Semantic Web 0 (0) 1–11
IOS Press

A Linked Data Wrapper for CrunchBase

Editor(s): Jens Lehmann, Universität Bonn & Fraunhofer IAIS, Germany
Solicited review(s): Konrad Höffner, Universität Leipzig, Germany; Marta Sabou, Technische Universität Wien, Austria; One anonymous reviewer

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 database 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. 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 access facilities of the CrunchBase API. Further, we describe the cases in which the Linked Data API for CrunchBase and the crawled CrunchBase RDF data have been used in other works.

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 the products they sell, key people they employ, and investments they made and received. CrunchBase is mainly used by entrepreneurs, investors, and business analysts to look up information for gaining market insights.[2]

Initially founded in 2007, CrunchBase is nowadays used "by millions of users"[3] 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*.[4] Given the graph-based data model, the CrunchBase data is in principle amenable to be modeled in RDF.

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 (companies, rounds, investments, and acquisitions) on the overall data.

[*] 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://about.crunchbase.com/partners/advertising-partners/`, requested on Oct 19, 2016.

[3] See `https://about.crunchbase.com/`, requested on Oct 16, 2016.

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

3. The *REST API* allows for accessing the entire contents of the CrunchBase database.

Noteworthy is the unusual licensing model of CrunchBase. On the one hand, all CrunchBase data is licensed partly under Creative Commons Attribution-NonCommercial License 4.0 (CC-BY-NC) and partly under Creative Commons Attribution License 4.0 (CC-BY), independent how it is provided (e.g., via browser or API).[5] This means that we are allowed to provide a data dump as well as a wrapper which uses CrunchBase data. On the other hand, for using the CrunchBase REST API, the user has to obtain an API key from the CrunchBase team. Presumably, CrunchBase wants to keep control over the usage of their infrastructure. There are different kinds of access granted. We use a free academic research access.

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.”[6]
   In contrast, many professional CrunchBase users may want to formulate more elaborate queries.[7] Such a query, formulated in natural language, might be: “Which companies in the category "Semantic Web" have got funded since 2000?”[8] 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, one 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. [8] 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

[5] See `https://about.crunchbase.com/docs/terms-of-service/`, requested Oct 19, 2016.

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

[7] The CrunchBase API mailing list provides examples for such requested queries, see, for instance, `https://groups.google.com/d/msg/crunchbase-api/xiAQdg5CAo4/GN51XIlptWMJ`, `https://groups.google.com/d/msg/crunchbase-api/k24Sy0tHOTo/7OrRJ3d6NXcJ`, and `https://groups.google.com/d/msg/crunchbase-api/g99E-Ft2aCk/cF89E44Z1egJ`; requested on Oct 17, 2016.

[8] This question is based on the CrunchBase mailing list post available at `https://groups.google.com/d/msg/crunchbase-api/xiAQdg5CAo4/GN51XIlptWMJ`, accessed on Oct 17, 2016.

Table 1

Links to resources.

Description	URI
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`
Ontology with links to external vocabularies:	`http://km.aifb.kit.edu/services/crunchbase/ontology.owl`
Visualizations based on SPARQL queries against CrunchBase RDF:	`http://km.aifb.kit.edu/sites/crunchbase/`

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 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 [4], five out of all 38 papers mention JSON as input data format, but only the description of the Facebook RDF Wrapper [10] and of LinkedSpending [3] describes a conversion of JSON to RDF. For the Facebook RDF Wrapper, JSON-LD was considered, but disregarded, "since its conventions varied too widely from the existing JSON format." [10]

Since the publication of [10], things have changed: JSON-LD became a W3C recommendation[9] in 2014. More and more developers use JSON-LD[10] 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 to access live data as RDF, and then create a data set via crawling. Such an approach allows for the collection of parts or all of the data, and provides up-to-date access to data about 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 Fig. 1. We first set up a simple RDF API to harvest an initial RDF data set via crawling. We use the initial RDF data set to enrich our Linked Data API with `owl:sameAs` links to DBpedia. The obtained links are 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. The code of the wrapper is available on GitHub[11] under the MIT license, and we maintain an instance of the wrapper.[12] 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 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

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

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

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

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

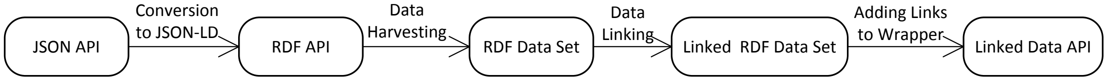

Fig. 1. Schematic view of the steps taken to create a Linked Data version of the CrunchBase API.

important for our news monitoring task. We crawled in October 2015 and retrieved 7,373,480 unique entities. The crawled CrunchBase RDF data set can be reused by all researchers who want to extend existing Knowledge Bases with CrunchBase data or who want to analyze the RDF data set for their own purposes.

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

We now give an overview of our implemented CrunchBase Linked Data API. Fig. 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 system Linked Data-Fu [9], calls the CrunchBase Linked Data API via a HTTP GET request. The request contains the URI, the requested content type, and the CrunchBase API user key.[13]
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 three different content types:

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)`: For providing data via our CrunchBase Linked Data API as JSON-LD, we restructure the JSON file retrieved from the official CrunchBase API. The main restructuring steps are removing meta-data and adding namespaces. Additionally, CrunchBase encapsulates properties (such as the date of birth of a person), relationships (such as the acquisitions of a company) and items in lists. To avoid blank nodes, we removed the list structure.
3. `RDF/N-Triples (text/turtle)`: We provide also N-Triples, a subset of the Turtle syntax for RDF, as one of the widely used formats in current Semantic Web systems.

Because 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. 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.[14] 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.

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

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

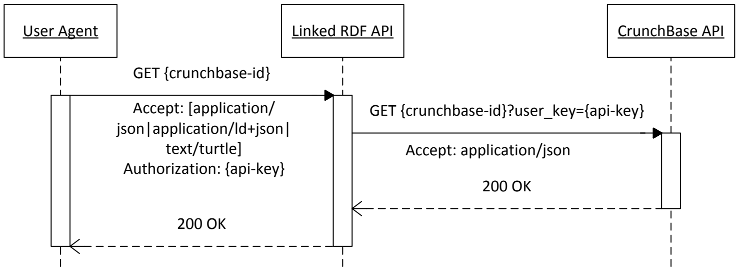

Fig. 2. UML sequence diagram illustrating the use of the wrapper. The wrapper supports different representations via content negotiation. The API key is passed to the wrapper via an `Authorization` header, and passed from the wrapper to the CrunchBase API via URI parameter.

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`

As the CrunchBase data is licensed under CreativeCommons licenses and can thus be reused,[15] 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 the header 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. Since the URIs for the official CrunchBase API[16] 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.

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 remain. Fig. 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)[17] 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.[18]

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 acquiree and an acquirer, but for instance also a date and a type of the acquisition. Events such as acqui-

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

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

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

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

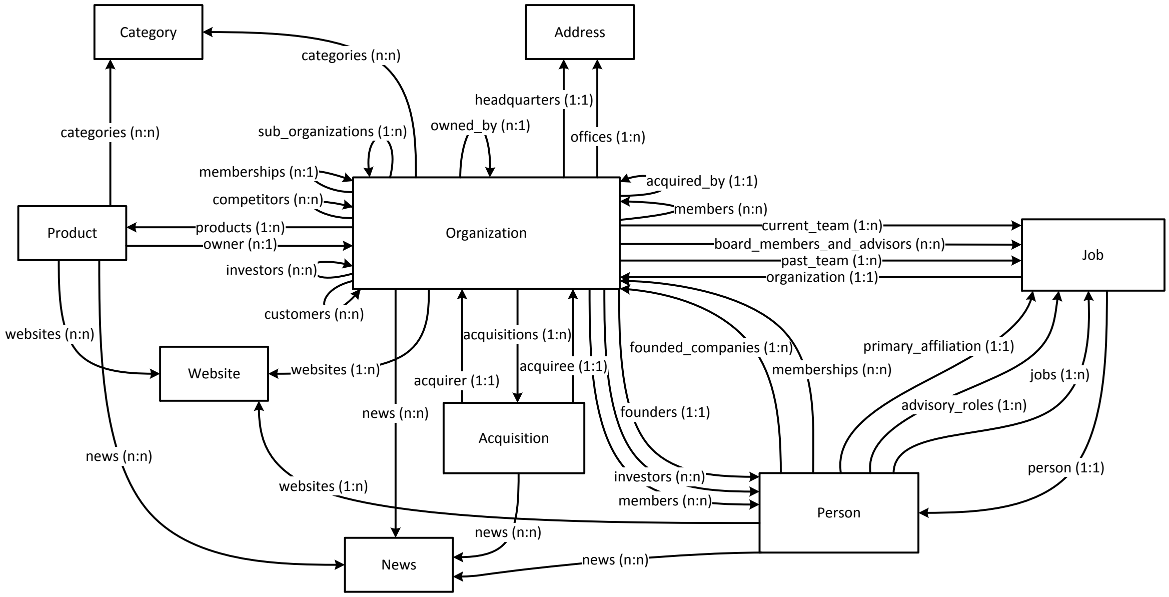

Fig. 3. Diagram showing a subset of the classes and relations supported by the CrunchBase Linked Data API. We use a (source, target) notation to indicate the cardinality of relations.

sitions are modeled as separate entities. In RDF, the concept of n-ary relations[19] is used to represent these entities. In our data model, 21 entity types are modeled as blank nodes due to that representation form.[20]

2. Noteworthy is also the possibility to model uncertainty for date values. The uncertainty value 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[21] such as `xsd:date` if they are valid.

Table 3

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

CrunchBase entity type	schema.org entity type[23]
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

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 to some degree domain-dependent, 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. We created 32 `equivalentProperty`, 16 `subPropertyOf`, 9 `subClassOf`, and 11 `equivalentClass` links. 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.[22]

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

[20] 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 the official CrunchBase API documentation.

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

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

2.4. Linking CrunchBase Entities to DBpedia

We created a list of `owl:sameAs` links between CrunchBase entities and the corresponding DBpedia entities to be able to include the corresponding `owl:sameAs` link in the returned data about an entity. We created mappings between 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. 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). One of the main reasons is that CrunchBase has no restrictions regarding the insertion of new entities or facts: Any contributor can add new entities. While the same also holds for Wikipedia, bots and Wikipedia contributors delete new entities if these entities do not seem to be of general public interest.[24] 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.

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. People entities, however, require higher effort for mapping. Using just the given name and surname leads to a very high rate (about 90%) 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).[25]

To evaluate the accuracy of this mapping strategy, 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 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.

Based on these results, we decided not to create `owl:sameAs` links between people in CrunchBase and DBpedia.

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 data set containing information about CrunchBase organizations, people, acquisition, and products (including their attributes and relations).[26] Our goal was to build a local Crunch-

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

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

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

Base 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. We restricted the crawling of CrunchBase data to the mentioned entity types as those are highly relevant for the scenario.

We built the CrunchBase RDF dump in conjunction with the information integration framework Linked Data-Fu [9] along the following steps:

1. We crawled all so-called summary data via the CrunchBase Linked Data API by using URIs for the summary lists of organizations, people, and products.[27] The summary list 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 the CrunchBase API spreads information across 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. We thus 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 the relation has more than eight different objects. Note that the API calls take time, as the Linked Data API is restricted to the same 1 second limit for requests that the official CrunchBase REST API has.

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 to entities of further entity types, our final CrunchBase RDF data set included 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.

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 VoID specification[28] is an RDF Schema vocabulary for describing linked data sets. VoID is intended as a bridge between the publishers and users of RDF data. On our API entry page, we provide a VoID file for further usage.[29]

There are two kinds of 5-star rating schemes in the Linked Data context:

- *The 5-star deployment scheme for Open Data developed by Tim Berners-Lee:*[30] Our CrunchBase RDF data set is a 5-star data set according to this scheme, since we provide our data set in RDF (leading to 4 stars) and link entity URIs (organizations) to DBpedia and our vocabulary URIs to other vocabularies (leading to 5 stars).
- *Linked Data vocabulary star rating [5]:* This rating is intended to rate the use of vocabulary within Linked (Open) Data. By providing an OWL-file, linking our vocabulary to schema.org, and creating a Vocabulary-of-a-Friend (VOAF) file,[31] we are able to provide the CrunchBase vocabulary with 4 stars.

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

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

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

[30] See `http://5stardata.info/`, requested on Nov 3, 2016.

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

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*[32] is an RDF wrapper for CrunchBase, released by Nowack shortly after the release of the official CrunchBase JSON API in July 2008.[33] 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. 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

RDF data sets of CrunchBase have been used in the following ways so far:

- Lee et al. [6] present an initiative of using Linked Data for financial data integration. 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.
  Regarding CrunchBase, RDF data about funding, competitors, company acquisitions, main people in charge, and products were integrated into the framework. The authors of [6] used a first version of our CrunchBase RDF data set. In the current article, we present an updated version of the CrunchBase RDF data set, which 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 used the `owl:sameAs`-relations between CrunchBase entities and DBpedia entities to be able to use an existing entity linking module for linking mentions in text to CrunchBase entities.

The CrunchBase RDF data set can also be used for data visualization and exploratory data analysis. Fig. 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." 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. [11] and Liang and Yuan [7] 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 the graph, they applied various link prediction techniques 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, in-

[32] 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.

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

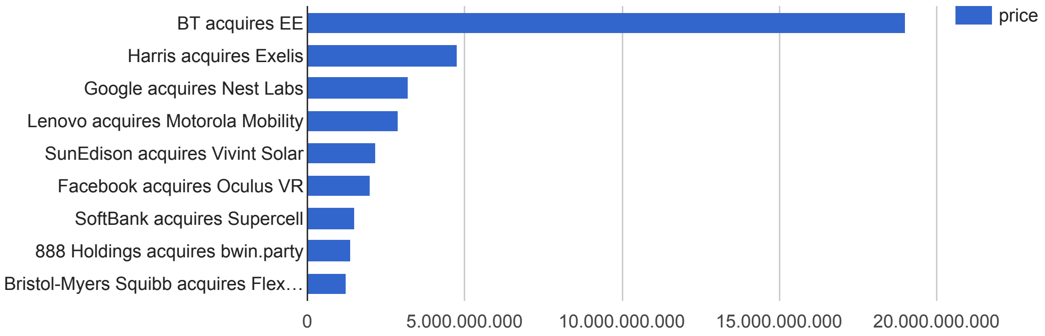

Fig. 4. Result of a SPARQL query based on the search intent "Get all acquisitions of start-ups founded after 2010 with an price greater than 1 bn USD, sorted by price in descending order" depicted as diagram; see also `http://km.aifb.kit.edu/sites/crunchbase/`.

vestors are less likely to invest in such companies. For creating their CrunchBase data set, Liang and Yuan [7] 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.

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. In the future, we intent work on a better linkage to DBpedia and on semantic question answering on top of the CrunchBase RDF data set.

References

[1] Michael Färber, Achim Rettinger, and Andreas Harth. Towards monitoring of novel statements in the news. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange, editors, *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*, volume 9678 of *Lecture Notes in Computer Science*, pages 285–299. Springer, 2016. DOI https://doi.org/10.1007/978-3-319-34129-3_18.

[2] Andreas Harth, Craig A. Knoblock, Steffen Stadtmüller, Rudi Studer, and Pedro A. Szekely. On-the-fly integration of static and dynamic sources. In Olaf Hartig, Juan Sequeda, Aidan Hogan, and Takahide Matsutsuka, editors, *Proceedings of the Fourth International Workshop on Consuming Linked Data, COLD 2013, Sydney, Australia, October 22, 2013*, volume 1034 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013. URL http://ceur-ws.org/Vol-1034/HarthEtAl_COLD2013.pdf.

[3] Konrad Höffner, Michael Martin, and Jens Lehmann. LinkedSpending: OpenSpending becomes Linked Open Data. *Semantic Web*, 7(1):95–104, 2016. DOI https://doi.org/10.3233/SW-150172.

[4] Aidan Hogan, Pascal Hitzler, and Krzysztof Janowicz. Linked dataset description papers at the Semantic Web journal: A critical assessment. *Semantic Web*, 7(2):105–116, 2016. DOI https://doi.org/10.3233/SW-160216.

[5] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman. Five stars of Linked Data vocabulary use. *Semantic Web*, 5(3):173–176, 2014. DOI https://doi.org/10.3233/SW-140135.

[6] Vivian Lee, Masatomo Goto, Bo Hu, Aisha Naseer, Pierre-Yves Vandenbussche, Gofran Shakair, and Eduarda Mendes Rodrigues. Exploiting linked data in financial engineering. In Kecheng Liu, Stephen R. Gulliver, Weizi Li, and Changrui Yu, editors, *Service Science and Knowledge Innovation - 15th IFIP WG 8.1 International Conference on Informatics and Semiotics in Organisations, ICISO 2014, Shanghai, China, May 23-24, 2014. Proceedings*, volume 426 of *IFIP Advances in Information and Communication Technology*, pages 116–125. Springer, 2014. DOI https://doi.org/10.1007/978-3-642-55355-4_12.

[7] Yuxian Eugene Liang and Soe-Tsyr Daphne Yuan. Predicting investor funding behavior using CrunchBase social network features. *Internet Research*, 26(1):74–100, 2016. DOI https://doi.org/10.1108/IntR-09-2014-0231.

[8] Malgorzata Mochól, Holger Wache, and Lyndon J. B. Nixon. Improving the accuracy of job search with semantic techniques. In Witold Abramowicz, editor, *Business Information Systems, 10th International Conference, BIS 2007, Poznan, Poland, April 25-27, 2007, Proceedings*, volume 4439 of *Lecture Notes in Computer Science*, pages 301–313. Springer, 2007. DOI https://doi.org/10.1007/978-3-540-72035-5_23. URL http://dx.doi.org/10.1007/978-3-540-72035-5_23.

[9] Steffen Stadtmüller, Sebastian Speiser, Andreas Harth, and Rudi Studer. Data-Fu: A language and an interpreter for interaction with read/write linked data. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo A. Baeza-Yates, and Sue B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1225–1236. International World Wide Web Conferences Steering Committee / ACM, 2013. URL http://dl.acm.org/citation.cfm?id=2488495.

[10] Jesse Weaver and Paul Tarjan. Facebook linked data via the Graph API. *Semantic Web*, 4(3):245–250, 2013. DOI https://doi.org/10.3233/SW-2012-0078.

[11] Guang Xiang, Zeyu Zheng, Miaomiao Wen, Jason I. Hong, Carolyn Penstein Rosé, and Chao Liu. A supervised approach to predict company acquisition with factual and topic features using profiles and news articles on TechCrunch. In John G. Breslin, Nicole B. Ellison, James G. Shanahan, and Zeynep Tufekci, editors, *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*. The AAAI Press, 2012. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/view/4621.