaned, tagged, classified and formatted [3,14]. Karmasphere[5] currently splits *Big Data* analysis into four steps: *Acquisition* or *Access*, *Assembly* or *Organization*, *Analyze* and *Action* or *Decision*. Thus, these steps are mentioned as the **"4 A's"**. The `Computing Community Consortium` [14] similarly to [3], divides the organization step into an *Extraction/Cleaning* step and an *Integration* step.

Acquisition. Big Data architecture has to acquire high speed data from a variety of sources (web, DBMS(OLTP), NoSQL, HDFS) and has to deal with diverse access protocols. It is where a filter could be established to store only data which could be helpful or "raw" data with a lower degree of uncertainty [14]. In some applications, the conditions of generation of data are important, thus it could be interesting for further analysis to capture these metadata and store them with the corresponding data [14].

Organization. At this point the architecture has to deal with various data formats (texts formats, compressed files, variously delimited, etc.) and must be able to parse them and extract the actual information like named entities, relation between them, etc. [14]. Also this is the point where data have to be clean, put in a computable mode, structured or semi-structured, integrated and stored in the right location (existing data warehouse, data marts, Operational Data Store, Complex Event Processing engine, NoSQL database) [14]. Thus, a kind of ETL (extract, transform, load) had to be done. Successful cleaning in *Big Data* architecture is not entirely guaranteed; in fact "the volume, velocity, variety, and variability of Big Data may preclude us from taking the time to cleanse it all thoroughly".[6]

Analyze. Here we have running queries, modeling, and building algorithms to find new insights. Mining requires integrated, cleaned, trustworthy data; at the same time, data mining itself can also be used to help improve the quality and trustworthiness of the data, understand its semantics, and provide intelligent querying functions [14]. **Decision.** Being able to take valuable decisions means to be able to efficiently interpret results from analysis. Consequently it is very important for the user to "understand and verify" outputs [14]. Furthermore, *provenance* of the data (supplementary information that explains how each result was derived) should be provided to help the user to understand what he obtains.

If we can easily see how volume, velocity, veracity and variety influence the pipeline of *Big Data* architecture, there is another important aspect in data to handle in Big Data Architecture: *privacy*. R. Hillard[7] considers it to be very important that privacy appears in a good place in his definition of Big Data. Privacy can cause problems at the *creation of data* (someone who wants to hide some piece of information), at the *analysis on data* [1] because if we want to aggregate data or to correlate it we could have to access private data; and privacy can also cause inconsistencies at the *purging of database*. Indeed if we delete all individuals data we can get incoherences with aggregate data.

To sum up handle *Big Data* implies having an infrastructure *linear scalable*, able to handle *high throughput multi-formatted data*, *fault tolerant, auto recoverable*, with a *high degree of parallelism* and a *distributed data processing* [3]. It is important to note that, in this management, integrating data (i.e "access, parse, normalize, standardize, integrate, cleanse, extract, match, classify, mask, and deliver data." [4, chap. 21]) represents 80% of a *Big Data* project. This aspect is deeply discussed in Section 3.3.

[3] http://www.gartner.com/newsroom/id/1731916.
[4] http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf.
[5] http://www.reuters.com/article/2011/09/21/idUS132142+21-Sep-2011+BW20110921.
[6] http://makingdatameaningful.com/2012/12/10/big-data-the-4-vs-the-simple-truth/.
[7] http://mike2.openmethodology.org/blogs/information-development/2012/03/18/its-time-for-a-new-definition-of-big-data/.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

3.1. *Big Data technologies*

There are various tools which can be used in *Big Data* management from data acquisition to data analysis. Most of these tools are parts of Apache projects and are constructed around the famous Hadoop. Written in Java and created by Doug Cutting, Hadoop brings the ability to cheaply process large amounts of data, regardless of its structure [12]. Hadoop is made up of two core projects: *Hadoop Distributed File System* (HDFS) and MapReduce.

HDFS. HDFS is a distributed file system designed to run on large clusters of commodity hardware based on Google File System (GFS) [15,16,3]. Shvachko et al. [17, page 1] add HDFS strengths in their definition when saying it "is designed to store very large datasets reliably, and to stream those datasets at high bandwidth to user applications". By large, we mean from 10 to 100 GB and above [12,16]. While the interface to HDFS is patterned after the UNIX file system, it trades off some POSIX requirements for performance [17,15,16]. HDFS is dedicated to batch processing rather than interactive use by users [16,12]. In HDFS applications, files are written once and accessed many times [16,18]; consequently data coherency is ensured and data are accessed in high throughput [16]. With HDFS file system metadata are stored in a dedicated server, the NameNode, and the application data in other servers called DataNodes. Except for processing large datasets, HDFS has many other goals whose major is to detect and handle failures at the application layer. This objective is realized through a well-organized mechanism of *replication* where files are divided into blocks. Each block is replicated on a number of datanodes; all the datanodes containing a replica of a block are not located in the same *rack*.

MapReduce. Originally put in place by Google to solve the web search index creation problem [12], MapReduce is nowadays the main programming model and associated implementation for processing and generating large datasets [19]. The input data format in MapReduce framework is application-specific, is specified by the user [20] and is suitable for semi-structured or unstructured data. The MapReduce's output is a set of $<key, value>$ pairs. The name "MapReduce" expresses the fact that users specify an algorithm using two kernel functions: *"Map"* and *"Reduce"*. The Map function is applied on the input data and produces a list of intermediate $<key, value>$ pairs; and the Reduce function merges all intermediate values associated with the same intermediate key [19] [20]. In a Hadoop cluster, a *job* (i.e a MapReduce program [11]) is executed by subsequently breaking it down into pieces called *tasks*. When a node in Hadoopcluster receives a *job*, it is able to divide it, and run it in parallel over other nodes [12]. Here the data location problem is solved by the *JobTracker* which communicates with the NameNode to help datanodes to send tasks to near-data datanodes. Let us note that this processing in form of $<key, value>$ pairs is not a limitation to processing which does not seem, at first glance, feasible in map-reduce manner. Indeed, MapReduce has been successfully used in RDF/RDFS and OWL reasoning [21,22] and in structured data querying [23].

Around HDFS and MapReduce there are tens of projects which cannot be presented in detail here. Those projects can be classified according to their capabilities:

- **Storage and Management Capability**
  - Cloudera Manager[8]: an end-to-end management application for Cloudera's Distribution of Apache Hadoop.
  - RCFile (*Record Columnar File*) [24], a data placement structure for structured data. Here, tables are vertically and horizontally partitioned, *lazily compressed*. It is an efficient storage structure which allows fast data loading and query processing.
- **Database Capability**:
  - Oracle NoSQL a high performance $<key, value>$ pair database convenient for non-predictive and dynamic data thus for Big Data;
  - Apache HBase a distributed, column-oriented database management system, modeled on Google's Big Table [10], that runs on top of HDFS [11,12,15];
  - Apache Cassandra a database which combines the convenience of column-indexes and the performance of log-structured updates;
  - Apache Hive can be seen as a distributed data warehouse [15]. It enables easy data ETL from HDFS or other data storage like HBase [11,15] or other traditional DBMS [25]. It has the advantage of using a SQL-like syntax, the Hive QL;
  - Apache ZooKeeper is "an open-source, in-memory, distributed NoSQL database" [3, page 69] that is used for coordination and naming services for managing distributed applications [3,12,11,15].
- **Processing Capability**
  - Pig which is intended to allow people using Hadoop to focus more on analyzing large datasets and thus spend less time having to write mapper and reducer programs [11,12];
  - Chukwa which is a data collection system for monitoring large distributed systems [26,15];
  - Oozie which is a open-source tool for handling complex pipelines of data processing [12,3,11]. Using Oozie, users can define actions and dependencies between them and it will schedule them without any intervention [11].
- **Data Integration Capability**
  - Apache Sqoop: a tool designed for transferring data from a relational database directly into HDFS or into Hive [12,18]. It automatically generates classes needed to import data into HDFS after analyzing the schema's tables; then the reading of tables' contents is a parallel MapReduce job;
  - Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It is designed to import streaming data flows [12,27].

Visualization techniques

Making valuable decisions is the ultimate goal of Big Data analysis and the achievement of this goal requires good visualization of Big Data content. For this reason, there is a real interest in the field of visualization [4,3] i.e "techniques and technologies used for creating images, diagrams, or animations to communicate, understand, and improve the

[8] http://www.cloudera.com/content/cloudera/en/products-and-services/cloudera-manager.html.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

results of big data analyses" [10]. Let us note that visualization in Big Data context is static. Indeed, data are not stored in a relational way and real-time updates require processing large amount of data; but this problem has started to be addressed [3]. Here we present some techniques for Big Data visualization.[9]

- **Tag Cloud.** It is a method for visualizing and linking concepts of a precise domain or web site. These concepts are written using text properties such as font size, weight, or color.
- **Clustergram.** M. Schonlau [28] defines clustergram as a visualization technique used for cluster analysis displaying how individual members of a dataset are assigned to clusters as the number of clusters increases. As for every clustering process the number of clusters is important and it has the advantage to easily perceive how the number influences partitioning results.
- **History Flow.** F.B. Viégas, M. Wattenberg and K. Dave [29] present history flow as a visualization technique designed to show the evolution of a document efficiently with respect to the contributions of its different authors. The horizontal axis of a history flow carries time and the vertical axis the names of the authors. A color code is assigned to each author and the vertical length of a bar indicates the amount of text written by each author.
- **Spatial information flow.** It is another visualization technique that represents spatial information flows. It is mostly represented as a lighting graph where edges connect sites located on a map.

Visualization can also be used to *solve Big Data* problems. For a brief review on this topic, see [30].

3.2. *Data analytics*

Big Data Analytics can be defined as the use of advanced analytic techniques on big data [31]. Nowadays, we can put big data and analytics together. The prior conditions are present for the development of big data Analytics. First of all, Tools and storage capabilities can handle big data. Next, by its size, big data provides large statistical samples and enhanced results of experiments. Finally, companies and governments have clearly identified the benefits to develop the economics of big data. Due to the characteristics of big data, mainly variety, there are many techniques used for analytics on big data [32].

- *Association rule learning* to find relationships among entities (mainly used in recommendation systems).
- *Machine learning* to bring computer to learn complex patterns and make intelligent decisions based on it [10].
- *Data mining* which can be seen as a combination of statistics and machine learning and statistics with database management [10].
- *Cluster analysis* used as unsupervised machine learning. It aims to divide data into smaller clusters having the same set of characteristics not known in advance.
- *Crowdsourcing* used to collect data and/or features and metadata to enhance the current semantics of data.
- *Text analytics* which aims to analyze large text collections (email, web pages, etc.) to extract information. It is used for topics modeling, question answering, etc.

Some proposals emphasize that those techniques rely on a generalized picture of the underlying knowledge. Due to their design they fail to capture the subtleties of the processes which produce these data [33,34]. Moreover, these techniques sometimes behave badly with very large datasets. It is the case for example of learning-based techniques. There, size of training data can exceed memory or the fast growing number of features can lead to a high execution time. Sengamedu [35] presents some scalable methods which can be applied for machine learning (Random Projections, Stochastic Gradient Descent and MinClosed sequences). Trends about big data analytics are summarized within [31]. They mainly concern *visualization* of multi-form, multi-source and real-time data. Moreover, the size of data limits *in-memory* processing.

3.3. *Adding Big Data capability to an existing information system*

A whole book can be written on this topic. It is what had been done by [3] by the study of data warehousing in the age of *Big Data*. A number of strategies of this integration are presented in Table 1. The first step of that integration is about data acquisition. Since traditional databases have to deal with structured data, existing ecosystem needs to be extended across all of the data types and domains. Then, data integration capability needs to deal with velocity and frequency. The challenge here is also about ever growing volume and, because many technologies leverage `Hadoop`, use technologies that allow you to interact with `Hadoop` in a bi-directional manner: load and store data (`HDFS`) and process and reuse the output (`MapReduce`) for further processing. [14, page 12] reminds us that the main challenge is not to build "that is ideally suited for all processing tasks" but to have an underlying architecture flexible enough to permit to processes built on top to work at their full potential. For sure there is not a commonly agreed solution, an infrastructure is intimately tied to the purpose of the organization in which it is used and consequently to the kind of integration (real-time or batch). More and other important questions have to be answered: are Big Data stored timeliness or not [4]?

4. Big data quality, the next semantic challenge

A question that experts of the knowledge management ask themselves is to know if Big Data can leverage on semantics. The answer to this question is obviously "yes". Companies and governments are interested in two types of data in a big data context. First, they consider data generated by human, mainly those disseminated through web tools (social networks, cookies, emails...). Secondly they want to merge data generated from connected objects. The Internet of human beings and the internet of things become a mix of big

[9] For technologies see [3].

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

Table 1 – Four types of data integration strategies described by K. Krishnan in [3] with their main characteristics, pros and cons.

Data-driven integration	External integration
-Categorization of data by type (transactional, analytical, semi-structured, unstructured) -*Pros*: infrastructure can be adapted to each category. Idem for workload types (w.r.t. Volume of data and latency) -*Cons*: possible various Efforts on the same architecture	-Big Data and classic warehouse in two platforms -A data bus for connection -*Pros*: the platforms can scale each, overload is reduced, modularity, etc. -*Cons*: complexity of data bus architecture integration can drop performance over time, poor metadata handling
Integration-driven approach	**Big Data appliances**
-Combining Big Data and existing warehouse platforms - A Hadoop/NoSQL connector links them *Pros*: the platforms can scale each, overload is distributed, modularity, good metadata handling -*Cons*: the connector is Achilles' heel, complexity of data integration	-A black box from vendors with three layers (Big Data, RDBMS and integration) -*Pros*: scalable and modular custom configuration for users (organizations) -*Cons*: custom configuration by vendors can change frequently and can be source of heavy maintenance

data that must be targeted to understand, plan and act in a predictive way. This perspective raises new questions about the quality of the data. In this context, people do not agree with the definition of quality. The quality of the data may be its high processing level or its relevance according to the reality they represent. In fact, since Big Data is big and messy, challenges can be classified into *engineering tasks* (managing data at an unimaginable scale) and *semantics* (finding and meaningfully combining information that is relevant to your needs) [36] have identified each a relevant challenge for *Big Data*:

1. the meaningful data integration challenge which can be seen as a five-step challenge: (1) *define* the problem to solve, (2) *identify* relevant pieces of data in Big Data, (3) *ETL* it into appropriate formats and store it for processing, (4) *disambiguate* it and (5) *solve* the problem.
2. the Billion Triple Challenge which aims to process large-scale RDF to provide a full description of each entity of the triple in a single target vocabulary and to link that entity to the corresponding sources.
3. the `Linked Open Data (LOD) Ripper` for providing good use cases for LOD and to able to link them with non LOD efficiently.
4. the value of the use of semantics in data integration and in the design of future DBMS.

Similar challenges have been identified by S. Auer and J. Lehmann [37]. Unlike [36], [37] proposes solutions for some of these challenges (data integration, scalable reasoning, etc.). Semantics could be considered as a magical world to bridge the gap of the hétérogénéity of data. Moreover, semantics can be used in a decidable system which makes possible to detect inconsistency of data, generates new knowledge using inference engine or simply links more accurately specific data not relevant for machine learning based techniques. In the literature, we can find work whose purpose is about the challenges mentioned before. Before presenting them, we must note that the relation between Big Data and semantics is bidirectional. As it is true for Big Data leverages on semantics, some semantics tasks are optimized by using tools designed for large dataset processing, especially `MapReduce` framework. More, in the articles cited in the following lines, the term *Big Data* is rarely explicitly mentioned; it could be hidden behind terms like "web scale/web-scale" or "large scale/large-scale" [21,38–41] to express the *volume* feature, "real-time" or "dynamic" [42,43] to express *velocity* and "informal/informality", "natural language", "unstructured" or "data streams" [44–47] to state the *variety/variability* feature. In another way, *Big Data* can be experienced through Linked Data: it has volume, variety, and veracity features and we can thus assume that other characteristics are under control [13].

4.1. *Identifying relevant pieces of information in messy data*

As mentioned in the challenges list, this task can be done before the disambiguation pile. In this case, we must prune irrelevant data. This pruning is mostly done by a *"bag of words"* approach. It helps through *cosine similarity* to rapidly compare things (documents in [48] and sentences in [49, chap. 5]) and to select relevant one, according to a threshold. If done after the pile, this task can be seen as the build of a *Big Data* index. Obviously, this problem is mainly broached by people who intend to design a search engine. Therefore, we have in [38] a built of an inverted index, for fast keyword-search answering, where a Lucene document is output for each entity and a structured index to easily retrieve pieces of information about a given entity. Likewise, but on RDF databases, [50] use B + − Trees to index object identifiers of RDF nodes and also use an inverted index to improve keyword queries. Unlike previous cited authors, [51] for Querying Distributed RDF Repositories purposes, built indices on "schema paths" (concepts whose instances have to be joined to answer a given query) to identify the sources which may contain the information needed.

4.2. *The disambiguation pile*

The last steps of the first challenge mentioned above can be reformulated by our *disambiguation pile* shown by Fig. 1.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

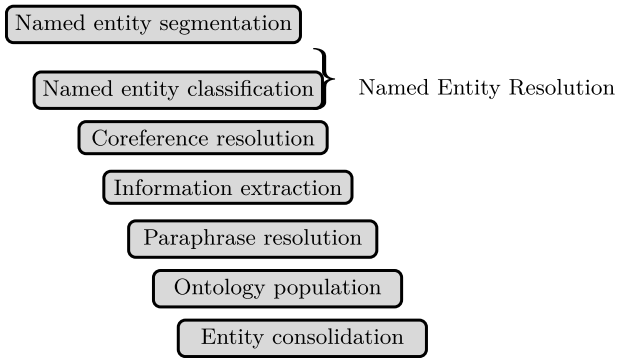

Fig. 1 – The disambiguation pile of natural language.

4.2.1. *Named entity resolution (NER)*

Usually, NER is a task well resolved by existing tools. But with advent of social media like Facebook and particularly Twitter, the writing style has deeply changed and new techniques have to be developed. It is the case of [45,52,53] which tackle NER in tweets. These basic tools are actually used to solve more complex problems like event extraction in tweets [46]. Since tweets are short, very informal and noisy (username mentions, URLs, various markers (@, #)) some changes have to be done to improve classic NLP tools like Stanford NER or Open NLP. [52,45,53] share the opinion that supervised learning gives good results and that learning corpus must be made up with tweets. To improve NER, gazetteers [52] or entities repositories like `Freebase` [45] have to be extended (many new entities are missing there e.g: "Nintendo DS lite" (a product), "Blue Stone 42" (a tv-show), etc.). Moreover, variations of words have to be clustered and normalized (e.g: "tomorrow" can be written *'2mr'*, *'2mro'*, *'2mrrw'*, *'2mrw'*, ...) [45,53,54] and we must know if we can learn something about capitalization of words (which is randomly done in tweets universe) in a given tweet [45].

4.2.2. *Coreference resolution*

Coreference resolution is the task of finding all expressions that refer to the same entity in a discourse [55]. In this domain, improvements are not related to *Big Data* features and are mainly focused on enrichment and precision of new lexical and syntactic features and global inference [56,55,57]. Haghighi and Klein [56] introduce new syntactic, semantic features and discourse phenomena to improve existing systems. Their work has been completed by additional features (e.g: *Denonym*, *Word inclusion* in [57], *Speaker identification* in [55], *Web features* like *General co-occurrence*, *Hearst co-occurrence*, *Entity-based context*, *Pronoun context* in [58], etc.). Most models for this task determine if two mentions refer to each other using a single function over a set of constraints or features, but some recent approaches tend to use multi-tiers methods where mentions are disambiguated gradually in well-ordered tiers which apply each, a specific function [55,57]. It is obvious that in a *Big Data* supply chain, such approaches can be difficultly used without modification. Indeed, analyzing billions of documents more than seven times is not realistic. We note that (the direct) approach of [58] (direct) is more scalable, but it is a pairwise disambiguation method.

Once more, we note that very few work have in mind *Big Data* characteristics while addressing coreference resolution.

In challenges about indexing billions of RDF triples or reasoning on them (see further), we see that scientists deal with data formats which are quite easy to handle by a computer (RDF/RDFS, OWL/OWL2). But the transformation of pieces of natural language-written texts into computer-understandable formats have to be done first.

4.2.3. *Information extraction*

One of the intuitive ways to perform this task is to provide hand-written regular expressions (REs) like [59,60]. The results are promising but the number of manually-written REs (165 REs for a 9-concept ontology [59]) makes it hard to handle. More, their approach does not focus on scalability unlike [61,40] who propose a REs pattern-based tool named `OnTeA`. `OnTeA` takes advantage of `Hadoop MapReduce` to scale. More and more, automatic approaches had been proposed. It is the case of `KNOWITALL` [62] and `TextRunner`. The former uses predefined patterns and rule templates to populate classes in a given ontology. Though automatic, `KNOWITALL` does not scale: a web-document is processed several times for patterns matching and many web-queries are done to assign a probability to a concept, etc. Thus, `TextRunner` which implements the new extraction paradigm of *Open Information Extraction (OIE)* had been introduced. In OIE, we are not limited in a set of triples but try to extract all of them [8,47]. More recently, following `REVERB`, [63] present `OLLIE`. Unlike `REVERB`, `OLLIE` can extract relation not mediated by verb and in certain case can provide the context of a relation (e.g: *"If he wins five key states, Romney will be elected President."* ⟶ (the wining of key states *determines* the election fact)).

In this facts harvesting task, some recent approaches focus on scalability in addition to recall and precision. It is the case of [41] which take advantage of `Hadoop MapReduce` to distribute the patterns matching part of their algorithm. Now focusing on the velocity, almost the same group of authors has proposed a novel approach for population of knowledge bases in [43]. Here, they propose to extract a certain set of relations from documents in a given "time-slice". This extraction can be improved based on the topics covered by the document (e.g do not try to extract music-domain relations from a sport document) or by matching patterns of relations on an index build from documents. More, since web is redundant (a given fact is published by tens of sites), a small percentage of documents can cover a significant part of facts. Likewise, [42] RDF-format unstructured data during a time-slice duration. It is important to note that the whole processing of data gather during a period of time must be done during that period of time, unless the processing cycle will be blocked. Recall that relations could be *n*-ary. For instance, in [64]'s web representative-corpus, *n*-ary relations represented 40% of all relations. About *n*-ary relations extraction, [65,66] are very relevant work. They both use `Stanford CoreNLP` typed dependencies paths to extract arguments of different facts. To end with information extraction, let us precise that is not all about free text. Some work has thus focus on web *tables* or *lists* [67–69].

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

4.2.4. *Semantic paraphrase resolution*

Paraphrase resolution is also known as *Synonym Resolution, Deduplication, Entity Resolution* [70]. It is related to OIE and can concern relations [71–73], entities or both [70]. In adequacy with our interest for work near *Big Data* features, [70,71] propose unsupervised and scalable approaches for paraphrase resolution. In [70], to scale, the number of comparisons between pairs of strings (which must share a <property, object> or <subject, property> pair) is limited and strings are clustered over time. This idea of gradual merging of clusters is also the reason of the scalability in [71]. Unlike these two previous, [72] tackle polysemy and use extra-characteristics of the input relation instances like distributional similarity, linguistic patterns, hypernym graph, etc. Let us note that if [70] tackle the problem of finding which mentions of entities in a text are equivalent, some authors address a similar problem called *Entity Linking* and which can be helpful in our disambiguation task. It aims to identify an entry in a given Knowledge Base (KB) to which an entity mention in a document refers to [74,75].

4.2.5. *Ontology population*

Since information has been extracted and synonyms identified, our unstructured data must be put in computer-processable form. The current task thus consists in organizing extracted tuples in a querying form such as instances of ontologies, tuples of a database schema or set of quads ($< subject, predicate, object, context >$). This idea is found in [59–61,40] but uses several hand-written regular expressions. It is also found in [76]'s On-demand IE approach and in [77] where they propose a method to map triples output from an OIE process to a domain-ontology. The former approach chooses to escape from an expensive computation problem by using only triples where the relation is verb-based unlike the latter, which takes into account "tuple from each pair of adjacent Noun Phrases". Moreover, the approach in [77] is very domain-specific and in their objective of mapping OIE tuples with a domain-ontology, the authors implicitly assume that all the facts of an event are *inside the same sentence*. This assumption, which is obviously too restrictive, is also found in ontology population tasks [78,79] and in OIE [63,80,47]. Hence, it is clear to us that the first task is to be able to chunk a whole text into a set of events (which are in forms of sentences that are not necessarily contiguous in a given document [49]) and then to map concepts and relations of a given ontology or columns of a given database schema into the extracted pieces of information (from binary or n-ary relations) of each chunk. We see that unlike [77], many approaches work with general concepts (named entities categories like *person, organization, location, date, etc.*) [76,81]. Some work like YAGO [82] try to have some specific concepts (e.g *"American person"*), but it seems to us too general in comparison to [77] where concepts such as *"NFLTeam"*, *"GameWinner"* or *"TeamScoringAll"* can be extracted.

Very few work focus on ontology population in *Big Data* context. The main aspect broached is the identification of the possible class of an entity. More, this identification is too general, and when it is very domain specific it implies a significant part of human intervention.

4.2.6. *Entity consolidation*

Entity consolidation can be seen as the building of `owl:sameAs` closure in OWL-data. In practice this is not always straight. In fact, `owl:sameAs` property is not always explicit. It can be hidden behind inference on an inverse functional property [83], a functional property [21], an equivalent property [84], cardinality restrictions [38,39]. Moreover, an equivalent property can be derived through heuristics (string similarity between properties' short names or labels). Concerning algorithms, [83,21] have similar approaches: to group all equivalent entities in a given set and to assign a unique identifier to them, which will replace entities of its set within real data. To achieve this goal, [83] propose a method which can be run many times due to new derivations implied by an inverse functional property! To obviate these limitations, [21] leverage on their ordering of rules and `MapReduce` parallel capabilities.

4.3. *The billion triple challenge*

At the end of the first challenge we have billions of RDF-triples and we must be able to reason on it. One of the most relevant works which tackle this problem is [21]. Their work has led to a tool termed `WebPIE` (*Web-scale Inference Engine*). In [21], inference rules are rewritten and *map* and *reduce* functions are specified for each of them. This work has inspired the work of [22] who propose a `MapReduce`-based algorithm for classifying $\mathcal{EL}+$ ontologies. Another relevant work in this challenge focuses on efficient RDF repositories partitioning and scalability of SPARQL queries [85]. We can also add [86] which proposes a way to store and retrieve large RDF graphs efficiently. Concerning the (complete) description of entities in the middle of billion RDF/RDFS triple mentioned in the third challenge, [38] designed a Semantic Web Search Engine (SWSE) which has many features including entities description. Here, this description is obtained by aggregating efficiently descriptions from many sources.

If we know how to infer over billion RDF-triples, it is not easy to deal with noise, inconsistency and various errors found in RDF datasets. [87] identify four sources of errors: (i) *accessibility and dereferenceability of URIs*, (ii) *syntax errors*, (iii) *noise and inconsistency* (e.g: use of undefined classes of properties, misuse of a class as a property and vice versa, etc.) and (iv) *ontology hijacking*. [88] propose to repair or to be able to infer in such a noisy context. For repairing, they identify the "minimal inconsistent subset" (MIS) of the ontology and the subsets the MIS will affect. For reasoning, [88] leverage the pioneering work of [89] and propose to answer queries based on consistent subsets (which grows inclusively) of the given ontology. The choice of the subsets are based on syntactic and semantic heuristics. In the same paper, uncertainty in reasoning is handled by adding confidence value to the elements of the ontology.

4.4. *Schema alignment*

Basically, data integration is done in three main steps: *Schema alignment, Record linkage* and *Data fusion* [9]. The previous paragraphs tackle problems relative to disambiguation and good understanding of data: we were working only on instances of knowledge bases. At the end of steps described

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

in these paragraphs, we have a clean knowledge base or ontology. But what if our ontology has to be queried, merged or linked with another one? Answer to this question is ontology alignment (a.k.a. ontology matching) and it has to be done in agreement to *Big Data* requirements (a recent and relevant review of schema alignment with structured data in *Big Data* era is presented in [9]). A deep and recent review of ontology matching is presented in [90]. Aspects of ontology matching which present an interest for us are mentioned there in terms of *challenge*. Some of those aspects like the use of external resources have a direct impact on ontology matching in the context of *Big Data*. It is the case of (i) *matcher selection, combination and tuning* and (ii) *user involvement*. Challenge (i) is relevant to us because matcher uses different techniques and to combine/tune them can improve results. Moreover, the improvements of these techniques can focus on specific aspects (volume, uncertainty) of ontologies. But these combinations can have a negative impact on processing time. The same remark can be done in the second "challenge" since the user can resolve matching errors but it is difficult to rely on users in large ontologies alignment.

In addition, Shvaiko and Euzenat [90] mention the lack of evaluation of scalability as a challenge. Likewise, all these remarks could be made ours, after we have presented main aspects of *Big Data* semantic management. Surely, all the techniques and tools aforementioned can be improved by various parameters or heuristics, but in *Big Data* era, a significant place must be made to optimization. Tools must handle exabytes of data, streaming data, fast changing ones, very informal data, etc.

5. Ethics and privacy

Ethics and privacy have always been a main concern in data management. They are now of big interest with big data. This is due to the multi-dimensionality of big data:

- Due to the huge *volume* of data more pieces of valuable information can be identified or inferred than it was possible before.
- The high *velocity* of data makes feasible analysis in real time and thus a continuous refining of users' profiles.
- The *variety* of data sources make users traceable. In addition the diversity of data types allows data owners to build more complex and rich profiles of users. Moreover, this variety leads to a diversification of business plans making big data more attractive at a bigger level.

This ability to infer new insights from big data with an impact on privacy brings Mayer-Schönberger and Cukier [91] to define big data as "things one can do at a large scale that cannot be done at a smaller one, to extract new insights or create new forms of value, in ways that change markets, organizations, the relationship between citizens and governments, and more."

Thus presented, one can think big data era is exclusively valuable for business people. In accordance to what Davis and Kord said [1], we think that big data era is also worth for the man in the street. It can be seen through services like Google Flu Trends[10] or user recommendation services as those proposed by Netflix or Amazon. The pivotal point is hence about the balance between benefits and drawbacks of snooping around people's big data. Mayer-Schönberger and Cukier [91] propose four principles which could help to find a trade-off in this era of big personal data flow:

- Privacy should be seen as a set of rules encompassing flows of information in ethical ways but not the ability to keep data secret.
- Shared information can still be confidential.
- Big data mining requires transparency.
- Big data can threaten privacy.

6. Conclusion

We are living in the era of data deluge. The term *Big Data* had been coined to describe this age. This paper defines and characterizes the concept of *Big Data*. It gives a definition of this new concept and its characteristics. In addition, a supply chain and technologies for *Big Data* management are presented. During that management, many problems can be encountered, especially during semantic gathering. Thus it tackles semantics (reasoning, coreference resolution, entity linking, information extraction, consolidation, paraphrase resolution, ontology alignment) with a zoom on "V's". It concludes that volume is the most tackled aspect and many works leverage `Hadoop MapReduce` to deal with volume [21,40,41,22]. More and more, unlike velocity, web and social media informality and uncertainty are addressed by scientists. We see that uncertainty can be handled manually (Ripple Down Rules [44]) or automatically (identification and/or isolation of inconsistencies [88]). About velocity, gazetteers and knowledge bases must be continually updated [88,45] and data processed periodically [43,42]. Similarly if we want to tackle variety, we must deal with various data formats (tweets in [45,46,88] and natural language texts [47,80,62,76]) and distributed data [38,39]. As [13] said, *Big Data* must be addressed jointly and on each axis to make significant improvement in its management.

REFERENCES

[1] K. Davis, D. Patterson, Ethics of Big Data: Balancing Risk and Innovation, O'Reilly Media, 2012.
[2] G. Halevi, H. Moed, The evolution of big data as a research and scientific topic: Overview of the literature, Res. Trends (2012) 3–6.
[3] K. Krishnan, Data warehousing in the age of big data, in: The Morgan Kaufmann Series on Business Intelligence, Elsevier Science, 2013.
[4] A. Reeve, Managing Data in Motion: Data Integration Best Practice Techniques and Technologies, Morgan Kaufmann, 2013.
[5] D. Agrawal, S. Das, A. El Abbadi, Big data and cloud computing: current state and future opportunities, in: Proceedings of the 14th International EDBT, EDBT/ICDT '11, ACM, New York, NY, USA, 2011, pp. 530–533.

[10] https://www.google.org/flutrends/.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

[6] A. Cuzzocrea, I.-Y. Song, K.C. Davis, Analytics over large-scale multidimensional data: the big data revolution!, in: Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP, DOLAP '11, ACM, New York, NY, USA, 2011, pp. 101–104.
[7] M. Pospiech, C. Felden, Big data—a state-of-the-art, in: AMCIS, Association for Information Systems, 2012.
[8] F. Suchanek, G. Weikum, Knowledge harvesting in the big-data era, in: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, ACM, New York, NY, USA, 2013, pp. 933–938.
[9] X. Dong, D. Srivastava, Big data integration, in: Data Engineering (ICDE), 2013 IEEE 29th International Conference on, 2013, pp. 1245–1248.
[10] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A.H. Byers, Big Data: The Next Frontier for Innovation, Competition, and Productivity, McKinsey Global Institute, 2011.
[11] P. Zikopoulos, C. Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Education, 2011.
[12] I. O'Reilly Media, Big Data Now: 2014 Edition, O'Reilly Media, 2014.
[13] P. Hitzler, K. Janowicz, Linked data, big data, and the 4th paradigm, Semant. web (2013) 233–235.
[14] H.V. Jagadish, D. Agrawal, P. Bernstein, E. e. a. Bertino, Challenges and Opportunities with Big Data, The Community Research Association, 2015.
[15] T. White, Hadoop: The Definitive Guide, first ed., O'Reilly Media, Inc., 2009.
[16] D. Borthakur, The hadoop distributed file system: Architecture and design, The Apache Software Foundation. (2007) 1–14.
[17] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), MSST '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 1–10.
[18] G. Turkington, Hadoop Beginners Guide, Packt Publishing, Limited, 2013.
[19] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[20] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, C. Kozyrakis, Evaluating mapreduce for multi-core and multiprocessor systems, in: Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture, HPCA '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 13–24.
[21] J. Urbani, S. Kotoulas, J. Maassen, F. Van Harmelen, H. Bal, Webpie: A web-scale parallel inference engine using mapreduce, Web Semant. 10 (2012) 59–75.
[22] R. Mutharaju, F. Maier, P. Hitzler, A mapreduce algorithm for $\mathcal{EL}^+$, in: Proc. 23rd Int. Workshop on Description Logics (DL'10), CEUR Workshop Proceedings, Waterloo, Ontario, Canada, 2010, pp. 464–474.
[23] T. Kaldewey, E.J. Shekita, S. Tata, Clydesdale: structured data processing on mapreduce, in: Proceedings of the 15th International Conference on Extending Database Technology, EDBT '12, ACM, New York, NY, USA, 2012, pp. 15–25.
[24] Y. He, R. Lee, Y. Huai, Z. Shao, N. Jain, X. Zhang, Z. Xu, Rcfile: A fast and space-efficient data placement structure in mapreduce-based warehouse systems, in: Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 1199–1208.
[25] A. Thusoo, J.S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, R. Murthy, Hive - a petabyte scale data warehouse using Hadoop, in: ICDE '10: Proceedings of the 26th International Conference on Data Engineering, IEEE, 2010, pp. 996–1005.
[26] J. Boulon, A. Konwinski, R. Qi, A. Rabkin, E. Yang, M. Yang, Chukwa, a large-scale monitoring system, Cloud Comput. Appl. (2008) 1–5.
[27] C. Wang, I.A. Rayan, K. Schwan, Faster, larger, easier: reining real-time big data processing in cloud, in: Proceedings of the Posters and Demo Track, Middleware '12, ACM, New York, NY, USA, 2012, pp. 4:1–4:2.
[28] M. Schonlau, The clustergram: A graph for visualizing hierarchical and nonhierarchical cluster analyses, Stata J. 2 (4) (2002) 391–402. 12.
[29] F.B. Viégas, M. Wattenberg, K. Dave, Studying cooperation and conflict between authors with history flow visualizations, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, ACM, New York, NY, USA, 2004, pp. 575–582.
[30] D. Keim, H. Qu, K.-L. Ma, Big-data visualization, computer graphics and applications, IEEE 33 (4) (2013) 20–21.
[31] P. Russom, et al. Big data analytics, TDWI Best Practices Report, Fourth Quarter.
[32] D. Maltby, Big data analytics, in: 74th Annual Meeting of the Association for Information Science and Technology (ASIST), 2011, pp. 1–6.
[33] A. Hoppe, R. Ana, N. Christophe, Automatic user profile mapping to marketing segments in a bigdata context, in: Proceedings of the 14th International Conference on informatics in Economy, IE'15, 2015, pp. 285–291.
[34] R. Peixoto, T. Hassan, C. Cruz, A. Bertaux, S. Nuno, Semantic hmc for business intelligence using cross-referencing, in: Proceedings of the 14th International Conference on Informatics in Economy, IE'15, 2015, pp. 571–576.
[35] S.H. Sengamedu, Scalable analytics–algorithms and systems, in: Big Data Analytics, Springer, 2012, pp. 1–7.
[36] C. Bizer, P. Boncz, M.L. Brodie, O. Erling, The meaningful use of big data: four perspectives – four challenges, SIGMOD Rec. 40 (4) (2012) 56–60.
[37] S. Auer, J. Lehmann, Creating knowledge out of interlinked data, Semant. web 1 (1,2) (2010) 97–104.
[38] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, S. Decker, Searching and browsing linked data with *swse*: The semantic web search engine, Web Semant. 9 (4) (2011) 365–401.
[39] A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres, S. Decker, Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora, Web Semant. 10 (2012) 76–110.
[40] M. Laclavík, M. Šeleng, L. Hluchý, Towards large scale semantic annotation built on mapreduce architecture, in: Proceedings of the 8th International Conference on Computational Science, Part III, ICCS '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 331–338.
[41] N. Nakashole, M. Theobald, G. Weikum, Scalable knowledge harvesting with high precision and high recall, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, ACM, New York, NY, USA, 2011, pp. 227–236.
[42] D. Gerber, A.-C. Ngonga Ngomo, S. Hellmann, T. Soru, L. Bühmann, R. Usbeck, Real-time rdf extraction from unstructured data streams, in: Proceedings of ISWC, 2013.
[43] N. Nakashole, G. Weikum, Real-time population of knowledge bases: opportunities and challenges, in: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 41–45.
[44] M.H. Kim, Ripple-down rules based open information extraction for the web documents (Ph.D. thesis), School of Computer Science and Engineering, The University of New South Wales, Australia, 2012.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

[45] A. Ritter, S. Clark, Mausam, O. Etzioni, Named entity recognition in tweets: an experimental study, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 1524–1534.

[46] A. Ritter, Mausam, O. Etzioni, S. Clark, Open domain event extraction from twitter, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, ACM, New York, NY, USA, 2012, pp. 1104–1112.

[47] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 2670–2676.

[48] N. Chambers, D. Jurafsky, Template-based information extraction without the templates, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 976–986.

[49] L. Jean-Louis, Approches supervisées et faiblement supervisées pour l'extraction d'événements complexes et le peuplement de bases de connaissances (Ph.D. thesis), Université de Paris 11 - Paris Sud, France, 2011.

[50] A. Harth, S. Decker, Optimized index structures for querying rdf from the web, in: Proceedings of the Third Latin American Web Congress, LA-WEB '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 71–80.

[51] H. Stuckenschmidt, R. Vdovjak, G.-J. Houben, J. Broekstra, Index structures and algorithms for querying distributed rdf repositories, in: Proceedings of the 13th International Conference on World Wide Web, WWW '04, ACM, New York, NY, USA, 2004, pp. 631–639.

[52] X. Liu, S. Zhang, F. Wei, M. Zhou, Recognizing named entities in tweets, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 359–367.

[53] K. Bontcheva, L. Derczynski, A. Funk, M. Greenwood, D. Maynard, N. Aswani, Twitie: An open-source information extraction pipeline for microblog text, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing, Association for Computational Linguistics, 2013.

[54] L. Derczynski, A. Ritter, S. Clark, K. Bontcheva, Twitter part-of-speech tagging for all: Overcoming sparse and noisy data, in: Proceedings of Recent Advances in Natural Language Processing (RANLP), Association for Computational Linguistics, 2013.

[55] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, D. Jurafsky, Deterministic coreference resolution based on entity-centric, precision-ranked rules, Comput. Linguist. 39 (4) (2013).

[56] A. Haghighi, D. Klein, Simple coreference resolution with rich syntactic and semantic features, in: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3, EMNLP '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 1152–1161.

[57] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, C. Manning, A multi-pass sieve for coreference resolution, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 492–501.

[58] M. Bansal, D. Klein, Coreference semantics from web features, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, ACL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 389–398.

[59] D.W. Embley, D.M. Campbell, R.D. Smith, S.W. Liddle, Ontology-based extraction and structuring of information from data-rich unstructured documents, in: CIKM, CIKM '98, ACM, New York, NY, USA, 1998, pp. 52–59.

[60] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.K. Ng, R.D. Smith, Conceptual-model-based data extraction from multiple-record web pages, Data Knowl. Eng. 31 (3) (1999) 227–251.

[61] M. Laclavík, M. Šeleng, E. Gatial, Z. Balogh, L. Hluchy, Ontology based text annotation—ontea, in: Proceedings of the 2007 conference on Information Modelling and Knowledge Bases XVIII, IOS Press, Amsterdam, The Netherlands, 2007, pp. 311–315.

[62] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, A. Yates, Web-scale information extraction in knowitall: (preliminary results), in: Proceedings of the 13th International Conference on World Wide Web, WWW '04, ACM, New York, NY, USA, 2004, pp. 100–110.

[63] Mausam, M. Schmitz, R. Bart, S. Soderland, O. Etzioni, Open language learning for information extraction, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 523–534.

[64] J. Christensen, Mausam, S. Soderland, O. Etzioni, An analysis of open information extraction based on semantic role labeling, in: M.A. Musen, S. Corcho (Eds.), K-CAP, ACM, 2011, pp. 113–120.

[65] A. Akbik, A. Löser, Kraken: N-ary facts in open information extraction, in: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 52–56.

[66] L. Del Corro, R. Gemulla, Clausie: clause-based open information extraction, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2013, pp. 355–366.

[67] R. Pimplikar, S. Sarawagi, Answering table queries on the web using column keywords, Proc. VLDB Endow. 5 (10) (2012) 908–919.

[68] M.J. Cafarella, A. Halevy, D.Z. Wang, E. Wu, Y. Zhang, Webtables: exploring the power of tables on the web, Proc. VLDB Endow. 1 (1) (2008) 538–549.

[69] H. Elmeleegy, J. Madhavan, A. Halevy, Harvesting relational tables from lists on the web, Proc. VLDB Endow. 2 (1) (2009) 1078–1089.

[70] A. Yates, O. Etzioni, Unsupervised resolution of objects and relations on the web, in: C.L. Sidner, T. Schultz, M. Stone, C. Zhai (Eds.), HLT-NAACL, The Association for Computational Linguistics, 2007, pp. 121–130.

[71] S. Kok, P. Domingos, Extracting semantic networks from text via relational clustering, in: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I, ECML PKDD '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 624–639.

[72] B. Min, S. Shi, R. Grishman, C.-Y. Lin, Ensemble semantics for large-scale unsupervised relation extraction, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 1027–1037.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002

[73] M. Paşca, P. Dienes, Aligning needles in a haystack: paraphrase acquisition across the web, in: Proceedings of the Second International Joint Conference on Natural Language Processing, IJCNLP'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 119–130.
[74] M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin, Entity disambiguation for knowledge base population, in: Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 277–285.
[75] Z. Zheng, X. Si, F. Li, E.Y. Chang, X. Zhu, Entity disambiguation with freebase, in: Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 82–89.
[76] S. Sekine, On-demand information extraction, in: Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL '06, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 731–738.
[77] S. Soderland, B. Roof, B. Qin, S. Xu, Mausam, O. Etzioni, Adapting open information extraction to domain-specific relations, AI Mag. 31 (3) (2010) 93–102.
[78] D. Tunaoglu, Ö Alan, O. Sabuncu, S. Akpinar, N.K. Çiçekli, F.N. Alpaslan, Event extraction from turkish football web-casting texts using hand-crafted templates, in: ICSC, 2009, pp. 466–472.
[79] S. Kara, O. Alan, O. Sabuncu, S. Akpinar, N.K. Çiçekli, F.N. Alpaslan, An ontology-based retrieval system using semantic indexing, Inf. Syst. 37 (4) (2012) 294–305.
[80] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 1535–1545.
[81] J.M. Ruiz-Martínez, J.A. Miñarro Giménez, D. Castellanos-Nieves, F. García-Sánchez, R. Valencia-García, Ontology population: An application for the e-tourism domain, Int. J. Innovative Comput. Inf. Control 7 (11) (2011) 6115–6133.
[82] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 697–706.
[83] A. Hogan, A. Harth, S. Decker, Performing object consolidation on the semantic web data graph, in: In Proceedings of 1st I3: Identity, Identifiers, Identification Workshop, 2007.
[84] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, S. Decker, Sig.ma: Live views on the web of data, Web Semantics: Sci. Serv. Agents on the World Wide Web 8 (4) (2010) 355–364.
[85] J. Huang, D.J. Abadi, K. Ren, Scalable sparql querying of large rdf graphs, Proc. VLDB Endow. 4 (11) (2011) 1123–1134.
[86] M. Farhan Husain, P. Doshi, L. Khan, B. Thuraisingham, Storage and retrieval of large rdf graph using hadoop and mapreduce, in: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 680–686.
[87] A. Hogan, A. Harth, A. Passant, S. Decker, A. Polleres, Weaving the pedantic web, in: LDOW, in: CEUR Workshop Proceedings, vol. 628, CEUR-WS.org, 2010.
[88] B. Liu, J. Li, Y. Zhao, Repairing and reasoning with inconsistent and uncertain ontologies, Adv. Eng. Softw. 45 (1) (2012) 380–390.
[89] Z. Huang, F. Van Harmelen, A. Ten Teije, Reasoning with inconsistent ontologies, in: IJCAI, vol. 5, 2005, pp. 254–259.
[90] P. Shvaiko, J. Euzenat, Ontology matching: State of the art and future challenges, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 158–176.
[91] V. Mayer-Schönberger, K. Cukier, Big Data: A Revolution that Will Transform How We Live, Work, and Think, Houghton Mifflin Harcourt, 2013.

Please cite this article in press as: C. Kacfah Emani, et al., Understandable Big Data: A survey, Computer Science Review (2015), http://dx.doi.org/10.1016/j.cosrev.2015.05.002