



## DataSynapse: A Social Data Curation Foundry

Amin Beheshti<sup>1</sup> · Boualem Benatallah<sup>2</sup> · Alireza Tabebordbar<sup>2</sup> ·  
Hamid Reza Motahari-Nezhad<sup>2,3</sup> · Moshe Chai Barukh<sup>2</sup> · Reza Nouri<sup>2</sup>

© Springer Science+Business Media, LLC, part of Springer Nature 2018

### Abstract

Social data analytics have become a vital asset for organizations and governments. For example, over the last few years, governments started to extract knowledge and derive insights from vastly growing open data to personalize the advertisements in elections, improve government services, predict intelligence activities, as well as to improve national security and public health. A key challenge in analyzing social data is to transform the raw data generated by social actors into curated data, i.e., contextualized data and knowledge that is maintained and made available for use by end-users and applications. To address this challenge, we present the notion of knowledge lake, i.e., a contextualized Data Lake, to provide the foundation for big data analytics by automatically curating the raw social data and to prepare them for deriving insights. We present a social data curation foundry, namely DataSynapse, to enable analysts engage with social data to uncover hidden patterns and generate insight. In DataSynapse, we present a scalable algorithm to transform social items (e.g., a Tweet in Twitter) into semantic items, i.e., contextualized and curated items. This algorithm offers customizable *feature extraction* to harness desired features from diverse data sources. To link contextualized information items to the domain knowledge, we present a scalable technique which leverages cross document coreference resolution assisting analysts to derive targeted insights. DataSynapse is offered as an extensible and scalable microservice-based architecture that are publicly available on GitHub supporting networks such as Twitter, Facebook, GooglePlus and LinkedIn. We adopt a typical scenario for analyzing urban social issues from Twitter as it relates to the government budget, to highlight how DataSynapse significantly improves the quality of extracted knowledge compared to the classical curation pipeline (in the absence of feature extraction, enrichment and domain-linking contextualization).

**Keywords** Social networks analytics · Big data analytics · Knowledge lake · Data curation · Feature engineering

---

✉ Amin Beheshti  
amin.beheshti@mq.edu.au

Extended author information available on the last page of the article

## 1 Introduction

Analyzing social data is important as the world, and the businesses that run it, are becoming increasingly social [1]. This has led to a dramatic increase in the amount of data available to us, thus making social data analytics a vital asset for organizations and governments [6,32]. For example, over the last few years, governments started to extract knowledge and derive insights from vastly growing open data to personalize the advertisements in elections [37], improve government services [15], predict intelligence activities [25], unravel human trafficking activities, understand impact of news on stock markets, analysis of financial risks; accelerate scientific discovery, as well as to improve national security and public health [37]. Consequently, acquiring knowledge at large-scale from social data will present prodigious potential over the next few years.

Social data itself - ranging from structured to unstructured data generated by social actors<sup>1</sup>—is useless unless processed in analytical tasks from which humans or downstream applications can derive insights. In this context, a key principal and challenge is to transform the raw data generated by social actors into curated data, i.e., contextualized data and knowledge that is maintained and made available for use by end-users and applications. This process, known as data curation [8,21], can have a significant impact on business operations, especially when it comes to the decision-making processes within organizations. Data curation involves identifying relevant data sources, extracting data and knowledge, cleaning, maintaining, merging, enriching and linking data and knowledge. For example, information extracted from tweets is often enriched with metadata on geolocation, in the absence of which the extracted information would be difficult to interpret and meaningfully utilise. Data curation thus acts as the glue between raw data and analytics, providing an abstraction layer that relieves users from time consuming, tedious and error prone curation tasks.

To facilitate the curation process, in this paper, we present the notion of Knowledge Lake, i.e., a contextualized Data Lake<sup>2</sup> [10,22], to provide the foundation for big data analytics by automatically curating the raw social data and to prepare them for deriving insights. We present a general purpose social data curation foundry, namely DataSynapse. In DataSynapse, we present a scalable algorithm to transform social items (such as a Tweet in Twitter or a post in Facebook) into semantic items, i.e., contextualized and curated items. This algorithm offers customizable *feature extraction* to harness desired features from diverse data sources. To link contextualized information items to the domain knowledge, we present a scalable technique which leverages Cross Document Coreference Resolution assisting analysts to derive targeted insights. DataSynapse provides analysts the foundation and the ability to analyze social data in an explorative manner and the flexibility to answer the questions they care about. The unique contributions of this paper are:

<sup>1</sup> A social actor is a conscious, thinking, individual who has an account a social network such as Facebook (facebook.com) and has the capacity to shape their world in a variety of ways by reflecting on their situation and the choices available to them on social networks.

<sup>2</sup> The notion of a Data Lake has been coined to convey the concept of a centralized repository containing limitless amounts of raw (or minimally curated) data stored in various data islands. The rationale behind a Data Lake is to store raw data and let the data analyst decide how to cook/curate them later.

- We introduce the notion of *Semantic-Item*, to enable customizable feature-based extraction and enrichment and to harness desired features from the raw data. A feature is an attribute or value of interest in the data. We present low-level features as deployable, small and modular services; examples of low-level features in the category of “extraction” include [8]: named entities, keywords, synonyms, stem and part-of-speech. Since APIs are akin to functions applied on features, through the use of APIs, lower-level features can easily be cascaded to produce higher-level features, and so on. For example, low-level features can be used to extract a Named Entity (e.g., a Person such as ‘Malcolm Turnbull’) and enrich it by linking it to an entity (e.g., ‘29th Prime Minister of Australia’) in a knowledge sources such as Wikidata<sup>3</sup> and/or Yago.<sup>4</sup> High-level capabilities are defined as micro-services that perform major data curation tasks such as Extracting, Linking, Merging and Summarizing data.
- We present *Contextualized-Item* as an approach to support linking extracted data to Domain Knowledge thus promoting contextualization of data into knowledge. This technique leverages cross document coreference resolution assisting analysts in computing feature-based summaries and to derive targeted insights. For example, considering an analyst who is interested to gain an accurate and deep understanding of cyber bullying, keyword-based summaries can be used to group social items (e.g., a Tweet) that have mentions of keywords (e.g., having intimidating or threatening nature) and prepare them for deeper analytic tasks such as sentiment analysis of the social item content. The goal is to inject meaning and greatly assists with interpretation of data in a given domain.
- We present a scalable algorithm to transform social items (e.g., a Tweet in Twitter) into contextualized and curated items. This algorithm offers customizable *feature extraction* to harness desired features from diverse data sources.
- We present the notion of knowledge lake, i.e., a contextualized data lake, to provide the foundation for big data analytics by automatically curating the raw social data and to prepare them for deriving insights. The term *Knowledge* here refers to a set of facts, information, and insights extracted from the raw data using data curation techniques such as extraction, linking, summarization, annotation, enrichment, classification and more.
- DataSynapse is offered as an extensible and scalable microservice-based architecture implemented as a set of general purpose social data curation APIs, that are publicly available on GitHub<sup>5</sup> supporting networks such as Twitter, Facebook, GooglePlus and LinkedIn. We evaluate the effectiveness (concerns with achieving a high quality result) and efficiency (concerns performing the proposed approach as fast as possible for large datasets) of DataSynapse as well as the curation services.

DataSynapse can be used in various scenarios such as detecting and/or predicting cyber bullying, fake news and intelligence/terrorism activities as well as improving government service. In this paper, we adopt a typical scenario for analyzing Urban Social Issues from Twitter as it relates to the Government Budget: it is paramount to

---

<sup>3</sup> <https://www.wikidata.org>.

<sup>4</sup> <https://github.com/yago-naga/yago3>.

<sup>5</sup> <https://github.com/unsw-cse-soc/datasynapse>.

stabilize the economy through timely and dynamic adjustment in expenditure plans by considering related social issues such as a problem or conflict raised by society ranging from local to national issues such as health, social security, public safety, welfare support, and domestic violence. We discuss how DataSynapse significantly improves the quality of extracted knowledge compared to the classical curation pipeline (in the absence of feature extraction, enrichment and domain-linking contextualization).

The rest of the paper is organized as follows: In Sect. 2 we provide the background and related work. We present the general purpose curation foundry framework in Sect. 3. In Sect. 4, we present a motivating scenario and experiments. In Sect. 5, we present the implementation and the results of the evaluation of the proposed approach before concluding the paper with remarks for future directions in Sect. 6.

## 2 Background and related work

Social networks have been studied fairly extensively in the general context of analyzing interactions between people, and determining the important structural patterns in such interactions [1]. One of the main challenges in this domain is to transform social data into actionable insights. To achieve this goal it will be vital to prepare and curate the raw social data for analytics. Data curation has been defined as the active and ongoing management of data through its lifecycle of interest and usefulness [21]. Data curation includes all the processes needed for principled and controlled data creation, maintenance, and management, together with the capacity to add value to data [7]. In this paper, we primarily aim at data creation and value generation, rather than maintenance and management of this data over time. More specifically, we focus on curation tasks that transform raw social data (e.g., a Tweet in Twitter) into contextualized data and knowledge include extracting, enriching, linking, annotating and summarizing social data.

The contributions of this paper aim to advance the field of social data curation: The process of breathing meaning into raw data generated on social networks and transforming it to contextualized knowledge, for effective consumption in social analytics and insight discovery. For example, information extracted from tweets is often enriched with metadata on geolocation, in the absence of which the extracted information would be difficult to interpret and meaningfully utilize. Data curation thus acts as the glue between raw data and analytics, providing an abstraction layer that relieves users from time consuming, tedious and error prone curation tasks. Current approaches in data curation rely mostly on data processing and analysis algorithms including machine learning-based algorithms for information extraction, item classification, record-linkage, clustering, and sampling [14]. For example, these algorithms can be used to extract named entities from tweets (e.g., ‘ISIS’ and ‘Palmyra’ in “There are 1800 ISIS terrorists in Palmyra, only 300 are Syrians”), link them to entities in a knowledge base (e.g., Wikidata), and classify tweets into a set of predefined topics (e.g., using Naive Bayes classifier). These algorithms are certainly the core components of data-curation platforms, where high-level curation tasks may require a non-trivial combination of several algorithms [2]; e.g., IBM Watson question-answering system use hundreds of various algorithms for producing an answer [20].

Modern data analytic platforms provide scripting-based, rule and query-based languages for describing data curation pipelines over data sources [18]. Examples of these languages in academia include: data extraction language (DEL<sup>6</sup>), as well as AQL [16] (query-based Information Extraction language). Examples of these languages from Industry include Google Cloud Dataflow, Amazon Kinesis, and eBay QL.io.<sup>7</sup> General-purpose languages include: R, Scala, Python and their extensions. Rule-based languages use regular expressions, dictionaries and taxonomies to specify user-defined classification, information extraction, and entity-matching tasks [28].

Set of related work [5,29,33,39,40] on Semantic Web and semantically oriented data warehousing solutions aim to establish a semantic layer on top of heterogenous metadata, where the goal is to breathe meaning into information extracted from raw data. Many of these approaches focused on creating, enriching or reusing Knowledge Graphs (KGs), i.e., large knowledge-bases that contain a wealth of information about entities (e.g., millions of people, organizations, places, topics, events) and their relationships. Existing KGs [35] (e.g., Wikidata,<sup>8</sup> YAGO,<sup>9</sup> DBpedia,<sup>10</sup> KnowItAll,<sup>11</sup> BabelNet,<sup>12</sup> ConceptNet<sup>13</sup> and DeepDive,<sup>14</sup> include both manually and automatically curated knowledge bases. These approaches can be used in enriching/annotating the raw data and therefore would be a great asset for the curation pipeline.

Another line of related work [34,41], focus on data integration, especially in ETL (Extract, Transform and Load) systems, data federators, data cleaning, schema integration and entity deduplication. However, there has been little work on collecting all of the curation components into an extensible and scalable curation system. For example, Apache UIMA<sup>15</sup> facilitates the analysis of unstructured content but does not consider the domain knowledge to contextualize the extracted information. Finally, there has been considerable work on curating open data. These works provide domain specific solutions for different curation tasks, including leveraging crowdsourcing techniques to extract keywords from tweets in Twitter [13,27], Named entity recognition in tweets [30], linking entities for enriching and structuring social media content [38], and sentiment analysis and identifying mental health cases on Facebook [31]. However, to the best of our knowledge, there has been no work on presenting a general purpose approach that can be used for curating open data: this will enable the analysts to link the contextualized data and knowledge generated on different social networks, uncover hidden patterns and generate insight.

---

<sup>6</sup> <https://www.w3.org/TR/data-extraction>.

<sup>7</sup> <https://github.com/ql-io>.

<sup>8</sup> <https://www.wikidata.org/>.

<sup>9</sup> <https://github.com/yago-naga/yago3>.

<sup>10</sup> <http://wiki.dbpedia.org/>.

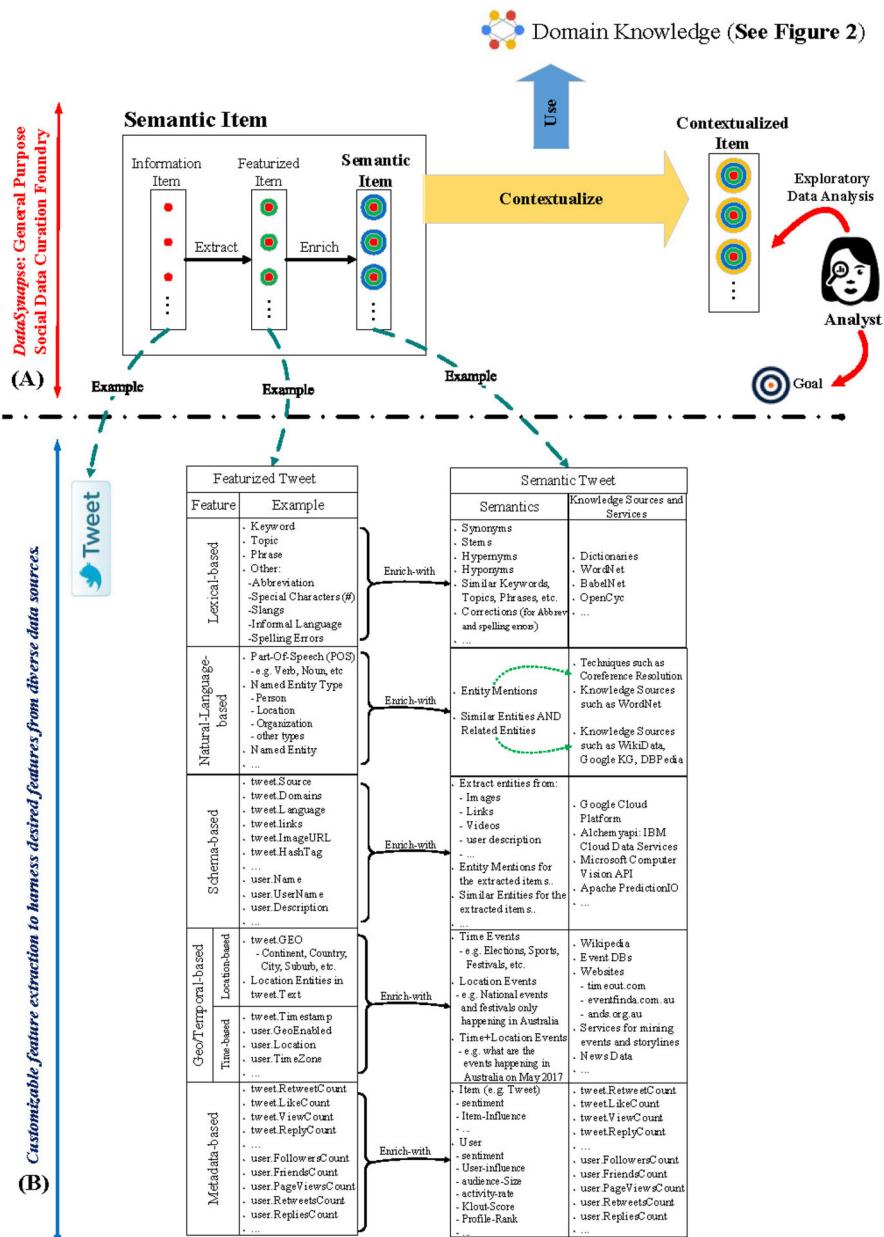
<sup>11</sup> <http://projectsweb.cs.washington.edu/research/knowitall/>.

<sup>12</sup> <http://babelnet.org/>.

<sup>13</sup> <http://conceptnet.io/>.

<sup>14</sup> <http://deepdive.stanford.edu/>.

<sup>15</sup> <https://uima.apache.org/>.



**Fig. 1** The general purpose social data curation foundry framework (a), and examples of customizable feature extraction to harness desired features from diverse data sources

### 3 General purpose social data curation foundry

Figure 1a illustrates the general purpose social data curation foundry framework. In the following we explain the main elements of the DataSynapse framework.

### 3.1 Information-, featurized-, and semantic-items

Social networks enables users to communicate with each other by posting information, comments, messages, images and more. Examples of social networks include Twitter,<sup>16</sup> Facebook,<sup>17</sup> GooglePlus<sup>18</sup> and LinkedIn.<sup>19</sup> Two main elements of a social network are an *Information-Item* (e.g., a tweet in Twitter or a post in Facebook) and a *Social Actor* (e.g., a person/organization who has an account on Twitter/Facebook). An Information Item in a social network is raw and need to be processed for use. In this context, the first step is to identify and extract features from such raw data. To achieve this, we propose data curation feature engineering: this refers to characterizing variables that grasp and encode information, thereby enabling to derive meaningful inferences from data. An example of a feature is ‘mentions of a Person in tweets or other social media posts’ [2]. We introduce the notions of *Featurized-Item* and *Semantic-Item* to enable customizable feature-based extraction and enrichment, to harness desired features from the raw data and to empower a degree of agility and customization.

**Definition** An **Information-Item** represented as a data object that exists separately, has a unique identity and can be described with an attribute vector  $[Item_{id}, Item_{type}, Item_{schema}]$  where,  $Item_{id}$  is a mandatory attribute whose value represents the unique identity of the social item;  $Item_{type}$  is a mandatory attribute whose value indicate the type of the social network this item is related to (e.g., ‘Twitter’, ‘Facebook’ or ‘GooglePlus’); and  $Item_{schema}$  is a mandatory attribute whose value represents the schema of the *Item*. Examples of Information-Item can be a Tweet in Twitter or a Post in Facebook. Figure 6, in Sect. 4, illustrates a sample Twitter Schema. For example, the schema of the Tweet presents attributes such as ‘Tweet.Text’, ‘Tweet.Source’, ‘Tweet.Geo’, and ‘Tweet.Hashtags’. Details about Information-Item schemas can be found in [8].

**Definition** A **Featurized-Item** *FI* is a high-level entity and can be described with an attribute vector  $[FI_{id}, Item, Feature-Set]$  where,  $FI_{id}$  is a mandatory attribute whose value represents the unique identity of the Featurized-Item; *Item* is an Information-Item; and *Feature – Set* is a set of features extracted from an *Item*, including:

- *Schema-based features*. This category is related to the properties of a social item. For example, according to the Twitter schema, a tweet may have attributes such as text, source and language; and a user may have attributes such as username, description and timezone.
- *Lexical-based features*. This category is related to the words or vocabulary of a language such as keyword, topic, phrase, abbreviation, special characters (e.g., ‘#’ in a tweet), slangs, informal language and spelling errors.
- *Natural-Language-based features*. This category is related to entities that can be extracted by the analysis and synthesis of natural language (NL) and speech;

<sup>16</sup> <https://twitter.com/>.

<sup>17</sup> <https://www.facebook.com/>.

<sup>18</sup> <https://plus.google.com>.

<sup>19</sup> <https://www.linkedin.com/>.

- such as part-of-speech (e.g., verb, noun, etc), named entity type (e.g., person, organization, product, etc), and named entity (i.e., an instance of an entity type such as ‘Malcolm Turnbull’ as an instance of entity type Person).
- *Time-based features.* This category is related to the mentions of time in the schema of the item (e.g., ‘tweet.Timestamp’ and ‘user.TimeZone’ in Twitter) or in the content of the social media posts (e.g., in Twitter the text of a tweet may contain ‘3 May 2017’).
  - *Location-based features.* This category is related to the mentions of locations in the schema of the item (e.g., in Twitter ‘tweet.GEO’ and ‘user.Location’) or in the content of the social media posts (e.g., in Twitter the text of a tweet may contain ‘Sydney’; a city in Australia).
  - *Metadata-based features.* This category is related to a set of data that describes and gives information about the social items and actors. For example, it is important to know the number of followers (followersCount) and friends (friendsCount) of a social actor, the number of times a social item has been viewed (viewCount), liked (likeCount) or shared(sharedCount); or the sentiment [26] of the content posted on a social network.

We present low-level features as deployable, small and modular services. Examples of low-level features in the category of “extraction” include [8]: named entities, keywords, synonyms, stem and part-of-speech. Since APIs are akin to functions applied on features, through the use of APIs, lower-level features can easily be cascaded to produce higher-level features, and so on. For example, low-level features can be used to extract a Named Entity (e.g., a Person such as ‘Malcolm Turnbull’) and enrich it by linking it to an entity (e.g., ‘29th Prime Minister of Australia’) in a knowledge sources such as Wikidata and Yago. High-level capabilities are defined as micro-services that perform major data curation tasks such as extracting, linking, merging and summarizing data. Figure 1b illustrates a list of extraction and enrichment features.

**Definition** A **Semantic-Item**  $SI$  is a high-level entity and can be described with an attribute vector [ $FI_{id}$ , *Enrichment-Set*] where,  $FI_{id}$  is the unique identity of the Featurized-Item; and *Enrichment – Set* is a set of annotations used to enrich the features extracted from an *Item*. We define a set of enrichment functions to enrich the extracted items. For example, if a tweet contains a keyword ‘Health’ (extracted using the ‘Keyword’ feature), the enrichment function ‘Synonym’ can be used to enrich this keyword with its synonyms (e.g., ‘well-being’ and ‘haleness’ using knowledge sources such as Wikidata). The result (e.g., set of synonyms) will be stored in the *Enrichment – Set*. The proposed enrichment functions include:

- *Schema-based Semantics.* We use knowledge services such as Google Cloud Platform,<sup>20</sup> Alchemyapi,<sup>21</sup> Microsoft Computer Vision API<sup>22</sup> and Apache PredictionIO<sup>23</sup> to extract various features from the social items properties. For example,

---

<sup>20</sup> <https://cloud.google.com/>.

<sup>21</sup> <https://www.ibm.com/watson/alchemy-api.html>.

<sup>22</sup> <https://azure.microsoft.com/en-gb/services/>.

<sup>23</sup> <https://github.com/PredictionIO/>.

if a tweet in Twitter contains an Image, it is possible to extract entities (e.g., people and objects) from the image.

- *Lexical-based Semantics*. We leverage knowledge sources such as dictionaries and WordNet,<sup>24</sup> to enrich Lexical-based features with their Synonyms, Stems, Hypernyms<sup>25</sup> Hyponyms<sup>26</sup> and more.
- *NL-based Semantics*. We leverage knowledge sources such as WikiData, Google-KG<sup>27</sup> and DBpedia<sup>28</sup> to enrich Natural-Language-based features with similar and related entities. For example, ‘Malcolm Turnbull’<sup>29</sup> is similar to ‘Tony Abbott’<sup>30</sup> (they both acted as the prime minister of Australia) but ‘Malcolm Turnbull’ is related to ‘University of Sydney’<sup>31</sup> (the University where he attended and graduated). We also use techniques such as Coreference Resolution [9] to enrich named entities with their mentions. For example, ‘Malcolm Turnbull’ is a named entity of type person whose entity mentions include ‘Malcolm Bligh Turnbull’, ‘Malcolm B. Turnbull’, ‘M. Turnbull’, ‘29th Prime Minister of Australia’ and more.
- *Geo/Temporal-based Semantics*. We leverage knowledge sources such as Wikidata and services (such as events and storyline mining) to enrich time-/location-based features with time and location events. For example, if a tweet posted from Australia we enrich it with all the events in that location. If a tweet posted on, for example, ‘3 May 2017’ we enrich it with all the events happening around that time frame. For example, if a tweet posted on ‘3 May 2017’ from any location within Australia, we enrich the tweet to be related to ‘Australian Budget’ as we know from knowledge sources that the Australian Treasurer handing the Budget on 3 May every year.
- *Metadata-based Semantics*. We use metadata-based features (such as followerCount and ShareCount) to calculate semantics such as the *influence* of an item or a social actor. These semantics will enable the analysts to get more insight from the social media posts and analyze the capacity to have an effect on the character, development, or behavior of other social users (e.g., in analyzing cases for social network recruitment and radicalization).

Identifying and writing features can be extremely time consuming and tedious, especially those that are executed over very large datasets [2]. We have designed a set of APIs to perform raw data transformation via features. We propose that features will be implemented and available as uniformly accessible data curation Micro-Services: functions or pipelines implementing features. Since an API-based approach treats features as functions, lower-level features can easily be cascaded to produce higher-

---

<sup>24</sup> <https://wordnet.princeton.edu/>.

<sup>25</sup> A Hypernym is a word with a broad meaning constituting a category into which words with more specific meanings fall; a superordinate. For example, colour is a hypernym of red.

<sup>26</sup> A Hyponym is a word of more specific meaning than a general or superordinate term applicable to it. For example, spoon is a hyponym of cutlery.

<sup>27</sup> <https://developers.google.com/knowledge-graph/>.

<sup>28</sup> <http://wiki.dbpedia.org/>.

<sup>29</sup> [https://en.wikipedia.org/wiki/Malcolm\\_Turnbull](https://en.wikipedia.org/wiki/Malcolm_Turnbull).

<sup>30</sup> [https://en.wikipedia.org/wiki/Tony\\_Abbott](https://en.wikipedia.org/wiki/Tony_Abbott).

<sup>31</sup> [https://en.wikipedia.org/wiki/University\\_of\\_Sydney](https://en.wikipedia.org/wiki/University_of_Sydney).

level features, and so on. For example, to identify tweets of a *positive sentiment* that relates to the 29th Prime Minister of Australia (Malcolm Turnbull), we would craft a high-level feature that combines sentiment analysis (Metadata-based feature) with named entities (Natural-Language-based feature). Figure 1b illustrates examples of these features and their categories.

### 3.2 Contextualized-item

Extracted data may be interpreted in many different ways. To make sense of this for a given context, it is highly beneficial that such data is *linked* with *domain knowledge* in order to produce *contextualized knowledge*. This can be supported by building a domain-specific knowledge base that provides a rich structure of relevant entities, their semantics, and relationships.

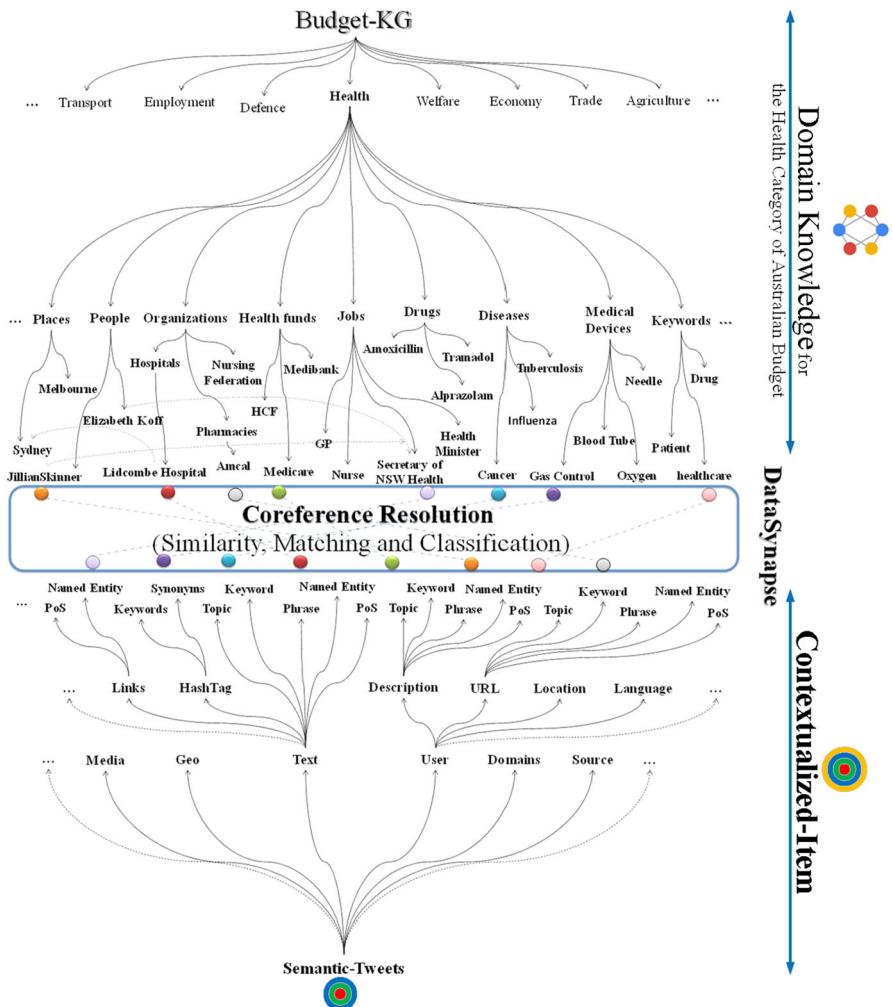
**Definition A Domain Knowledge**  $DK$  is a knowledge base that consists of a set of concepts organized into a taxonomy, instances for each concept and relationships among the concepts. Figure 2 illustrates a sample domain knowledge for the health category of Australian budget: details about this scenario and how we construct this domain knowledge can be found in Sect. 4.

**Definition** We present **Contextualized-Item** as an approach to support linking extracted data to Domain Knowledge thus promoting contextualization of data into knowledge. A Contextualized-Item  $CI$  is a high-level entity and can be described with an attribute vector  $[SI_{id}, DK, Linking-Set]$  where,  $SI_{id}$  is the unique identity of the Semantic-Item;  $DK$  is the domain knowledge consists of a set of entities (where  $DK_{entity(X)}$  is an entity in the Domain Knowledge, e.g., ‘Malcolm Turnbull’ which is an instance of the concept ‘Person’); and  $Linking - Set$  is a set of related pairs  $[Item_j, DK_{entity(Y)}]$ , where  $Item_j$  is the  $j^{th}$  extracted item in the Semantic-Item and  $DK_{entity(Y)}$  is an entity (uniquely identified as ‘Y’) in the domain knowledge.

### 3.3 Linking contextualized-item to the domain knowledge

In our previous work [9], we provided a systematic review and comparative analysis of cross-document coreference resolution (CDCR) methods and tools. As a novel application of this process, we propose to leverage the Cross-Document Coreference Resolution process to facilitate linking extracted data in the Semantic-Item to the entities in the domain knowledge.

**Definition** We present **CDCR-Similarity** as an approach to automatically link extracted data in the Semantic-Item (i.e., Source Entity) to the entities in the domain knowledge (i.e., Target Entity). If the overlap between the Source Entity and Target Entity was statistically significant, the two entities are deemed to be functionally similar and the relationship ‘Similar-To’ will be automatically constructed between them. For a given Source Entity  $S_E$  and Target Entity  $T_E$ , this method calculates a score to measure the correlation of the given similarity among the lexical-based, NL-based,



**Fig. 2** A typical scenario for analyzing urban social issues from Twitter as it relates to the government budget, to highlight how DataSynapse transform a semantic item into contextualized item by leveraging a domain knowledge

schema-based, time-based, location-based and metadata-based features (Fig. 1), as follows:

*Step 1:* Given the feature  $f$  (e.g., Lexical-based features such as keyword or topic) in a feature set  $F$  (i.e., Lexical-based, NL-based, Scheme-based, Time-based, Location-based and Metadata-based features), the semantic similarity between the Semantic-Item Source Entity  $S_E$  and the Target Entity  $T_E$  in the Domain Knowledge is calculated as:

$$\text{Sim}(S_E, T_E) = \frac{\sum_{f \in F_{S_E} \cap F_{T_E}} F(S_E, f) + \sum_{f \in F_{S_E} \cap F_{T_E}} F(T_E, f)}{SV(S_E) + SV(T_E)} \quad (1)$$

where  $SV(S_E)$  denotes the semantic value of the source entity  $S_E$  and  $SV(T_E)$  denotes the semantic value of the target Entity ( $T_E$ ).

*Step 2:* The similarity among the features of Source Entity  $S_E$  and Target Entity  $T_E$  is calculated from Source Entity feature set  $F(S_E)$  with the Target Entity feature set  $F(T_E)$ . For example, if  $F(S_E) = \{\text{Keyword}(\text{'health'})\}$  in Lexical-based feature (KL), Topic('budget') in Lexical-based feature (TL) and  $F(T_E) = \{\text{Phrase}(\text{'Disease marketing'})\}$  in Lexical-based feature (PL), Topic('budget') in Lexical-based feature (TL)}, then the similarity of a Source Entity from feature set  $F(S_E)$  with a Target Entity feature set  $F(T_E)$  is denoted as:

$$\begin{aligned} Sim[KL, F(T_E)] &= max(Sim(KL, PL), Sim(KL, TL)) \\ Sim(TL, F(T_E)) &= max(Sim(TL, PL), Sim(TL, TL)) \\ Sim(CL, F(S_E)) &= max(Sim(PL, TL), Sim(PL, KL)) \\ Sim(TL, F(S_E)) &= max(Sim(TL, KL), Sim(TL, TL)) \end{aligned}$$

*Step 3:* The similarity of two feature sets  $F(S_E)$  and  $F(T_E)$  is calculated as:

$$FSim(F(S_E), F(T_E)) = \frac{\sum_{1 \leq i \leq |F(S_E)|} S(f_i, F(T_E)) + \sum_{1 \leq j \leq |F(T_E)|} S(f_j, F(S_E))}{|F(S_E)| + |F(T_E)|} \quad (2)$$

In our approach we calculate the functional similarity between each pair of features to construct a functional similarity network. Suppose the feature sets  $F(S_E) = \{f_1, f_2, f_3\}$  and  $F(T_E) = \{f_2, f_3, f_4, f_5\}$  contained features that were similar between the source and target entities. The combined feature set of source and target features is denoted as  $F_{S_E, T_E} = \{f_1, f_2, f_3, f_4, f_5\}$ . If  $FSV$  is the functional similarity vector of Source Entity features with respect to  $F_{S_E, T_E}$ , the  $i^{th}$  element of  $FSV$  is denoted as:

$$FSV(i) = \begin{cases} FSim(S_E, f_i) & \text{for } FSim(S_E, f_i) \geq 0 \\ 0 & \text{for Otherwise} \end{cases} \quad (3)$$

Then, the cosine distance, i.e., a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them, between the two vectors  $FSV_{Source}$  and  $FSV_{Target}$  is denoted as  $Cosim(S_E, T_E)$ , and the average cosine distance between Source Entity features and each element in a target feature set  $F(T_E)$  acted as the second part of the correlation score and denoted as:

$$\frac{\sum_{i=1}^m Cosim[FSV_{Source}, FSV_{Target}(i)]}{m} \quad (4)$$

Then, the relevance score of the source and target entity is denoted as:

$$\begin{aligned} Relevance - Score &= \frac{\sum_{i=1}^m FSim[FSV_{Source}, FSV_{Target}(i)]}{m} + W \\ &\times \frac{\sum_{i=1}^m Cosim[FSV_{Source}, FSV_{Target}(i)]}{m} \end{aligned} \quad (5)$$

where  $W$  is the weight (in the range 0.1–1) of the  $Cosim$  value. The correlation score was calculated as the sum of similarity between a feature and some of the most similar neighbours that are associated with the feature set. Figure 3 illustrates the steps in computing the scalable CDCR-similarity.

*Scalable similarity.* To provide a scalable approach, we divide the *CDCR-Similarity* process into several stages and assign each stage into a specific MapReduce<sup>32</sup> [17] (MR) job (Fig. 4). As illustrated in Algorithm 1, in the first MR job, we preprocess the information item based on the social network schema. The schema for various social networks is available on the DataSynapse project GitHub page.<sup>33</sup> After this phase, we use the curation micro-services to generate the featurized-items, semantic-item and contextualized-items respectively. In the final MR job, we generate the (cross-document) coreference entities and classify them into related summaries. Linking information items (e.g., tweets) to the objects in the domain knowledge greatly assists with interpretation of data in a given domain. An example application would be to generate feature-based summaries. For example, keyword-based summaries (in the category of Lexical-based features) may contain all the social items that contain a specific keyword. As another example, person-based summaries (in the category of NL-based features) may contain all the items that contain a specific person (e.g., ‘Malcolm Turnbull’, an instance of type person). In Sect. 4, we will demonstrate how feature-based summaries can assist analysts in analyzing social networks.

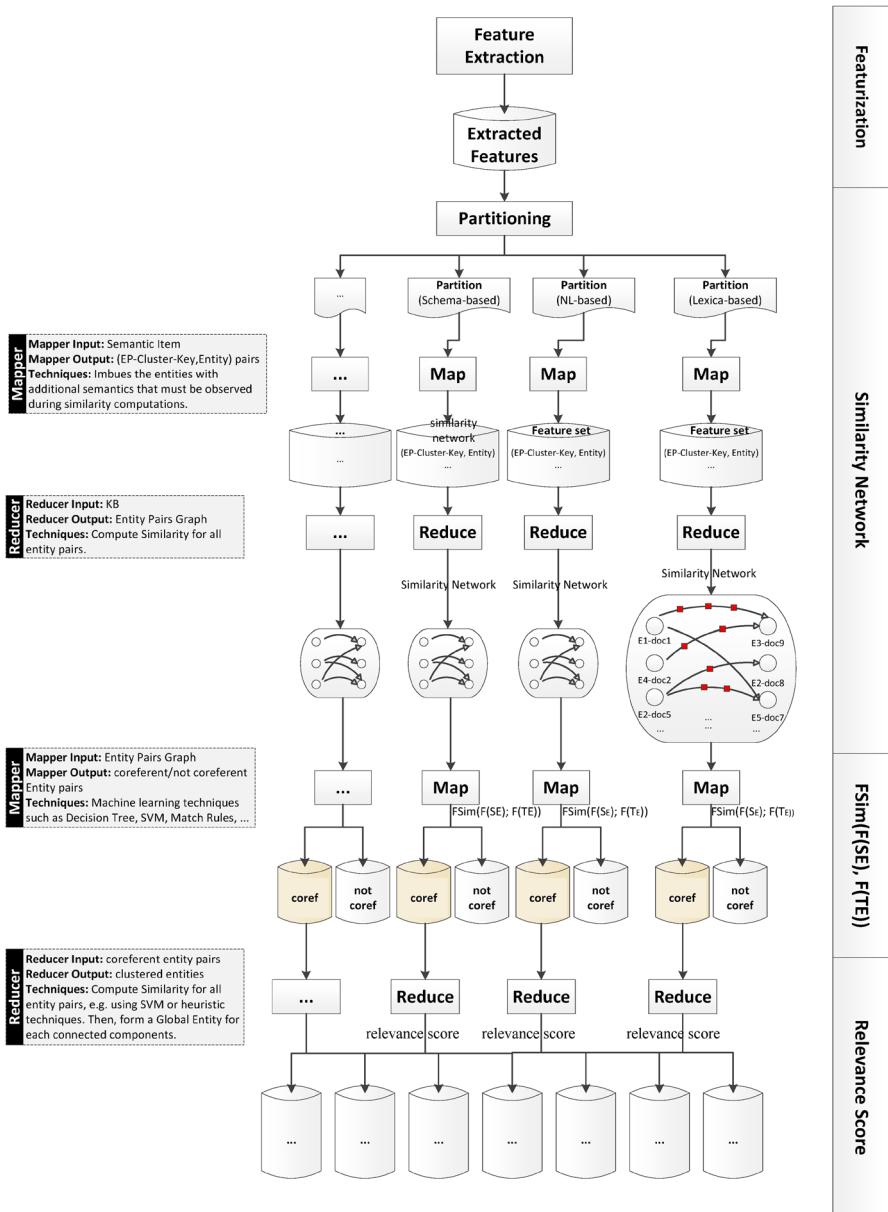
```

Data: Information-Item
Result: Contextualized-Item
MR Job1: Retrieve Information-Item Schema and Pre-Process;
MR Job2: Construct Feature-Set: for Each Information-Item do
    | Extract-Feature(Schema-based, Lexical-based, Natural-Language-based, Time-based,
    | Location-based, Metadata-based features);
end
MR Job3: Construct Enrichment-Set: for Each Item in Feature-Set do
    | Annotate Item with Schema-based, Lexical-based, NL-based, Geo/Temporal-based and
    | Metadata-based Features
end
MR Job4: Construct Linking-Set: for Each Item in Enrichment-Set do
    | for Each Entity in Domain-Knowledge do
        | Compute Similarity;
        | if Coreference Decision then
        |   | Link Item to Entity;
        | else
        |   | Annotate the Item and Entity;
        | end
        | Update Training-Data;
    | end
end
MR Job5: Generate Summaries;
```

**Algorithm 1:** General purpose social data curation algorithm.

<sup>32</sup> MapReduce ([hadoop.apache.org/](http://hadoop.apache.org/)) is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

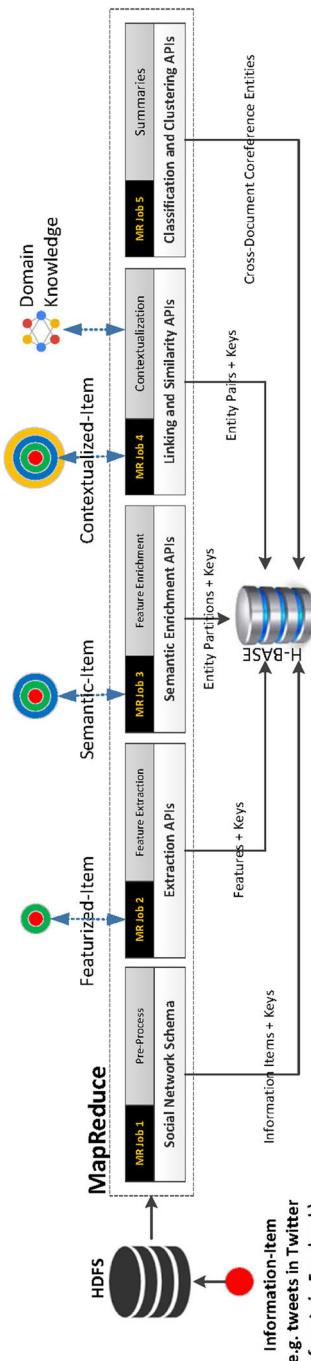
<sup>33</sup> <https://github.com/unsw-cse-soc/datasynapse>.



**Fig. 3** Scalable CDR-similarity process

### 3.4 Knowledge lake

Organizing vast amount of social data gathered from various data islands, i.e., *Data Lake* [10], will facilitate dealing with a collection of independently-managed datasets



**Fig. 4** Dividing a CDGR-similarity process into several stages and assign each stage into a specific MapReduce job

such as Twitter, Facebook, LinkedIn and GooglePlus. The notion of a Data Lake has been coined to address this challenge and to convey the concept of a centralized repository containing limitless amounts of raw (or minimally curated) data stored in various data islands. The rationale behind the Data Lake is to store raw data and let the data analyst decide how to cook/curate them later.

In this paper, we introduce the notion of *Knowledge Lake*, i.e., a contextualized Data Lake, as a centralized repository containing virtually inexhaustible amounts of both data and *contextualized data* that is readily made available anytime to anyone authorized to perform analytical activities. The term *Knowledge* here refers to a set of facts, information, and insights extracted from the raw social data using data curation techniques used to transfer an Information-Item into a Featurized-, Semantic- and Contextualized-Items. The Knowledge Lake will provide the foundation for big data analytics by automatically curating the raw data in the Data Lake and to prepare them for deriving insights.

For the raw information items stored in the Lake, we provide services to automatically: (a) extract features such as keyword, part-of-speech, and named entities such as persons, locations, organizations, companies, products and more; (b) enrich the extracted features by providing synonyms and stems leveraging lexical knowledge bases for the English language such as WordNet; (c) Link the extracted enriched features to external knowledge bases (such as Google Knowledge Graph<sup>34</sup> and Wikidata<sup>35</sup>) as well as the contextualized data islands; and (d) Annotate the items in a data island by information about the similarity among the extracted information items, classifying and categorizing items into various types, forms or any other distinct class.

On top of the Knowledge Lake, we provide a single service which enables the analyst querying the raw data (in the data lake) as well as the contextualized items in the knowledge lake using full-text search, SQL and SPARQL. We exposes the power of Elasticsearch without the operational burden of managing it by developers and supports querying both raw data and the meta-data generated during the curation process. We enables the power of standard SQL with full ACID transaction capabilities to support a federated query layer on top of the large amount of curated data in the Knowledge Lake held in relational/NoSQL databases. We model the contextualized data and knowledge as a graph of typed nodes (e.g., raw data and extracted features) and edges (relationships among items such as ‘keyword–extractedFrom–eMail’, ‘Person–extractedFrom–Tweet’ or ‘item–similarTo–item’). To enable querying this large graph, we leverage our previous work [4], a SPARQL query engine for analyzing large graphs, to organize the data and extracted-enriched-linked features.

Figure 5 illustrates the architecture and the main components of the Knowledge Lake, namely CoreKG service. Technical details of these services and how we organize and query the data in the Knowledge Lake, can be found in [11].<sup>36</sup> As illustrated in the figure, we not only support organizing and querying the data in the Knowledge Lake, but also we provide a database security protection mechanism including authentication, access control and data encryption for both data and the contextualized data (security

<sup>34</sup> <https://developers.google.com/knowledge-graph/>.

<sup>35</sup> <https://www.wikidata.org/>.

<sup>36</sup> <https://github.com/unsw-cse-soc/CoreKG>.

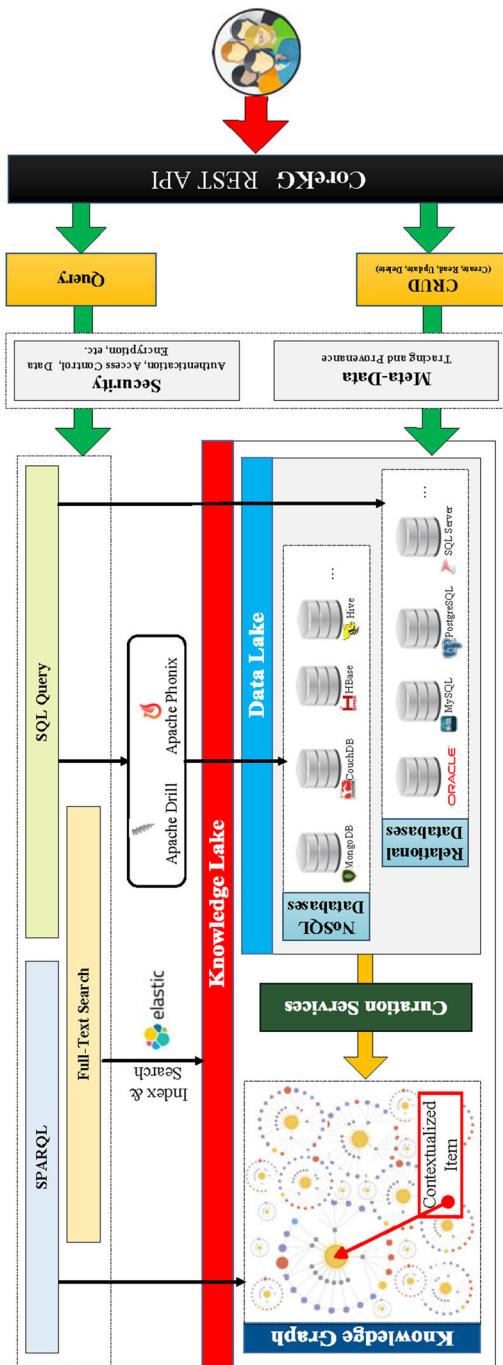


Fig. 5 Knowledge lake architecture [11]

and access control); as well as collecting and aggregating tracing metadata (including descriptive, administrative and temporal metadata and build a provenance [3] graph) for both data and the contextualized data. As an ongoing work [12], we are working on novel techniques for intelligent narrative discovery and to summarize the large amount of curated data in the Knowledge Lake.

### 3.5 Microservice-based architecture

Data curation activities are heavily dependent on the challenges of scale. To address this challenge, we present an *extensible* and *scalable* microservice-based architecture, to enable applications to be decomposed into components and provides capabilities to wrap components as network services. This will enable dealing with large amount of open data and can scale to very large volumes by replicating processing pipelines over a cluster of networked nodes. To achieve this goal we have leveraged Apache UIMA<sup>37</sup> to support the reuse and composition of independently-developed micro-services, for example keyword extraction, named entity detection, similarity comparison and entity linking APIs. The frameworks support configuring and running pipelines of curation APIs. We identified and implemented a set of features, implemented as micro-services, to enable analysts in transforming their raw social data (e.g., a tweet) into curated data (i.e., Semantic-Tweet). These services, available as an open source, are able to extract various features (such as Lexical-based, Natural-Language-based, Schema-based, Geo/Temporal-based and Metadata-based features) from an information item, enrich them with semantic features and assist analysts in discovering similarity among the extracted information items, classifying Semantic-Items into various types and indexing them to deal with large amount of data. We use our previous work [10], an open source data lake service,<sup>38</sup> to organize and query the large social data in big data platforms. We design and write UIMA annotators to enable wrapping curation APIs as network services, and to scale to very large volumes by replicating processing pipelines over a cluster of networked nodes.

## 4 Motivating scenario and experiments

Consider the analytics task related to “*understanding a Governments’ Budget in the context of Urban Social Issues*”: A typical government’s budget denote how policy objectives are reconciled and implemented in various categories and programs. In particular, budget categories (e.g., ‘Health’, ‘Social-Services’, ‘transport’ and ‘employment’) defines a hierarchical set of programs (e.g., ‘Medicare Benefits’ in Health, and ‘Aged Care’ in Social-Services). These programs refers to a set of activities or services that meet specific policy objectives of the government [23]. Using traditionally adopted budget systems, it would be difficult to accurately evaluate the governments’ services requirements and performance. For example, it is paramount to stabilize the economy through timely and dynamic adjustment in expenditure plans

<sup>37</sup> <http://uima.apache.org/>.

<sup>38</sup> <https://github.com/unsw-cse-soc/CoreDB>.

by considering related *social issues*. For instance, a problem or conflict raised by society ranging from local to national issues such as health, social security, public safety, welfare support, and domestic violence [23]. Therefore the opportunity to link active social issues (e.g., public opinions harvested from Tweets) to Budget categories will provide the public with increased transparency, and likewise government agencies with realtime insight about how to make decisions (e.g., reshape policies). In the rest of this paper, we focus this scenario on the *Australian Government's Budget: Health* category, and only consider extracting issues from *Twitter*.

In order to facilitate an analytics task for such or any similar scenario, we propose a three-step data curation pipeline, presented as follows:

#### 4.1 Extract: agile features extraction over raw data

The successful transformation of data into knowledge requires going beyond mere extraction and traditional database-like manipulation operators such as filtering, projection and join. Accordingly, we propose data curation feature engineering. Features-based data extraction also empowers a degree of agility and customization. Prebuilt algorithms become easily outdated (e.g., hashtag or keywords rapidly evolve); while curation scripts prove too complex for end-users, domain-experts or data-analysts. We support the notion of both “low-level” features, such as: Extracting, Cleaning, Merging, Summarizing and Classifying data. As well as “higher-level” features, such as in the category of “extraction” alone, these include: Named Entities, Keywords, Synonyms, Stem and Part-of-Speech extraction.

**Use Case 1: extraction on twitter data.** Open data is complex, unstructured and generated at a high rate, resulting in many challenges to understand and analyze the data. For example, consider the Twitter dataset in the motivating scenario. Machine learning techniques can be used to construct the schema for Twitter data [19]. Using this schema, it is possible to write features to extract relevant elements (e.g., Time, Location, Domains and Text of the Tweet). It is also possible to dig deeper and write more higher-level features to extract keywords, named-entities and part-of-speech from the Text (previously extracted) of the Tweet.

Figure 6 illustrates the main artifacts in Twitter along with items that can be extracted from Twitter (e.g., ‘Tweet.Text’ which illustrates the text of the Tweet or ‘Tweet.Text.Keywords’ which illustrates the keywords that can be extracted from the text of the tweet). Extracting all these features will be a great asset to analyze the large number of Tweets. For example, ‘entity summaries’ of tweets containing the same named entity such as a person or organization; and ‘keyword summaries’ of Tweets containing similar keywords. We may then analyze these related tweets to get valuable insights from the Twitter open data, e.g., identifying social issues on Twitter regarding the government budgets. Figure 7 illustrates a semantic Tweet. As illustrated in this figure, in the first step we retrieve the schema of a Tweet and extract a set of attributes such as Text, Source and the location (Geo) where the Tweet has been posted. Then for the unstructured attributes, we use extraction services to extract features such as Part of Speech and Named Entities.

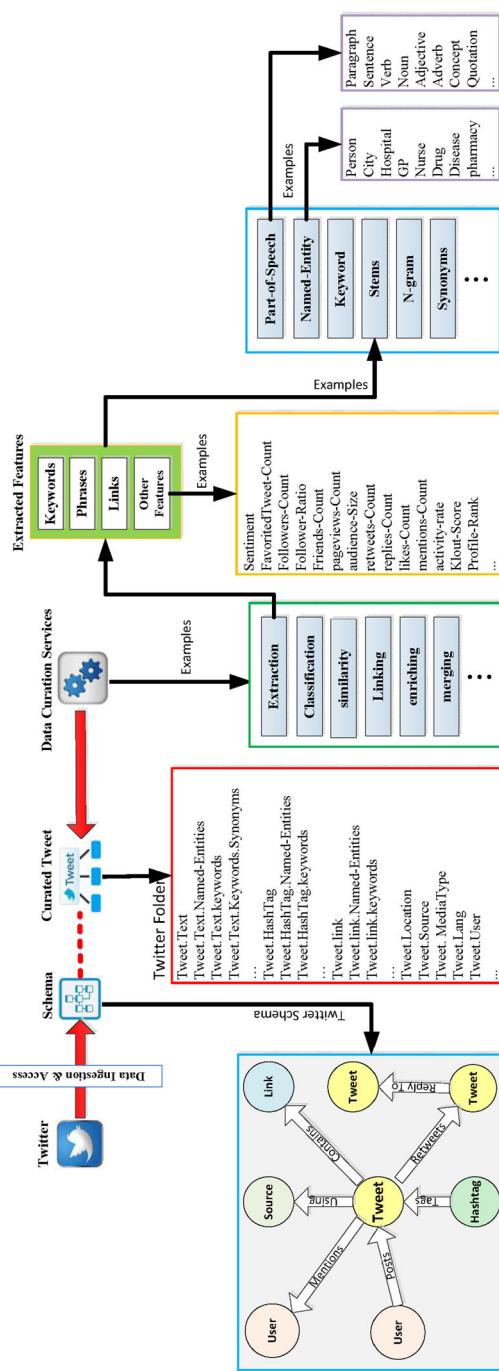
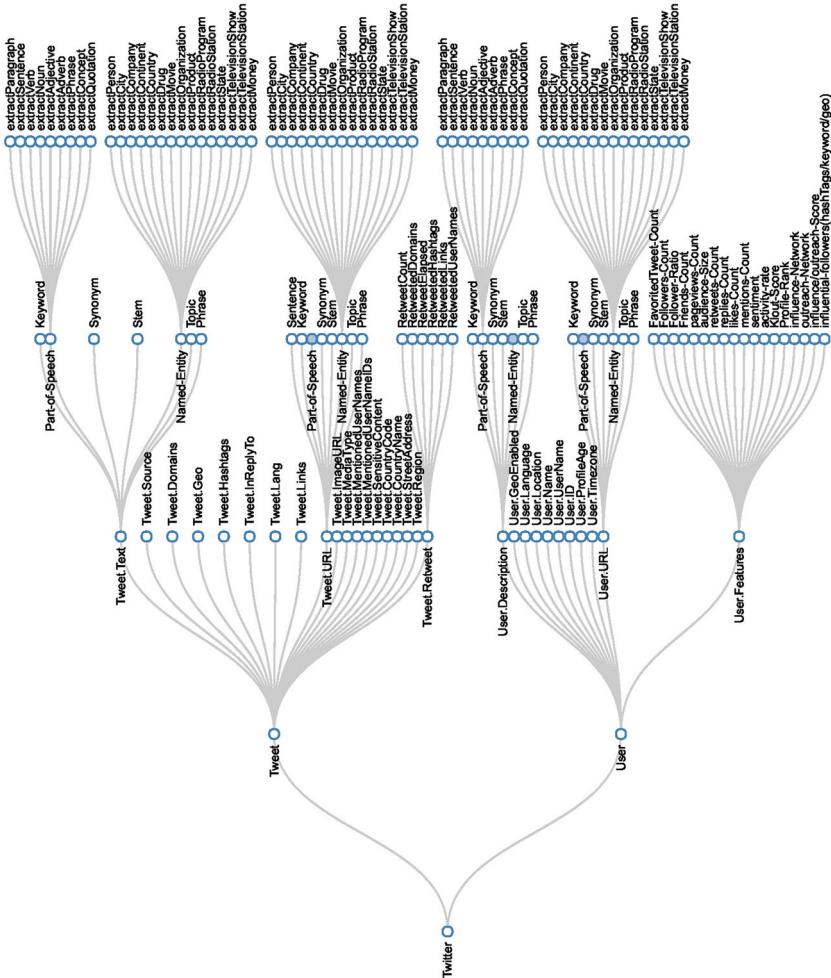


Fig. 6 Feature extraction on Twitter data



**Fig. 7** An overview of a semantic Tweet

## 4.2 Contextualize: linking extracted data with domain-knowledge

Extracted data may be interpreted in many different ways. To make sense of this for a given context, it is highly beneficial that such data is *linked* with *domain knowledge* in order to produce *contextualized knowledge*. We support this by first building a domain-specific knowledge base that provides a rich structure of relevant entities, their semantics, and relationships.

**Use Case 2: building domain-knowledge.** We explain the techniques to construct background domain-knowledge for the Australian Government’s Budget, i.e., Budget-KB. The Budget-KB consists of a set of concepts related to the Australian budget organized into a taxonomy, instances for each concept, and relationships among these concepts. Figure 8 illustrates a sample fragment of the Budget-KB. To build this knowledge-base, we first identified the list of budget categories and their related programs provided by Australian government data services.<sup>39</sup> Then we have focused on the *Health* category in the government budget and identified popular concepts and instances related to this category on the Web. For example, we have identified: (i) people, from GPs and nurses to health ministers and hospital managers; (ii) organizations, such as hospitals, pharmacies and nursing federation; (iii) locations, states, cities and suburbs in Australia; (iv) health funds, such as medibank, bupa and HCF; (v) drugs, such as amoxicillin, tramadol and alprazolam; (vi) diseases, such as cancer, influenza and tuberculosis; (vii) medical devices, such as gas control, blood tube and needle; (viii) job titles, such as GP, nurse, hospital manager, secretary of NSW health and NSW health minister; and (ix) keywords, such as healthcare, patient, virus, vaccine and drug.

To enable Web-based injections, we have implemented APIs to extract instances of concepts: locations from auspost,<sup>40</sup> doctors from Australian doctors directory<sup>41</sup> (including GPs, specialists and nurses), hospitals from myHospitals,<sup>42</sup> health funds from health-services,<sup>43</sup> Drugs from drug-index,<sup>44</sup> Diseases from medicine-net,<sup>45</sup> Medical Devices from FDA,<sup>46</sup> Job titles from compdata,<sup>47</sup> and keywords from Australia national health and medical research council.<sup>48</sup> Then we have implemented APIs to enrich these entities using KBs such as Wikidata,<sup>49</sup> Google Knowledge Graph<sup>50</sup> and Wordnet.<sup>51</sup> For example we extract relationships from Wikidata to form a relationship

<sup>39</sup> <http://data.gov.au/>.

<sup>40</sup> <http://auspost.com.au/postcode/>.

<sup>41</sup> <https://www.ahpra.gov.au/>.

<sup>42</sup> <https://www.myhospitals.gov.au/browse-hospitals/>.

<sup>43</sup> <http://www.privatehealth.gov.au/>.

<sup>44</sup> <http://www.rxlist.com/>.

<sup>45</sup> <http://www.medicinenet.com/>.

<sup>46</sup> <http://www.fda.gov/>.

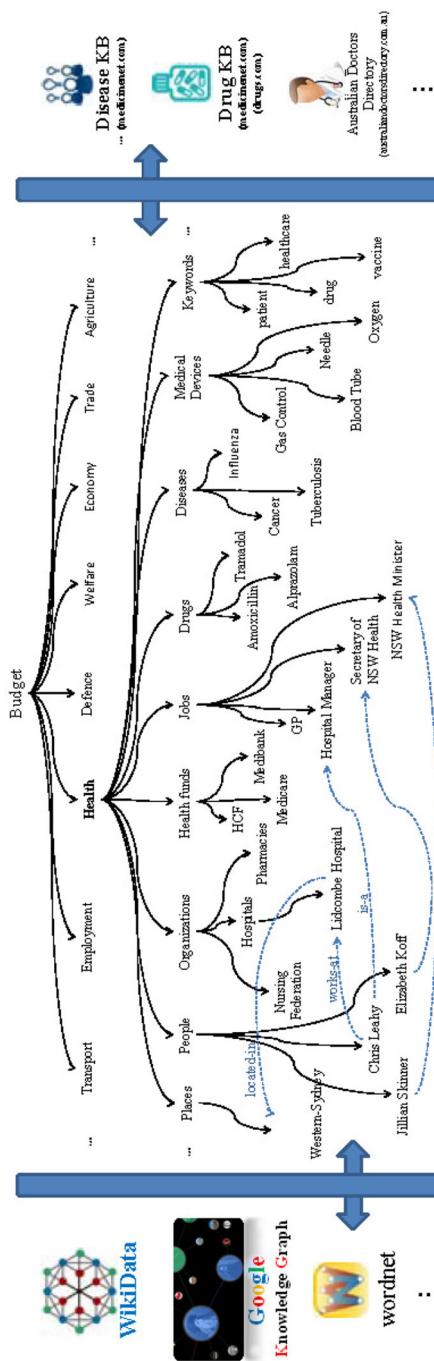
<sup>47</sup> <http://compdatasurveys.com/compensation/healthcare>.

<sup>48</sup> <https://www.nhmrc.gov.au/>.

<sup>49</sup> <https://www.wikidata.org/>.

<sup>50</sup> <https://developers.google.com/knowledge-graph/>.

<sup>51</sup> <https://wordnet.princeton.edu>.



**Fig. 8** A sample fragment of the Budget-KB

graph, e.g., ‘Bankstown Lidcombe Hospital’ located-in ‘Bankstown, Sydney, NSW, Australia’; and we have used Google KG API to link entities to Wikipedia, e.g., by using ‘Jillian Skinner’ as an input we have learned that ‘Jillian Skinner’ is-a ‘person’, linked-to ‘[https://en.wikipedia.org/wiki/Jillian\\_Skinner](https://en.wikipedia.org/wiki/Jillian_Skinner)’; is a member-of ‘New South Wales Legislative Assembly’; and is-a ‘New South Wales Minister for Health’ for Australia. We have also used Wordnet to extract synonyms and hypernyms. Figure 8 shows a small snippet of the formulated KB, which illustrates the above notions. In this KB, for example, ‘Jillian Skinner’ is an instance of the concept Person and is the Health Minister for New South Wales in Australia (see the link between this person and the Job title in Fig. 8). As another example ‘Lidcombe Hospital’ is an instance of a Hospital and is located in Western Sydney (a location, suburb, in NSW Australia).

**Use Case 3: linking extracted twitter data with budget-KG.** So far we have presented how we construct the Budget-KB, and how we leverage the curation APIs to curate Tweets. Next, we explain the method to link tweets to one of several predefined budget categories, e.g., health. In our previous work [9], we provided a systematic review and comparative analysis of cross-document coreference resolution (CDCR) methods and tools. As a novel application of this process, we leverage the CDCR process to link contextualized tweets to government categories and programs. We achieve this by identifying coreferent entities between extracted entities from the tweets and the ones that exist in the Budget-KB knowledge base. More specifically, we identify and implement a set of features to find the similarity among the data objects in Tweets (e.g., named entities that have been extracted from the Text of the Tweet) and the entities in the Budget-KG (e.g., keywords and named entities -such as hospitals, GPs and drugs—related to health). We have designed a similarity API to find similarity not only among strings, numbers and dates; but also among entities (e.g., finding similarity among the attributes and their values), using a wide range of similarity techniques [9] such as dice, cosine, TF-IDF, jaccard, euclidean, city block and levenshtein similarity techniques. For the linking, we go one step further, inferring that for example ‘Bankstown-Lidcombe Hospital’ is related to an item in the local KB (Budget-KG) or an external KB (e.g., Google-KG or a Webpage in Wikipedia<sup>52</sup>).

## 4.3 Analyze

### 4.3.1 Insight discovery

The final step involves deriving relevant *insights* in order to draw decisions. The preceding steps focused on enhanced extraction of raw data, followed by contextualizing that data to a given domain. However, even within a particular, it is important for an effective data curation pipeline to pinpoint the insights that are relevant to a particular *goal*. To address this challenge, we use feature-based summaries to assist analysts deriving insights from the contextualized knowledge, that are most relevant to user’s use-case inquiry. For example, consider an analyst who is interested in identifying tweets on Twitter discussing a social issue related to health. Identifying whether a tweet relates to ‘health’ is largely subjective: it depends on the specific goals of the

<sup>52</sup> [https://en.wikipedia.org/wiki/Bankstown\\_Lidcombe\\_Hospital](https://en.wikipedia.org/wiki/Bankstown_Lidcombe_Hospital).

analysts. For example, consider that the analyst consider a tweet related to health social issue if: tweet contain the keyword ‘Health’ (or any of its Synonyms) or the tweet contains mentions of current Australia’s Health Minister ‘Hon Greg Hunt’ and the tweet express a negative view or opinion (i.e., a negative sentiment). It is also possible to extract other features, e.g., Geo/Temporal-based features, to check the location and the time of the tweet and enrich them with time/location semantics. For example, the analyst may assume the tweet is related to Australian budget if the tweet has mentions of Australia (an instance of type Location) and the tweet is tweeted on or around 3 May 2017 (in Australia, the Treasurer handing down the Budget each year on third of May). We adopted the CDCR process, presented in previous section, to link tweets to instances of entities in the Budget-KG. As the result, the following feature-based summaries have been generated:

- keyword-based summaries (in the category of Lexical-based features): for example the feature *keyword*(‘*health*) can be used to identify tweets that contain mentions of the keyword ‘*health*’.
- Named-entity summaries (in the category of NL-based features): for example the feature *named-entity*(‘*Hon Greg Hunt*’,*Person*) can be used to identify tweets that contain mentions of Australia’s Health Minister Mr. Hunt.
- Negative-Sentiment summaries (in the category of Metadata-based features): for example the feature *Sentiment*(‘*Negative*’) can be used to identify tweets that express a negative opinion.

The conjunction and disjunction of these summaries may enable analysts to build higher level summaries which may contain the union and/or intersection of tweets in the above mentioned lower-level summaries.

### 4.3.2 User-guided insight discovery

Insight discovery deals with finding evidences from the data to enable analysts gain an accurate and deep understanding of their targeted analytic goal. As discussed in previous section, feature-based summaries can facilitate the insight discovery process. For example, considering an analyst who is interested to gain an accurate and deep understanding of cyber bullying, keyword-based summaries can be used to group social items (e.g., a Tweet) that have mentions of keywords (e.g., having intimidating or threatening nature) and prepare them for deeper analytic tasks such as sentiment analysis of the social item content. As an ongoing and future work, and to facilitate using the feature-based summaries, we present a method to assist analysts deriving insights from the contextualized knowledge, that are most relevant to user’s use-case inquiry, using a simple rule language.

Rule-based techniques can provide a declarative supplement for dealing with many of shortcomings inherent in algorithmic approaches. Namely, algorithms are good for solving concrete tasks, yet in reality datasets are large and evolving over time. Using rule-based approaches have several benefits such as [28]: (i) Writing rules are easier than designing algorithms; (ii) Correcting mistake for rules are faster than learning algorithms; and (iii) Rules can consider cases that learning cannot yet cover. As an ongoing work, we are designing a declarative rule language to simplify the analysis

tasks such as classifying, linking, merging, enriching, sampling, and summarization. At a general level a curation rule may be defined as follows:

$$\begin{aligned} < \text{Rule} > ::= & < \text{Dataset} > . < \text{feature} > . \text{feature}(< \text{string|integer|boolean} >) \\ < \text{Rule} > ::= & < \text{Rule} > [\text{AND} | \text{OR} | \text{NOT} < \text{Rule} >], \end{aligned}$$

where a Rule is expressed in terms of operations on features: these features correspond to programmatic-like functions whose determined feature values have common input and output types such as string, integer and boolean. Moreover, composite rules may be composed over one or more other rules allowing for the definition of initial base level rules associated with coarser-grain curation tasks whose output results or data may then form the input for a subsequent rule for finer-grain curation tasks.

**Use Case 4: identifying tweets related to social health issues.** Identifying whether a Tweet relates to ‘health’ is largely subjective: it depends on the specific goals of the analysts. For example, consider Adam, an analyst, who is interested to classify Tweets as health related, based on their Keywords. An example using the proposed language is:

$$\text{Rule} := \text{Twitter.Tweet.Text.contains('Health')}$$

Adam may be also interested to delve into the Tweets’ sentiment, and determine if it is negative:

$$\text{Rule} := \text{Twitter.Tweet.Text.SentimentNegative(true)}$$

Adam may also apply more complex queries by using the conjunction and/or disjunction of the above mentioned examples. For example, he may be interested to classify a Tweet in the ‘Issue’ class if it contains a Keyword related to health and it has a negative sentiment:

$$\begin{aligned} \text{Rule} := & \text{Tweets.keywords.contains('health') AND} \\ & \text{Tweets.sentimentNegative(true)} \end{aligned}$$

In future and ongoing work, we are working on the detailed design and implementation of the proposed language.

## 5 Implementation and evaluation

### 5.1 Implementation

To facilitate the data curation process and enhance the productivity of researchers and developers, we identify a set of features, implemented as micro-services, and make them available as services to researchers and developers to assist them in transforming their raw social data (e.g., a tweet) into curated data (i.e., Semantic-Tweet). These

services enable developers to easily add features - such as extracting keyword, part of speech, and named entities such as Persons, Locations, Organizations, Companies, Products, Diseases, Drugs, etc.; providing synonyms and stems for extracted information items leveraging lexical knowledge bases for the English language such as WordNet; linking extracted entities to external knowledge bases such as Google Knowledge Graph and Wikidata; discovering similarity among the extracted information items, such as calculating similarity between string and numbers; classifying, sorting and categorizing data into various types, forms or any other distinct class; and indexing structured and unstructured data - into their data applications. These services can be accessed via a REST API, and the data is returned as a JSON file that can be integrated into data applications. The basic data curation APIs has been used to implement an extensible and scalable microservice-based architecture implemented as a set of general purpose social data curation APIs, that are publicly available on GitHub<sup>53</sup> supporting networks such as Twitter, Facebook, GooglePlus and LinkedIn.

## 5.2 Experiment

### 5.2.1 System setup

All the experiments were performed on Amazon EC2 platform<sup>54</sup>, Sydney Australia region, using instances running Ubuntu Server 14.04. Having all the machines on a region, minimizes network latency and assures that capacity of machines is nearly identical. For the *effectiveness* experiment (concerns with achieving a high quality result) we have used a single Virtual Machine (VM) of type t2.large that provides 8GB of memory, 2 virtual CPUs and 20GB EBS storage. For the *efficiency* experiment (concerns performing the proposed approach as fast as possible for large datasets), considering the execution time, we have scaled the experiment over three different configurations on Amazon EC2: single machine, four machines and eight machines.

### 5.2.2 Dataset

The Australian Government budget sets out the economic and fiscal outlook for Australia, and shows the Government's social and political priorities. The Treasurer handed down the Budget 2016–17 at 7.30 pm on Tuesday 3 May, 2016. To properly analyze the proposed budget, we have collected all tweets from one month before and two months after this date. In particular, for these three months, we have selected 15 million tweets, persisted and indexed in the MongoDB ([mongodb.com](http://mongodb.com)) database.

### 5.2.3 Evaluation

We evaluated DataSynapse over Twitter data using (*efficiency*, performing the approach as fast as possible for large datasets, and *effectiveness*, achieving a high quality result in terms of precision and recall) metrics. The effectiveness is determined

---

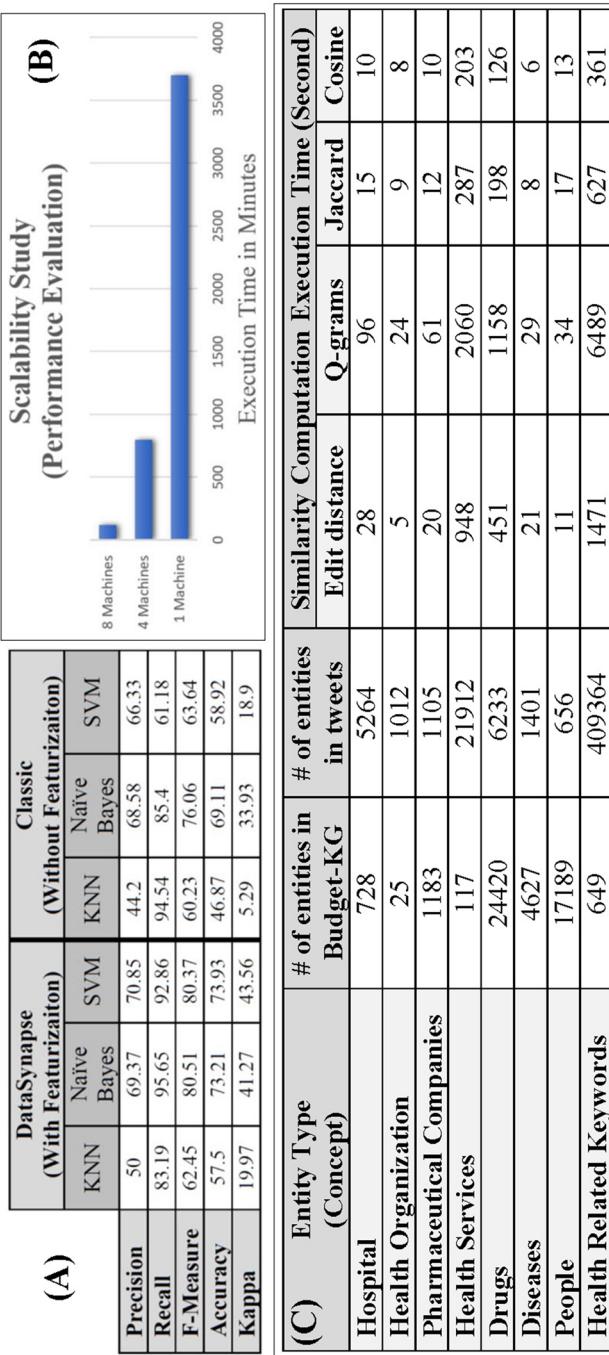
<sup>53</sup> <https://github.com/unsw-cse-soc/datasynapse>.

<sup>54</sup> <https://aws.amazon.com/ec2/>.

with the standard measures precision, recall and F-measure. Precision is the number of correctly identified tweets divided by the total number of tweets, recall is the number of correctly identified tweets divided by the total number of related tweets, and F-measure is the harmonic mean of precision and recall. Let us assume that TP is the number of true positives, FP the number of false positives (wrong results), TN the number of true negatives, and FN the number of false negatives (missing results). Then, Precision =  $\frac{TP}{TP+FP}$ , Recall =  $\frac{TP}{TP+FN}$ , and F-measure =  $\frac{2*Precision*Recall}{Precision+Recall}$ . Experiments concerning the front-end of the system are not reported in this paper and considered as future work.

For the initial evaluation we focus on linking semantic-items (curated tweets) with the domain knowledge (budget-KB). We compared feature attributes' using various similarity functions. For example, we used the following four string similarity functions: edit distance, Q-grams, jaccard, and cosine functions; then, using the similarity API (see Use Case 3), we calculate the *average* similarity score and use it for linking the entities. Figure 9c lists the execution times taken in making decisions by comparing entities (such as hospitals, health organizations, pharmaceutical companies, health services, drugs, diseases and people). We have also considered the Health related keywords listed in the budget-KB. In particular, generating entity pairs and computing similarity among them is a time consuming task and requires high performance computing resources on very large datasets such as Twitter. For example for around 20k entities the algorithm generated around nine million pairs which highlights that pairwise entity comparison will become exponential across tweets. All the functions performed reasonably well in terms of precisions (average percision = 0.845), but they all suffered from low recall (average recall = 0.265) which result in a low F-measure (average F-measure = 0.4), which means they missed many true coreferent entity pairs that should be contained in the returned results. Following we explain that, featurizing and contextualizing the extracted entities (i.e., the Semantic-Item approach) will improve the accuracy of the result.

To further analyse the performance of the approach, we created a set of classifiers using machine learning algorithms to classify tweets relevant to ‘health’. For creating classifiers, we have used two different approaches. First, we used the classic Keyword Matching (KEYM) approach [24], which uses a Bag of Words (BoW) to identify health related Tweets. Next, we have used our proposed featurization technique, which uses variety of features for annotating and extracting Tweets. The goal of this experiment is to identify the performance of the proposed featurization technique in boosting the performance of classifiers in training machine learning models. The approach that better improves the precision and recall would have considered as the successful one. For the proposed featurization technique, we have used three different features: (i) We have used Budget-KB Entity Matching to link an entity in a tweet to the entities in the KB; (ii) We have used Google Knowledge Graph API to indicates the existence of a health related entity in the Google KG with an entity in a tweet; and (iii) We have used URL Entity Matching to analyze the content of the URLs provided in the tweet and to identify the health related entities and keywords. Then, we created a set of binary classifiers to classify the extracted Tweets. A binary classifier receives a set of input data as the training set, and creates a model to identify an item is relevant or not. For example, in our scenario a classifier predicts a Tweet is relevant to



**Fig. 9** The comparison between the DataSynapse and the classic classification approach for the health related tweets (a), the scalability experiment (b) and similarity comparison among data objects in the semantic-item (curated tweet) and the domain knowledge (budget-KB)

‘health’ or not. Next section explains how we created the training set and classifiers in detail.

To train binary classifiers we created two different training sets. The first training set was created through KEYM approach and the second training set was created through the proposed featurization technique. Considering that we have around 15 million tweets, using the KEYM approach we identified 50 thousand tweets as relevant to health. Next, we applied some preprocessing on tweets: for example, we have eliminated tweets containing less than four keywords and also tweets that contain non English words, and eliminated the URLs and twitter domains. Also it is possible to extract keywords from hashtags and replace them with the hashtag. We have also removed the duplicate tweets (e.g., retweeted tweets). Finally, we have generated around 20 thousand preprocessed tweets. We labelled the extracted tweets as relevant and feed them as an input to the machine learning algorithm (e.g., Naive Bayes, KNN and SVM classifiers). In addition, we feed the classifier with a dataset of irrelevant tweets from our previous work [13,36] which manually labelled through crowds. For the test set, we have manually labeled 600 tweets which contains 322 health related and 278 unrelated tweets. We consider each tweet as a document, and process it by stemming, removing stop words, punctuations and numbers, and lower casing the entire tweet. We followed the same procedure to create the second dataset for evaluating the performance of featurization technique.

As illustrated in Figure 9a, DataSynapse significantly improves the quality of extracted knowledge compared to the classical curation pipeline (in the absence of feature extraction and domain-linking contextualization). The proposed technique will identify many related tweets (that should be contained in the returned results) and accordingly the accuracy of the result can be improved. Notice that, *accuracy* is the proximity of measurement results to the true value, and calculated as  $Accuracy = (TP + TN) / (TP + TN + FP + FN)$ , where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. Accordingly, the *Error Rate* is Equal to  $(1 - accuracy)$ . As an ongoing work, to improve the precision and recall: (i) we are going to use rules in combination with the machine learning approach for further filtering results; (ii) we will use some refinement techniques, e.g., to merge the results obtained from KEYM approach with the Budget-KB; (iii) we will add more feature to our model; and (iv) we are enhancing the model, currently supporting unigram, to support n-grams and leverage multiple machine learning techniques for further filtering the results.

### 5.2.4 Scalability study

For evaluating the efficiency, we have scaled the experiment over three different configurations on Amazon EC2. The specification of these VMs have been specified earlier in the paper: one, four and eight machines. As discussed in Sect. 3.5, we have leveraged UIMA framework to wrap curation APIs as network services, and to scale to very large volumes by replicating processing pipelines over a cluster of networked nodes. Figure 9b illustrates the execution times and the scalability evaluation.

### 5.2.5 Social issues

Identifying social issues is challenging as it requires the budget analyst to properly understand the candidate tweets. To provide the candidate tweets, we identified the tweets having negative sentiments. To achieve this goal we have used the classified tweets. For example, the DataSynapse approach classified 5823 tweets related to anxiety, 2934 tweets related to diabetes, 22,430 tweets related to cancer, and 16,931 tweets related to mental health. We have reused the sentiment classifier implemented in the Apache PredictionIO to identify the tweets with negative sentiment. For example, out of 2934 diabetes related tweets the algorithm identified 615 tweets with negative sentiment. As another example, we have identified 1549 tweets with negative sentiment in the Mental Health category. Later on, the analyst is able to use the feature-based summaries to analyze the candidate tweets based on a specific goal, e.g., identify tweets discussing about a social issue related to health and specifically about the Medicare (federal health insurance program) or an issue related to public Hospital services.

## 6 Conclusion and future work

Understanding and analyzing open data now is recognized as a strategic priority for governments. In this context, the *data curation* process becomes a vital analytics asset for understanding the open data. To address this need, we have introduced DataSynapse, a general purpose curation pipeline. The goal here has been to facilitate analytical tasks through transforming raw data into featurized (through the proposed feature engineering approach) and thereafter contextualized (requires contracting the domain knowledge and linking extracted data to that) data. DataSynapse is offered as an extensible and scalable microservice-based architecture implemented as a set of APIs, that are publicly available as an open source project on GitHub.

As for future work, we are designing a declarative rule-based language to assist analysts query and analyze the curated data in an easy way. We are also extending the Budget-KB by identifying further relevant concepts and their instances in other budget categories program. We are working on novel techniques for intelligent narrative discovery and to summarize the large amount of curated data in the Knowledge Lake.

**Acknowledgements** We Acknowledge the data to decisions CRC (D2D CRC) and the cooperative research centres program for funding this research.

## References

1. Aggarwal, C.C.: An Introduction to Social Network Data Analytics, pp. 1–15. Springer, Berlin (2011)
2. Anderson, M.R., Antenucci, D., Bittorf, V., Burgess, M., Cafarella, M.J., Kumar, A., Niu, F. et al.: Brainwash: a data system for feature engineering. In: CIDR (2013)
3. Beheshti, S.-M.-R., Nezhad, H.R.M., Benatallah, B.: Temporal provenance model (TPM): model and query language. CoRR, abs/1211.5009 (2012)
4. Beheshti, S.-M.-R. et al.: Galaxy: a platform for explorative analysis of open data sources. In: Proceedings of the 19th International Conference on Extending Database Technology, (EDBT), pp. 640–643 (2016). <https://dblp.org/rec/bibtex/conf/edbt/BeheshtiBM16>

5. Beheshti, S.-M.-R., Benatallah, B., Motahari-Nezhad, H.R.: Scalable graph-based OLAP analytics over process execution data. *Distrib. Parallel Databases* **34**(3), 379–423 (2016)
6. Beheshti, S.-M.-R., Benatallah, B., Sakr, S., Grigori, D., Motahari-Nezhad, H.R., Barukh, M.C., Gater, A., Ryu, S.H.: *Process Analytics—Concepts and Techniques for Querying and Analyzing Process Data*. Springer, Berlin (2016)
7. Arocena, P.C., Glavic, B., Mecca, G., Miller, R.J., Papotti, P., Santoro, D.: Benchmarking data curation systems. *IEEE Data Eng. Bull.* **39**(2), 47–62 (2016)
8. Beheshti, S.-M.-R., Tabebordbar, A., Benatallah, B., Nouri, R.: On automating basic data curation tasks. In: *Proceedings of the 26th International Conference on World Wide Web Companion*, Perth, Australia, April 3–7, 2017, pp. 165–169 (2017)
9. Beheshti, S.-M.-R., Benatallah, B., Venugopal, S., Ryu, S.H., Motahari-Nezhad, H.R., Wang, Wei: A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Computing* **99**(4), 313–349 (2017)
10. Beheshti, A., Benatallah, B., Nouri, R., Chhieng, Van M., Xiong, H., Zhao, X.: Coredb: a data lake service. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM 2017, Singapore, November 06–10, 2017, pp. 2451–2454 (2017)
11. Beheshti, A., Benatallah, B., Nouri, R., Tabebordbar, A.: Corekg: a knowledge lake service. *PVLDB* **11**(12), 1942–1945 (2018). <https://dblp.org/rec/bibtex/journals/pvldb/BeheshtiBNT18>
12. Beheshti, A., Schiliro, F., Ghodratnama, S., Amouzgar, F., Benatallah, B., Yang, J., Sheng, Q.Z., Casati, F., Motahari-Nezhad, H.R.: iprocess: Enabling iot platforms in data-driven knowledge-intensive processes. In: *Business Process Management Forum - BPM Forum 2018* (2018)
13. Beheshti, A., Vaghani, K., Benatallah, B., Tabebordbar, A.: Crowdcorrect: A curation pipeline for social data cleansing and curation. In: *Information Systems in the Big Data Era—CAiSE Forum 2018*, Tallinn, Estonia, June 11–15, 2018, Proceedings, pp. 24–38 (2018)
14. Chai, X., Deshpande, O., Garera, N., Gattani, A., Lam, W., Lamba, D.S., Liu, L., Tiwari, M., Tourn, M., Vacheri, Z., Prasad, S.T.S., Subramaniam, S., Harinarayan, V., Rajaraman, A., Ardalani, A., Das, S., Suganthan, G.C.P., Doan, A.: Social media analytics: the kosmix story. *IEEE Data Eng. Bull.* **36**(3), 4–12 (2013)
15. Chen, H., Chiang, R.H.L., Storey, V.C.: Business intelligence and analytics: from big data to big impact. *MIS Q.* **36**(4), 1165–1188 (2012)
16. Chiticariu, L., Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F., Vaithyanathan, S.: Systemt: an algebraic approach to declarative information extraction. In: *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, July 11–16, 2010, Uppsala, pp. 128–137 (2010)
17. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM.* **51**(1), 107 (2008)
18. Deshpande, M., Ray, D., Dixit, S., Agasti, A.: Shareinsights: an unified approach to full-stack data processing. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, Melbourne, Victoria, Australia, May 31–June 4, 2015, pp. 1925–1940 (2015)
19. Doan, A., Domingos, P.M., Halevy, A.Y.: Reconciling schemas of disparate data sources: a machine-learning approach. In: *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, Santa Barbara, CA, USA, May 21–24, 2001, pp. 509–520 (2001)
20. Ferrucci, D.A.: Introduction to ‘this is watson’. *IBM J. Res. Dev.* **56**(3,4), 4:1–4:11 (2012)
21. Freitas, A., Curry, E.: Big data curation. In: Cavanillas, J.M., (ed.), *New Horizons for a Data-Driven Economy*, pp. 87–118. Springer, Berlin (2016)
22. Terrizzano, I. et al.: Data wrangling: the challenging journey from the wild to the lake. In: *CIDR* (2015)
23. Kim, N.W., Jung, J., Ko, E.-Y., Han, S., Lee, C.W., Kim, J., Kim, J.: Budgetmap: engaging taxpayers in the issue-driven classification of a government budget. In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, CSCW 2016, San Francisco, CA, USA, February 27–March 2, 2016, pp. 1026–1037 (2016)
24. Lee, K., Agrawal, A., Choudhary, A.: Real-time disease surveillance using twitter data: demonstration on flu and cancer. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’13, pages 1474–1477, New York, NY, USA (2013). ACM
25. Lohr, S.: The age of big data. *New York Times*, 11 (2012)
26. Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., Stoyanov, V.: SemEval-2016 task 4: sentiment analysis in twitter. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16–17, 2016, pp. 1–18 (2016)

27. Pandey, N., Natarajan, S.: How social media can contribute during disaster events? case study of chennai floods 2015. In: 2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016, Jaipur, India, September 21–24, 2016, pp. 1352–1356 (2016)
28. Paul Suganthan, G.C., Sun, C., Krishna Gayatri, K., Zhang, H., Yang, F., Rampalli, N., Prasad, S., Arcaute, E., Krishnan, G., Deep, R., Raghavendra, V., Doan, A.: Why big data industrial systems need rules and what we can do about it. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31–June 4, 2015, pp. 265–276 (2015)
29. Pu, X., Jin, R., Wu, G., Han, D., Xue, G.-R.: Topic modeling in semantic space with keywords. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19–23, 2015, pp. 1141–1150 (2015)
30. Ritter, A., Clark, S., Mausam, E., Oren: named entity recognition in tweets: an experimental study. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1524–1534 (2011)
31. Ruder, T.D., Hatch, G.M., Ampanozi, G., Thali, M.J., Fischer, N.: Suicide announcement on facebook. Crisis (2011)
32. Russom, P., et al.: Big data analytics. TDWI best practices report, fourth quarter **19**, 40 (2011)
33. Sellam, T., Müller, E., Kersten, M.L.: Semi-automated exploration of data warehouses. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19–23, 2015, pp. 1321–1330 (2015)
34. Stonebraker, M., et al.: Data curation at scale: the data tamer system. In: CIDR (2013)
35. Fabian, M.: Suchanek and Gerhard Weikum. Knowledge bases in the age of big data analytics. Proc. VLDB Endow. **7**(13), 1713–1714 (2014)
36. Tabebordbar, A., Beheshti, A.: Adaptive rule monitoring system. In: 40th International Conference on Software Engineering (ICSE), International Workshop on Software Engineering for Cognitive Services (SE4COG) (2018)
37. Tene, O., Polonetsky, J.: Big data for all: Privacy and user control in the age of analytics. N. J. Tech. Intell. Prop. **11**, xxvii (2012)
38. Troncy, R.: Linking entities for enriching and structuring social media content. In: WWW, pp. 597–597 (2016)
39. Karlgren, J., Bohman, M., Ekgren, A., Isheden, G., Kullmann, E., Nilsson, D.: Semantic topology. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3–7, 2014, pp. 1939–1942 (2014)
40. Wang, S., Tang, J., Aggarwal, C.C., Liu, H.: Linked document embedding for classification. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24–28, 2016, pp. 115–124 (2016)
41. Zarras, A.V., Vassiliadis, P., Dinos, I.: Keep calm and wait for the spike! insights on the evolution of amazon services. In: Advanced Information Systems Engineering - 28th International Conference, CAiSE 2016, Ljubljana, Slovenia, June 13–17, 2016. Proceedings, pp. 444–458 (2016)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Amin Beheshti<sup>1</sup>  · Boualem Benatallah<sup>2</sup> · Alireza Tabebordbar<sup>2</sup> · Hamid Reza Motahari-Nezhad<sup>2,3</sup> · Moshe Chai Barukh<sup>2</sup> · Reza Nouri<sup>2</sup>**

Boualem Benatallah  
boualem@cse.unsw.edu.au

Alireza Tabebordbar  
alirezat@cse.unsw.edu.au

Hamid Reza Motahari-Nezhad  
hamidm@cse.unsw.edu.au; hamid.motahari@ey.com

Moshe Chai Barukh  
mbarukh@cse.unsw.edu.au

Reza Nouri  
snouri@cse.unsw.edu.au

<sup>1</sup> Macquarie University, Sydney, Australia

<sup>2</sup> University of New South Wales, Sydney, Australia

<sup>3</sup> EY AI Lab, Palo Alto, USA