Semantic Web 1 (2016) 1–5
IOS Press

A Linked Data Wrapper for CrunchBase

Michael Färber [*,**], Carsten Menne, and Andreas Harth
Karlsruhe Institute of Technology (KIT), Institute AIFB, 76131 Karlsruhe, Germany

Abstract. CrunchBase is a database about startups and technology companies. The data can be searched, browsed, and edited via a website, but is also accessible via an entity-centric HTTP API in JSON format. We present a wrapper around the API that provides the data as Linked Data. The wrapper provides schema-level links to schema.org, Friend-of-a-Friend and Vocabulary-of-a-Friend, and entity-level links to DBpedia for organization entities. Further, we describe how to harvest the RDF data to obtain a local copy of the data for further processing and querying that goes beyond the query facilities of the CrunchBase API. Our Linked Data API for CrunchBase and a previous version of it have already been used in two cases, whereas our crawled CrunchBase RDF data set has been used once for data integration and once for information extraction on text. CrunchBase has also been used twice for exploratory data analysis.

Keywords: Linked Data, CrunchBase, RDF, API, Startups

1. Introduction

CrunchBase[1] in an online platform providing information about startups and technology companies, including related entities such as their sold products, their key people, and their made and received investments. CrunchBase is mainly used by entrepreneurs, investors, and business analysts to look up information to gain market insights. A typical CrunchBase user may be interested in answering questions such as: "Should we invest in startup X?".

CrunchBase was founded in 2007 by Mike Arrington, the founder of the TechCrunch weblog, to track data about startups covered in posts. Nowadays, CrunchBase is used by millions of users to track the fast-changing world of startups. The CrunchBase data is edited by a community: Users with an account can add and modify facts via forms in a browser. Facts are thereby attributes of entities, such as the birth date of a person, or relations between entities, such as the acquisition of a company by another company. The CrunchBase schema predefines entity types, attributes and relations. Since the CrunchBase data is internally stored as a graph, the database is also called *Business Graph*.[2] Given the graph-based data model, the CrunchBase data is in principle amenable to be modelled in RDF.

Although CrunchBase users contribute worldwide, CrunchBase covers mainly the US market: About every second company office listed on CrunchBase is located in the US.[3]

The data stored in the CrunchBase database is usually accessed via a Web browser. However, CrunchBase also provides its data in other ways. The following options for data access are provided:

1. *Open Data Map (ODM)* is a package of JSON or CSV files which provides daily updated information about people and organizations. The ODM only provides a restricted set of entity attributes.
2. The *Excel Data Export* provides a monthly updated spreadsheet, containing a partial view

[*]Corresponding author. E-mail: michael.faerber@kit.edu.
[**]This work was carried out with the support of the German Federal Ministry of Education and Research (BMBF) within the Software Campus project *SUITE* (Grant 01IS12051).

[1]See `http://crunchbase.com/`, requested on Feb 4, 2016.

[2]See `https://info.crunchbase.com/the-business-graph/`, requested on Feb 4, 2016.

[3]About 55% (135,316 of 245,454) of the offices listed on CrunchBase exhibit U.S.A. as location country. For this information, we queried our CrunchBase RDF data which we retrieved via our Linked Data API. See Section 3 for more information.

(companies, rounds, investments, and acquisitions) on the overall data.
3. The *REST API* allows for accessing the entire contents of the CrunchBase database (which is available under the CreativeCommons license). To use the REST API, the user has to obtain an API key. There are different kinds of licenses available; we operate with the free academic research license.

Despite the REST API, CrunchBase data is neither available in RDF nor linked to other data sets on the Web. While Nowack already provided an RDF wrapper for the CrunchBase API called Semantic CrunchBase in 2008[4], the service is no longer available.

Using Semantic Web technologies such as RDF on the CrunchBase API leads to the following benefits:

1. *More complex queries*: Although the CrunchBase data is internally stored as a graph, CrunchBase does not provide an interface for querying with a graph query language such as SPARQL. Instead, the CrunchBase API only allows entity-centric requests, revealing information in JSON format about specific entities with their attributes and relations. A typical API request can be formulated in natural language as: "Show me all stored acquisitions of Facebook Inc."[5]
   In contrast, many professional CrunchBase users may want to formulate more elaborate queries. Such a query, formulated in natural language, might be: "Which companies existing at most 5 years have been acquired for more than 1 bn USD?" Having up-to-date answers to such questions can result in better market insights, and, hence, in increased investment performance and in improved business planning for entrepreneurs.
2. *Using CrunchBase data with existing Semantic Web data*: Semantic Web technologies are often used to integrate data from separate data sources. The integration becomes possible once data has been transformed into RDF. With the CrunchBase data available in RDF, the user/developer can combine the data with other RDF data. For instance, the information about the location and the technology sector of companies in CrunchBase can be combined with information about job offers from an online job seeker platform. By integrating data from both platforms, one can pose queries such as: "Find all companies within the area of city X which offer jobs in the field of Y." Mochol et al. [6] give an example of how to use Semantic Web data to achieve the answering of such questions.
3. *Using existing analytics methods in conjunction with CrunchBase data*: For market insight purposes (e.g., detecting acquisitions in news texts), already some well-performing Semantic Web methods such as text annotation – i.e., linking mentions in a text to their corresponding Knowledge Base entries – and relation extraction – extracting triples from text – are available. However, these methods often only work well for specific underlying data sets such as Wikipedia or DBpedia. The data which is useful for market monitoring tasks (e.g., acquisitions of companies) such as CrunchBase data, in contrast, is often not supported by these tools. However, if entities in CrunchBase are linked via `owl:sameAs` to other Knowledge Bases such as DBpedia (as it is provided by our proposed CrunchBase Linked Data API), these links can be exploited in order to use both CrunchBase data and the well-performing Semantic Web analytics tools.

In this paper, we make the following contributions:

- We provide a process-oriented description of creating a Linked Data wrapper, which transforms JSON provided by an API into RDF (in both JSON-LD and N-Triples serializations). We implement our workflow on the CrunchBase REST API, but the method can serve as template for wrapping any access-restricted REST API with JSON output. Both an implementation of the Linked Data wrapper and a deployed version of it are available online (see Table 1). The CrunchBase Linked Data API has been applied in two use cases so far (see Section 4).
- We show how an up-to-date RDF data set of CrunchBase can be obtained at any time with the help of the Linked Data wrapper. The data set can subsequently be used for a variety of use cases such as market monitoring and is freely available for further usage. So far, besides internal usage

[4] See `http://bnode.org/blog/2008/07/29/semantic-web-by-example-semantic-crunchbase` and `http://cb.semsol.org/` (requested on Feb 5, 2016), which is no longer available.

[5] The corresponding HTTP GET request looks like: `https://api.crunchbase.com/v/3/organizations/facebook/acquisitions?user_key={api-key}`

CrunchBase Linked Data API entry page:	http://km.aifb.kit.edu/services/crunchbase/
Source code of the Linked Data wrapper for CrunchBase:	https://github.com/aifb/linked-crunchbase/
CrunchBase RDF data set:	http://km.aifb.kit.edu/sites/crunchbase/crunchbase-dump-201510.nt.gz
Visualizations based on SPARQL queries against CrunchBase RDF:	http://km.aifb.kit.edu/sites/crunchbase/

Table 1
Links to ressources.

for information extraction on text, the crawled CrunchBase RDF data set has been used by others for data integration. Similar CrunchBase data sets have been used for exploratory data analysis.

Regarding the linked data set description papers published by the Semantic Web Journal so far [3], five out of all 38 papers mention JSON as input or output data format, but only the description of the Facebook RDF Wrapper [8] describes a conversion of JSON to RDF. As pointed out in the paper, JSON-LD was considered, but disregarded, "since its conventions varied too widely from the existing JSON format."

Since the publication of that article, things have changed: JSON-LD became a W3C recommendation[6] in 2014. More and more developers use JSON-LD[7] as it is easy to transform existing JSON to JSON-LD. If JSON-LD is used, the many existing Web applications and Web services which are so far based on JSON can then also be used in the Semantic Web. Moreover, JSON-LD can be easily converted into other RDF serialisations; thus, JSON-LD applications and services are compatible with RDF-based applications and services.

Other approaches often convert entire data sets to RDF. In contrast, we first provide a Linked Data interface to the API, and then create a data set via crawling the API. Such an approach allows for the collection of parts or all of the data, and provides up-to-date access to data about individual entities.

In the following, we give a short overview of the Linked Data API and of the RDF data set, before describing them in more detail in the following sections.

Our workflow to create the Linked Data API is shown in Figure 1. To enrich our Linked Data API with `owl:sameAs` links to DBpedia we first set up the API without any links. With this simple API we obtained the data set and linked entity URIs to DBpedia. The obtained links were then integrated into the API, so that the links are available when a URI of the wrapper is dereferenced.

We have implemented the Linked Data API for CrunchBase presented in this paper. The code of the wrapper is available on GitHub[8] under the MIT license, and we maintain an instance of the wrapper[9]. Additional information about the wrapper is made available at the entry page. The wrapper provides data in different formats via content negotiation, and enriches CrunchBase entities retrieved from the CrunchBase REST API with `owl:sameAs` links to DBpedia, which is a hub in the Linking Open Data cloud. Besides the CrunchBase Linked Data API implementation, we provide a description of the service and of the used schema (predefined by CrunchBase) as OWL file and as a VoID file.

For setting up a local CrunchBase RDF Knowledge Base for research on news monitoring [1], we built a CrunchBase RDF data set with the help of the implemented CrunchBase Linked Data API. We thereby restricted ourselves to facts of organizations, people, products, and acquisitions, since entities of those types contain the facts which are – in our minds – the most important for our news monitoring task. We crawled in October 2015 and retrieved 7,373,480 unique entities. Our crawled RDF data set can be reused by all researchers who want to extend existing Knowledge Bases by CrunchBase data or who want to analyze the RDF data set for their own purposes. Note, however, that the CrunchBase data is released under the Creative Commons Attribution-NonCommercial 4.0 International Public License.[10] Any crawled CrunchBase RDF data set is therefore only to be used for non-commercial purposes.

[6] See https://www.w3.org/TR/json-ld/, requested on Feb 5, 2016.

[7] See https://trends.builtwith.com/docinfo/JSON-LD, requested on Feb 5, 2016.

[8] See https://github.com/aifb/linked-crunchbase, requested on Feb 5, 2016.

[9] See http://km.aifb.kit.edu/services/crunchbase/, requested on June 28, 2016.

[10] See https://creativecommons.org/licenses/by-nc/4.0/, requested on Feb 4, 2016.

Fig. 1. Schematic view of the steps taken to create a Linked Data version of the CrunchBase API.

The rest of the paper is organized as follows: In Section 2, we present our Linked Data API for CrunchBase, which is designed as a wrapper around the official CrunchBase REST API. In Section 3, we give insights into our CrunchBase RDF Knowledge Graph whose data was crawled with the help of the CrunchBase Linked Data API. After describing the usage of the Linked Data API and the crawled RDF data in Section 4, we conclude in Section 5.

2. The CrunchBase Linked Data API

In this section we give an overview of our implemented CrunchBase Linked Data API.

Figure 2 shows the basic workflow when accessing data via our Linked Data API. We can distinguish between the following steps:

1. A user application such as the data integration framework Linked Data-Fu [7] calls the CrunchBase Linked Data API via a HTTP GET request. Thereby, the requested API URI, the CrunchBase API user key, and the requested content type (`JSON`, `JSON-LD`, or `N-Triples`) is specified.[11]
2. The Linked Data API servlet takes the HTTP request and calls the official CrunchBase REST API using the specified information.
3. The Linked Data API servlet receives the data from the CrunchBase REST API and transforms it into one of the provided content types. As far as mappings to DBpedia are available, links to DBpedia entities are included.
4. The user application receives the data from the Linked Data API and further processes the data.

Our CrunchBase Linked Data API provides its data in three different formats: JSON, JSON-LD, and RDF/N-Triples.

1. `JSON (application/json)`: The official CrunchBase REST API provides data as JSON. For JSON responses, we forward the data retrieved from the CrunchBase REST API without any modifications.
2. `JSON-LD (application/ld+json)`: We provide data in our CrunchBase Linked Data API as JSON-LD by restructuring the data in JSON retrieved from the official CrunchBase API. The main restructuring steps are removing meta-data and adding namespaces.
3. `RDF/N-Triples (text/turtle)`: We provide also N-Triples as format for RDF, as it is one of the widely used formats in current Semantic Web systems.

As we provide the CrunchBase Linked Data API as a third-party tool on top of the CrunchBase REST API (currently in version 3), the RDF wrapper needs to be modified as soon as the CrunchBase API changes. This is ensured by a process of monitoring the CrunchBase mailing list.

2.1. API Authorization

Since the official CrunchBase API is only accessible with an API key, users of the CrunchBase Linked Data API also need to provide a valid API key for requesting data; otherwise HTTP status code 401 is returned. When using the CrunchBase JSON API, the key is passed via a parameter in the URI. However, applying this method to the CrunchBase Linked Data API, the API key would be part of the identifier and public for everyone. To resolve this issue, user agents can pass the API key through the `Authorization` header field.[12] Our approach allows a neat integration of the CrunchBase Linked Data API in other services and frameworks, since the URIs do not need to be modified due to authorization and since standard web technologies are used.

The CrunchBase API key needs to be requested at the CrunchBase developer team. As the data is licensed

[11] An example API call with cURI is `cURI -v -H "Accept:text/turtle" -header "Authorization: Basic {Base64-encoded key}" http://km.aifb.kit.edu/services/crunchbase/api/organizations/facebook`.

[12] We use the `Basic Authentication` method. The key is stored in the "user" field; the "password" field remains empty.

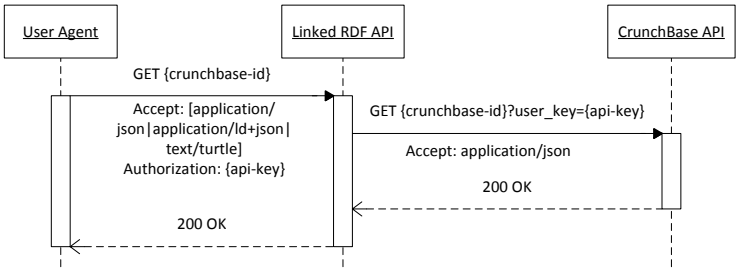

Fig. 2. UML sequence diagram illustrating the use of the wrapper. The wrapper supports different representations via content negotiation. The API key is passed to the wrapper via an `Authorization` header, and passed from the wrapper to the CrunchBase API via URI parameter.

Table 2
URI Design for the CrunchBase Linked Data API using organizations as example entity type.

URI Template	Description
`/`	Index page
`/api/`	Base for every request
`/api/organizations`	Returns all organizations in CrunchBase
`/api/organizations/:permalink`	Returns information about a given entity encoded as `permalink`, e.g. `facebook`
`/api/organizations/:permalink/:relationship`	Returns information about a given `relation`, e.g. `acquisitions`

under a CreativeCommons license,[13] we decided to provide some RDF data from a static copy of CrunchBase if no API key is given. To do so, the Linked Data API checks if the `Authorization` header is set in the HTTP request. If it is set, the procedure as outlined above is invoked. If it is not set, a SPARQL query against a triple store with CrunchBase data is executed and the results are served to the user. This approach enables that all URIs provided by the CrunchBase Linked Data API are dereferencable and can be requested by anyone on the Web. Our Linked Data API is therefore also visible and partly usable for users who follow a link to our API, but who do not possess an API key.

2.2. URI Schema Used by the Linked Data API

Table 2 shows the URI design for accessing the Linked Data API. As we built a wrapper around the original CrunchBase API and since the URIs for the official CrunchBase API and the Linked Data API are designed in the same way, every request sent to the official CrunchBase API can be sent to our wrapper. For more information regarding the API URI structure, we can refer to the official CrunchBase documentation.[14]

2.3. Schema Used by the Linked Data API

For the CrunchBase Linked Data API, the data model of the official CrunchBase REST API is reused and only slightly modified. All entity types and the set of possible attributes and relations between entities are not changed. Also the domains and ranges of the relations were retained. Figure 3 illustrates the classes and relations used in the data returned from the wrapper. The schema of the Linked Data API is dereferencable and described in an OWL file, which is provided on our Linked Data API entry page. Furthermore, we enriched our ontology with VOAF (Vocabulary-of-a-Friend)[15] descriptors. VOAF is an extension of VoID, in order to link our ontology to other vocabularies and to introduce the vocabulary to the Linking Open Data community.[16]

We can outline further characteristics of the data modeling used by the Linked Data API:

1. Not all relations between entities are modeled as single triples in the CrunchBase database. For instance, acquisitions do not only have an ac-

[13] This is indicated in each returned RDF document by additional triples dedicated to the license.

[14] See `https://data.crunchbase.com/docs/using-the-api/`, requested on Aug 2, 2016.

[15] See `http://lov.okfn.org/vocommons/voaf`, requested on Feb 5, 2016.

[16] See `https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`, requested on Aug 1, 2016.

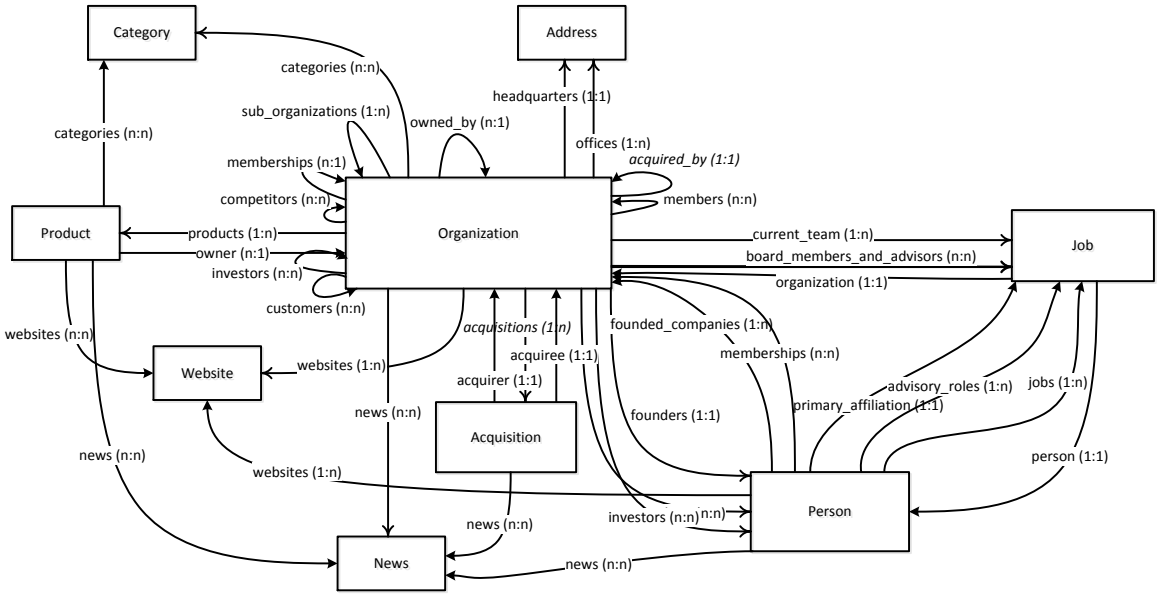

Fig. 3. Diagram showing the classes and relations supported by the CrunchBase Linked Data API. We use a (source, target) notation to indicate the cardinality of relations.

quiree and an acquirer, but for instance also a date and a type of the acquisition. Events such as acquisitions and investments are modeled as separate entities. In RDF, the concept of n-ary relations[17] is used to represent these entities. In our data model, 21 entity types are modeled as blank nodes due to that representation form.[18]

2. Noteworthy is also the modeling of the uncertainty/trustworthiness of data values such as dates. CrunchBase allows the editor to specify to which degree he/she knows the value. This confidence value is encoded in binary format: If the first bit is set, the year is valid; if the second bit is set, the month is valid, etc. The final bit vector is stored as decimal representation ranging from 0 (complete unknown/unsure) to 7 (very confident/knowing the exact date). Based on this encoding, property values stored as strings can be easily converted to the XML schema definition (XSD) format[19] such as `xsd:date` if they are valid.

Linked Data is based on the best practice to use existing vocabularies and to link entities, classes, and properties between data sources in the Linking Open Data (LOD) cloud. As the topic of CrunchBase is a bit special, we did not find a proper vocabulary for CrunchBase. Therefore we decided to use our own vocabulary and link it to suitable entity types and properties from schema.org (with the prefix `schema` in the following). Table 3 shows some examples of entity types which are linked to schema.org. The list of all mappings is provided in the form of an OWL file on our API entry page.[20]

2.4. Linking CrunchBase Entities to DBpedia

Furthermore we created a list of `owl:sameAs` links between CrunchBase entities and the corresponding DBpedia entities. If one of the CrunchBase entities in this list is requested via the CrunchBase Linked Data API, the corresponding `owl:sameAs` link is added to the returned data. We implemented mappings between

[17] See `https://www.w3.org/TR/swbp-n-aryRelations/`, requested Aug 1, 2016.

[18] These include the following entity types: Organizations, News, Images, Videos, Acquisitions, Categories, Addresses, Websites, Jobs, Investments, FundRaises, Products, Locations, Degrees, Markets, and WebPresence. The last two are not listed in official CrunchBase API documentation.

[19] See `https://www.w3.org/2001/XMLSchema`, requested on Feb 5, 2016.

[20] See `http://km.aifb.kit.edu/services/crunchbase/`, requested Feb 5, 2016.

Table 3
Mappings (sample) between CrunchBase and schema.org entity types.

CrunchBase entity type	schema.org entity type[21]
cbw:Address	schema:Place
cbw:Image	schema:ImageObject
cbw:News	schema:NewsArticle
cbw:Organization	schema:Organization
cbw:Person	schema:Person
cbw:Product	schema:Product
cbw:Video	schema:VideoObject
cbw:Website	schema:WebSite

CrunchBase entities and DBpedia entities for different entity types such as organizations, people, and products. However, our evaluation revealed that only mappings of CrunchBase organizations provide an acceptable precision rate. We therefore included only those mappings into our Linked Data API. In the following, we describe the details of the different mapping methods.

2.4.1. Organization Mappings

The mapping between CrunchBase entities of type organization and DBpedia entities led to acceptable precision rates. For each CrunchBase organization, the mapping method checks whether it can find a DBpedia entity which possesses the same homepage domain as the CrunchBase entity; i.e., it compares the attribute value of homepage in CrunchBase with the property value of foaf:homepage in DBpedia. For a better string comparison, we only considered the Fully Qualified Domain Name (FQDN). If there is a match, the entity pair is added to the mapping list. In total, we obtained 16,702 mappings for all 567,937 CrunchBase organization entities by means of the described mapping method. The recall value of the mappings seems to be low. Keep in mind, however, that, firstly, the overlap between CrunchBase entities and DBpedia entities is generally quite low (see below for an analysis on this). One of the main reasons for that is that CrunchBase has no restrictions regarding the insertion of new entities or facts: Any contributor can add new entities. This also holds for Wikipedia. However, bots and Wikipedia contributors delete new entities if these entities do not seem to be of general public interest.[22] Secondly, another reason for the low recall might be that the entity types and the property values of foaf:homepage are not always very clean in DBpedia, since DBpedia is extracted automatically from Wikipedia.

To evaluate the precision of the gained owl:sameAs links, we manually evaluated 100 randomly chosen owl:sameAs triples of CrunchBase organization entities. 92 of them were completely correct. Most of the incorrect mappings (six out of eight) went wrong since organization A was mapped to a sub-organization B of A or the company A had been acquired.

2.4.2. People Mappings

People constitute the second-largest group of entities after organizations. Therefore, it is worthwhile to map people represented in CrunchBase to DBpedia. People entities, however, require higher effort for mapping. Using just the given name and surname leads to a very high rate of false positives, since a lot of people have the same names (e.g., Brian Ray, who is the CEO of Link Labs on CrunchBase, but a musician on DBpedia).[23]

Also for people we can find a low intersection between CrunchBase and DBpedia. In addition, it is striking that many people in CrunchBase are represented, but do not exhibit for a disambiguation enough attributes or relations. Often, people do not have any attributes or relations at all.

Our evaluation regarding the accuracy of the mappings between CrunchBase people and the corresponding DBpedia entities was as follows: We randomly picked 300 CrunchBase person entities and for each entity verified via manual investigation on Wikipedia whether there is a corresponding entity in DBpedia, whether there is no corresponding entity in DBpedia for sure, or whether no statement can be made, since, for instance, not enough information is available for disambiguation. We therefore only considered people in Wikipedia with the same given name and family name as given in CrunchBase. We came to the following conclusions:

- 263 out of the 300 (87.7%) CrunchBase people entities do not exist in DBpedia.
- 5 out of the 300 have a counterpart in DBpedia.
- For 32 people, not enough information was available to determine with confidence whether the entity exists in DBpedia.

[22] See Wikipedia's notability guidelines at https://en.wikipedia.org/wiki/Wikipedia:Notability (requested Aug 2, 2016).

[23] See https://www.crunchbase.com/person/brian-ray#/entity and http://dbpedia.org/page/Brian_Ray, requested on Feb 2, 2016.

Hence, if the mapping method is just based on the name (given and family name), about 90% of the `owl:sameAs` relations would be incorrect.

2.4.3. Product Mappings

Products exhibit similar difficulties w.r.t. mappings to DBpedia as people. There are almost no modeled relations or attributes for products which we could use, only the manufacturer/owner, the name, and the description. We leave product mappings therefore for future work.

3. The CrunchBase RDF Data Set

Besides creating a Linked Data API for CrunchBase, we also obtained an RDF dataset containing information about CrunchBase organizations, people, acquisition, and products (including their attributes and relations).[24] Our goal was to build a local CrunchBase RDF Knowledge Base which can be used in the context of news monitoring in the technology business domain. Thereby, facts extracted from the news texts such as acquisitions or products of companies are compared with the facts stored in the CrunchBase KB. Due to that scenario, we restricted the crawling of CrunchBase data to the mentioned entity types as those are highly relevant for that scenario.

Technically, we used the Linked Data API of CrunchBase in conjunction with Linked Data-Fu [7], a framework for information integration.

We built the CrunchBase RDF dump along the following steps:

1. We crawled all so-called summary data via the CrunchBase Linked Data API by using the API endpoints for organizations, people, and products.[25] This summary data contained the most important facts about the entities of the mentioned entity types. To retrieve all the data we were following the `next_page_URI` since CrunchBase spreads its information over multiple pages.
2. We crawled all the information about people, products, and organizations by requesting the `api_path` URIs which are contained in the summary data. In this way, we obtained all missing data (attributes and relations) regarding people, products, and organizations.
3. As CrunchBase lists only eight entities per relation via those API requests, we have to crawl separately every relation of any entity in case this relation has more than eight different objects. Note that the API calls are very expensive in time. The Linked Data API is restricted by a 1 second limit of the official CrunchBase REST API. Thus, we crawled first the acquisitions and the product-news, and then the remaining relations. We thereby followed the `next_page_URI` links if necessary.

Table 4
Distribution of entities among the different entity types in our crawled CrunchBase data set (as of October 2015).

Entity type	Number of instances
Job	1,946,435
Website	1,348,449
Organization	567,937
News	519,763
Person	430,093
Product	60,076
Acquisition	33,127

The original CrunchBase API uses some meta-data attributes such as `uuid` (as id for an entity), the `web-path`, the `api-path`, etc. As this meta-data is not relevant for building a Knowledge Base with CrunchBase data, it is excluded from the CrunchBase RDF file.

Since the crawled CrunchBase entities are highly linked via relations to entities of further entity types, our final CrunchBase RDF data set comprised entities of 25 different entity types and 210 different relations (KB properties). We retrieved 83,737,509 unique triples in total. Table 4 shows the distribution of the entities among the different entity types. Not surprisingly, CrunchBase' main focus is on organizations (including companies) and related entities such as people, products, acquisitions, and jobs. News and websites are also well covered due to the affiliation of CrunchBase to TechCrunch.

The VoID specification[26] is an RDF Schema vocabulary for describing linked data sets. VoID is in-

[24] The RDF file is available at `http://km.aifb.kit.edu/sites/crunchbase/crunchbase-dump-201510.nt.gz`.

[25] I.e., `/api/organizations`, `/api/people` and `/api/products`.

[26] See `http://www.w3.org/TR/void/`, requested on Feb 3, 2016.

tended as a bridge between the publishers and users of RDF data. On our API entry page, we provide a VoID file for further usage.[27] Originally, the Crunchbase vocabulary would be a 1-star vocabulary according to Tim Berners-Lee's star rating about Linked Data, since the used vocabulary is only documented online in a human-readable way.[28] By providing an OWL-file, linking our vocabulary to schema.org, and creating an VOAF-File[29] we were able to increase the rating to 4 stars.

Our CrunchBase RDF data set is a 5-star data set, as we provide our data set in RDF and link entity URIs (organizations) to DBpedia and our vocabulary URIs to other vocabularies.

4. Usage

The presented CrunchBase Linked Data API and CrunchBase RDF data is useful in a variety of scenarios as CrunchBase provides data which is in most parts not covered by other Linked Open Data (LOD) data sets. We implemented the CrunchBase Linked Data API by means of the W3C Semantic Web standards RDF and JSON-LD; furthermore, we provide a schema description in OWL and a vocabulary description in VoID. All this allows the Linked Data API and the crawled data to be integrated in any Semantic Web application.

In the following, we elucidate concrete scenarios in which a CrunchBase Linked Data API or an RDF version of CrunchBase has been used.

4.1. Usage of the CrunchBase Linked Data API

The following RDF wrappers for CrunchBase have been developed and used so far:

- *Semantic CrunchBase*[30] is an RDF wrapper for CrunchBase, released by Nowack shortly after the release of the official CrunchBase JSON API in July 2008.[31] This initial CrunchBase wrapper transformed JSON provided by the CrunchBase API into RDF. However, no other data (such as `owl:sameAs` links) had been included and no external vocabulary (such as RDF, RDFS, or FOAF) had been used. The API is no longer available, but it shows that early efforts had been done for providing CrunchBase data in RDF.
- Harth et al. [2] demonstrated in 2013 an on-the-fly integration of static and dynamic sources for applications that consume Linked Data. Given the overall goal of supporting timely decision making, the scenario considered in the paper was given as follows: Assuming that a user wants to go to an event and that the user's current location is known, determine when the user should be at the next bus stop to arrive on time. Among the data sources, an RDF version of CrunchBase was integrated to include office locations of technology companies in the overall system. Harth et al. used a first version of the CrunchBase Linked Data API as presented in this paper.

4.2. Usage of the CrunchBase RDF Data Set

For our purpose, the Linked Data API was used to create a locally available Knowledge Base with CrunchBase data; loading this data in a triple store allowed us to perform SPARQL queries against the data and to use the results for further information retrieval tasks. RDF data sets of CrunchBase have been used in the following ways so far:

- Lee et al. [4] present an initiative of using Linked Data for financial data integration. According to the authors, the reported work has been well-accepted at several public events and conferences such as the 26th XBRL conference. The purpose of integrating CrunchBase and other financial data sources was to allow "both professional analysts and amateur individual investors to understand the performance of a particular company more efficiently." This was achieved by linking heterogeneous financial data and by tracing their provenance. Lee et al. show that the integrated RDF data allows a better comparison of financial reports, that it supports new KPI definitions, and that it allows timely access to external data.

[27] See `http://km.aifb.kit.edu/services/crunchbase/void.ttl`, requested on Feb 5, 2016.

[28] See `https://data.crunchbase.com/v3/docs/getting-started-1`, requested on Feb 1, 2016.

[29] See `http://km.aifb.kit.edu/services/crunchbase/voaf.ttl`, requested on Feb 5, 2016.

[30] See `http://bnode.org/blog/2008/07/29/semantic-web-by-example-semantic-crunchbase` and the dedicated host `http://cb.semso.org/`, which is not available any more; requested on Apr 6, 2016.

[31] See `http://techcrunch.com/2008/07/15/crunchbase-now-has-an-api-so-grab-our-data/`, requested on May 2, 2016.

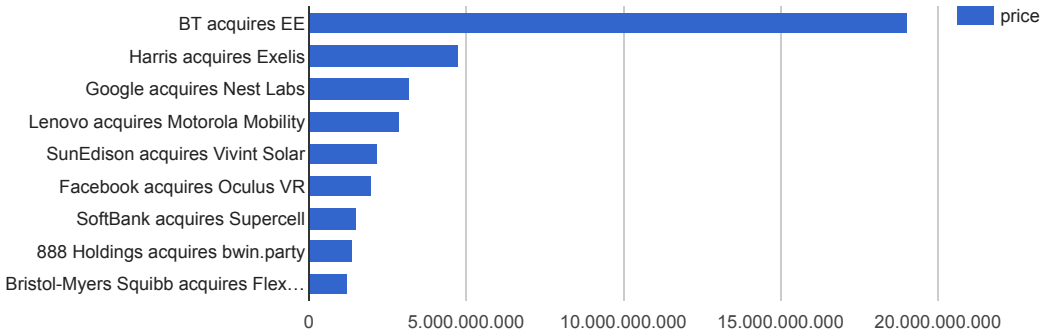

Fig. 4. Result of the query "Get all acquisitions of start-ups founded after 2010 with an price greater than 1 bn USD, sorted by price in descending order" as diagram; see also `http://km.aifb.kit.edu/sites/crunchbase/`.

Regarding CrunchBase, RDF data regarding funding, competitors, company acquisitions, main people in charge, and products were integrated into the framework. Hereby, a first version of our CrunchBase RDF data set was used by the authors. In the current article, we present an updated version of the CrunchBase RDF data set. The current CrunchBase RDF dump contains considerable more entities and more diverse entity types.

– In [1] we showed the usage of the CrunchBase RDF dump as it is presented in this article for the purpose of monitoring news to find statements which are not in a Knowledge Base so far. We thereby used the `owl:sameAs`-relations between CrunchBase entities and DBpedia entities in order to be able to use an existing entity linking module for linking mentions in text to CrunchBase entities. In the evaluation, we focused on relations between organizations and on relations between people and organizations; we demonstrated the following competencies:
  * The implemented system can detect facts, such as acquisitions, which are sometimes leaked, rumored or discussed publicly before they are officially announced. This is of great interest in media monitoring.
  * Given extracted new facts, the system provides a way to insert these facts to the Knowledge Base and in addition to link the facts to the news articles which the facts were extracted from. This option is interesting for people using a local CrunchBase Knowledge Base as well as for editors of the official CrunchBase platform.
  * For all facts stored in the CrunchBase Knowledge Base, the system can show when and how often these facts have been mentioned in the news. This feature can be used for tracking facts, e.g., in the context of beat reporting.

Besides using the CrunchBase RDF data set for data integration and for information extraction from text, it can also be used for data visualization and exploratory data analysis. Figure 4 shows an example visualization, given the natural language query "Get all acquisitions of start-ups founded after 2010 with an price greater than 1 bn USD, sorted by price in descending order." However, the CrunchBase RDF data set can be utilized not only by business people, but also by researchers such as of social studies. Xiang et al. [9] and Liang and Yuan [5] give an idea of that: They have made analyses on a CrunchBase data set which they created on their own. Liang and Yuan, for instance, used CrunchBase data to build a social network graph. Based on this graph, they applied various link prediction techniques in order to explore how similarity between investors and companies affects the investing behavior. One of their findings is that if investors and companies share too many common neighbors, investors are less likely to invest in such companies. It is one of the first studies on CrunchBase for social analysis. For creating their CrunchBase data set, Liang and Yuan [5] used Facebook as seed entity and then crawled entities which are at most four hops away. This procedure resulted in about 12,000 companies and 12,000 people. The CrunchBase RDF data set proposed in this article, in contrast, contains about 568,000 organizations (including companies) and 430,000 people. Thus, more comprehensive and in-depth studies are possible based on our data set. RDF provides thereby an intuitive data structure for analyzing characteristics of the graph (and to apply, for instance, link prediction algorithms).

5. Conclusions

We have presented a method for bringing the CrunchBase API to the Semantic Web. To that end, (i) we have implemented a Linked Data API as wrapper around the publicly available CrunchBase REST API; (ii) we have crawled the data from the wrapper for building a local CrunchBase data set. Both the Linked Data API and the RDF data set are freely available. To ensure the best possible usage and impact of the Linked Data API and RDF data set, we proceeded along Linked Data best practices such as describing the API, the RDF dump, and the schema via published OWL and VoID files, mapping CrunchBase relations and classes to relations and classes from other vocabularies, and integrating `owl:sameAs` links to entities in DBpedia. Our work can serve as blueprint for providing a Linked Data interface to other JSON APIs. Other people used our CrunchBase API and our crawled CrunchBase RDF data set for data integration purposes. Existing works show that CrunchBase RDF data can be applied for exploratory data analysis, too.

References

[1] M. Färber, A. Rettinger, and A. Harth. Towards Monitoring of Novel Statements in the News. In *Proceedings of the 13th Extended Semantic Web Conference (ESWC'16)*, Berlin, 2016. Springer.

[2] A. Harth, C. A. Knoblock, S. Stadtmüller, R. Studer, and P. Szekely. On-the-fly Integration of Static and Dynamic Linked Data. In *Proceedings of the 4th International Conference on Consuming Linked Data (COLD'13)*, pages 1–12. CEUR-WS. org, 2013.

[3] A. Hogan, P. Hitzler, and K. Janowicz. Linked Dataset Description Papers at the Semantic Web Journal: A Critical Assessment. *Semantic Web*, 7(2):1–13, 2016.

[4] V. Lee, M. Goto, B. Hu, A. Naseer, P.-Y. Vandenbussche, G. Shakair, and E. M. Rodrigues. Exploiting Linked Data in Financial Engineering. In *Service Science and Knowledge Innovation*, pages 116–125. Springer, 2014.

[5] Y. E. Liang and S. D. Yuan. Predicting investor funding behavior using crunchbase social network features. *Internet Research*, 26(1):74–100, 2016.

[6] M. Mochol, H. Wache, and L. Nixon. Improving the Accuracy of Job Search with Semantic Techniques. In *Proceedings of the 10th International Conference on Business Information Systems (BIS'07)*, pages 301–313. Springer, 2007.

[7] S. Stadtmüller, S. Speiser, A. Harth, and R. Studer. Data-Fu: A Language and an Interpreter for Interaction with Read/Write Linked Data. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*, pages 1225–1236. ACM, 2013.

[8] J. Weaver and P. Tarjan. Facebook Linked Data via the Graph API. *Semantic Web*, 4(3):245–250, 2013.

[9] G. Xiang, Z. Zheng, M. Wen, J. I. Hong, C. P. Rosé, and C. Liu. A supervised approach to predict company acquisition with factual and topic features using profiles and news articles on techcrunch. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM'12)*, pages 607–610, Dublin, Ireland, 2012. The AAAI Press.