Maximilian Eibl, Martin Gaedke (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017

Serving Live Multimedia for the Linked Open Data Cloud

Sebastian Serth, Stephan Haarmann and Lukas Faber[1]

Abstract: DBpedia is a community-driven project to extract semantic data from Wikipedia articles. It publishes the results as RDF data in the Linked Open Data Cloud. With DBpedia Live, the community enabled live updates of linked data using the OAI-PMH protocol to receive and process changes on Wikipedia. The MediaWiki foundation discontinued their support for OAI-PMH in March 2016 causing DBpedia Live to no longer receive live updates. In this work, we use RCStream, the new MediaWiki protocol to notify other systems of changes, to re-enable live updates in DBpedia Live. Currently, users need to consume two DBpedia resources to access general information and multimedia files about one entity. On the one hand DBpedia holds the structured information. On the other DBpedia Commons holds most multimedia information. We improve the integration of multimedia data into DBpedia by introducing a new extractor to the DBpedia Extraction Framework that extracts most multimedia data from a Wikipedia page. Additionally, we present two further extractors that link pages in DBpedia with pages in DBpedia Commons and vice versa. All our changes are available in the DBpedia Extraction Framework and in use, e.g. for DBpedia Live.

Keywords: Linked Data, DBpedia Live, DBpedia Commons, Wikipedia, Wikimedia Commons

1 Introduction

The Linked Open Data Cloud (LODC) is a set of data providers in the World Wide Web. Their data is publicly available for use to everyone and usually contains references to other Linked Open Data sources. DBpedia is one of the key providers of such data [Le15]. It extracts structured information from three dozen language versions of Wikipedia and publishes it as semantic data in RDF format. Wikipedia itself is the largest online encyclopaedia and the fifth most visited page on the internet[2]. It contains about 44.1 Million pages in over 100 languages and its users edit articles approximately 10.8 Million times a month[3]. To keep up with the fast-changing data on Wikipedia the community of DBpedia created DBpedia Live. This uses a technology called *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) [Op15] to synchronize through a local MediaWiki mirror with Wikipedia. DBpedia Live receives events when pages on Wikipedia change and extracts these pages again. In March 2016, the Wikimedia foundation discontinued their support

[1] Hasso Plattner Institute, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany {sebastian.serth, stephan.haarmann, lukas.faber}@student.hpi.de

[2] `http://www.alexa.com/topsites` — Alexa rank on the most visited pages on the internet

[3] `https://stats.wikimedia.org/EN/TablesWikipediaZZ.htm` — Statistics on Wikipedia as of March 2017

of OAI-PMH meaning that DBpedia Live can no longer receive notifications of changes this way. Instead, the Wikimedia foundation offers a technology called *Recent Changes Stream* (RCStream) [Me16b]. In this work, we integrate RCStream into DBpedia Live to make it aware of changes on Wikipedia. This change allows a simplification of the architecture of the DBpedia Live system and does not require the local MediaWiki mirror any longer. Even when running live, DBpedia Live hardly contains multimedia data that the equivalent Wikipedia article does contain. DBpedia Commons [Va15] is a DBpedia instance that stores multimedia data while the other instances focus on textual data. The stored multimedia files are retrieved from Wikimedia Commons. In the current state, corresponding resources on DBpedia Commons and other DBpedia instances do not link to each other. Thus, consumers of both resources have to connect these resources on their own. Our goal was to ease the exploration of corresponding multimedia files with a Creative Commons license. Therefore, we show two ways to create connections between DBpedia Commons and other DBpedia instances. First, we link the corresponding resources on both systems with each other. Second, when extracting a Wikipedia page we also extract most multimedia files and link the equivalent DBpedia Commons resources for those.

Our source code containing the proposed changes has been merged with the open-source DBpedia Extraction Framework[4] and in use today to run DBpedia Live with additional support for the enhanced linkage between DBpedia Live and DBpedia Commons.

In section 2 we introduce how existing work led to DBpedia Live and DBpedia Commons. We present the changes required when switching from OAI-PMH to RCStream in section 3 and show our approach to link semantic resources of DBpedia and DBpedia Commons in section 4. We create links between correlating resources (e.g., between `http://dbpedia.org/resource/Eurasian_blue_tit` and `http://commons.dbpedia.org/resource/Cyanistes_caeruleus`). Additionally, we directly link the media resources in DBpedia Commons to the resource in DBpedia. In section 5 we examine the performance and the extracted triples together with the benefits and shortcomings of the solution. Section 6 summarizes the results of our work and shows paths for future development.

2 Related Work

DBpedia has been initiated eleven years ago in 2006 [AL07]. Lehmann and others [Le15] summarize the development that has been done in the nine years until 2015. In this section, we focus on work regarding DBpedia Live and the inclusion of multimedia data.

Suh and others extract “Common Sense Knowledge” from Wikipedia texts using natural language processing and they publish the information as RDF data [SHK06]. Conversely, Auer and others extract semantic data from structured information in Wikipedia [AL07]. Based on this work, Auer and others [Au07] introduce the DBpedia as a storage of semantic

[4] `https://github.com/dbpedia/extraction-framework` — DBpedia Extraction Framework with our changes

information from Wikipedia. They acquire the pages containing the information by parsing the periodically created Wikipedia dumps. Thus, the DBpedia instance created on this dump is partly outdated and misses further changes made to Wikipedia [He09]. Hellmann and others use OAI-PMH to receive change events from Wikipedia and process them to create DBpedia Live. Morsey and others refine the Extraction Framework by adding a tool to continuously import new change sets, additional extractors and more [Mo12].

Garcia and others propose a solution to extend the data from DBpedia with multimedia content. They use machine learning and search engines to find relevant media on the web [Ga11]. Vaidya and others instead focus on images that have been associated by humans to Wikipedia topics leveraging the media from Wikimedia Commons [Va15]. They extend the Extraction Framework with new extractors that are capable of extracting multimedia data from Wikimedia Commons and publish the result to DBpedia Commons. If consumers require the multimedia data used on a Wikipedia page, they need to link the related DBpedia resource to their equivalent on DBpedia Commons by themselves. Kontokastas and other combine information of different DBpedia instances [Ko12] and create links between different language versions but not to DBpedia Commons, which is our motivation to ease access to multimedia information and their meta data.

OAI-PMH was developed by the Open Archives Initiative [Op15] and is used in a variety of use cases to exchange metadata[5]. The Wikimedia Foundation discontinued their usage of OAI-PMH. It developed their own technology called RCStream [Me16b] specifically for notifying subscribers of changes in a MediaWiki instance.

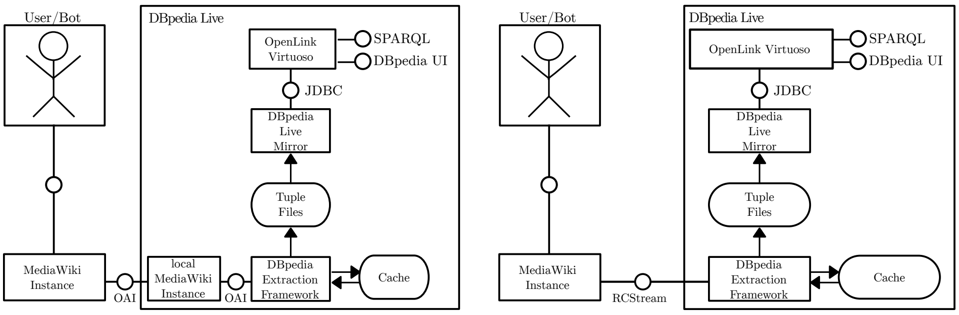
3 Method and Approach to Make DBpedia Live Again

We integrate our RCStream-based solution into the current DBpedia Live architecture, which is depicted in Figure 1a (see also [He09]). An OAI-PMH event notifies the DBpedia Live system, whenever a user or a bot makes changes. OAI-PMH includes the whole content of the changed wiki page in the event thus creating a high load on the MediaWiki instance. Therefore, DBpedia Live contains an additional local mirror of that specific MediaWiki that forwards events to the DBpedia Extraction Framework. It then extracts structured information from the page content and creates a so called changeset in the form of triple files. The changeset contains the triples which need to be added or deleted in the RDF storage (OpenLink Virtuoso). An additional program, the DBpedia Live mirror[6], loads these triples and imports them into Virtuoso.

The DBpedia Extraction Framework contains a set of components that trigger the live extraction of pages in the Wikipedia. These components are called "feeders". One example for such a feeder is the `OAIFeeder` that listens for changed pages on Wikipedia using OAI-PMH. All feeders publish their events to an intermediate internal queue. From there, page

[5] `http://www.openarchives.org/Register/BrowseSites/` — A repository of OAI data providers

[6] `https://github.com/dbpedia/dbpedia-live-mirror` — source for the DBpedia Live mirror



(a) A FMC model describing the architecture of the Extraction Framework using OAI-PMH

(b) A FMC model describing the architecture of the Extraction Framework using RCStream

Fig. 1: The high-level architecture of the extraction framework

processors take pages and process them. If a feeder did not add the wiki page content to it, the processor first uses the MediaWiki API to query it. Then, the page content is given to several mutually independent components, so called “Extractors”. Every extractor can use the content of the whole page to extract triples, for example the `AbstractExtractorWikipedia` extracts the abstract of a wiki text. The processor then collects all triples of all extractors. If the page has been extracted before, the Extraction Framework compares the produced triples with those of the previous extraction by using a cache. Based on the comparison, it creates a minimal set of changes needed to update the triple store to the current data.

3.1 RCStream Integration

Recent versions of MediaWiki switched their update mechanism from OAI-PMH to RCStream, a new technology that provides notifications about events (such as editing a page). One can subscribe to the RCStream of a MediaWiki instance by establishing a socket connection. All future events are broadcasted to each subscriber through their individual connection. An RCStream event is a JSON object whose content describes meta data about the change as well as the page on which the event occurred [Me16b].

Our RCStream based solution integrates as a feeder component called `RCStreamFeeder` with the Extraction Framework. It establishes a socket connection to the MediaWiki instance and feeds incoming events into the internal queue of the Live Extraction Framework. In contrast to OAI-PMH, events from RCStreams do not contain the page content but only the title of the changed page. Consequently, the Extraction Framework queries the page content at a later step in the process. Amongst others, an RCStream event contains the attribute *type*, whose value encodes the cause for the event. For our case, only events of type *edit* (a page has been modified) or *new* (a new page has been created) are relevant. In addition, only a small subset of the received `namespaces` are relevant, which differs on the MediaWiki instance (for example the “File” namespace is relevant for Wikimedia Commons but not for any other Wikipedia instance) Based on these attributes we can filter events before processing them, thus reducing network traffic and increasing performance significantly.

The use of RCStream allows an architectural simplification of the DBpedia Live system. Figure 1b gives an overview of the architecture of the Extraction Framework using RCStream. RCStreams produce far less load on the MediaWiki instance compared to OAI-PMH, because the events do not contain the page content. Because of that, the additional MediaWiki mirror is no longer required. However, the Extraction Framework has to use the MediaWiki API to query the page content later in the process.

3.2 AllPagesFeeder

When running DBpedia Live, it is important to keep the triple store synchronised with the internal cache, which is used to create the changesets, at all times. The triple store also needs to be synchronised with the MediaWiki instance. This poses problems when creating a new DBpedia Live instance. Currently, there are two ways to create a new instance of DBpedia Live. First, one can start an instance with a dump based extraction using the DBpedia Extraction Framework. However, this process only updates the triple store but not the live cache. This can lead to inconsistencies like duplicated triples with contradicting values. Nor does this approach take changes into account that happened after the creation of the dump. Second, one can start with an empty graph and only insert the triples of pages for which the Extraction Framework received an event and produced triples for. This keeps the live cache and the triple store synchronised, but the triple store misses all pages that have never been edited. We propose the `AllPagesFeeder` uses the MediaWiki API to retrieve the names of all pages for a specifiable set of namespaces and puts all page titles in the queue for extraction. As a result, every page of the specified namespaces gets extracted (and not only those which have been edited), putting the information in both the triple store and the cache. In addition, the `AllPagesFeeder` might also be used to re-sync a DBpedia instance after an outage or connection loss. This is required because RCStreams do not store any state for their connections and thus do not offer any way to retrieve changes during a downtime.

4 Extracting and Linking Multimedia Data

The current extraction for a Wikipedia page is limited to one image which is extracted by the `ImageExtractor`. It parses the wiki page to find the first image and determines if the file also exists on Wikimedia Commons. To do so, it tests if the image is not present in a local image dump for the specified Wikipedia instance. This check determines the usage rights for that image – files on Wikimedia Commons are licensed under Creative Commons or a public domain licence. In contrast, files that are stored on Wikipedia have non-free usage rights. Either way the image is added to the extraction as a thumbnail. We see five drawbacks in this approach: a) No link to a possible corresponding resource in DBpedia Commons is extracted. b) The image extraction uses an image dump, which can become outdated. c) The extraction might add images that are non-free images. This could lead to copyright infringement. d) Only the first image of a Wikipedia page is extracted. e) Audio and video files are ignored.

4.1 Extracting Multimedia files from an Wikipedia Article

We introduce the `MediaExtractor` to extract multimedia information from pages on Wikipedia and include them in DBpedia. The DBpedia Extraction Framework parses the Wikipedia page content given in wiki markup and constructs a tree structure out of it. This structure is given to all extractors. In a first step the `MediaExtractor` extracts file names out of this tree structure. Files can be embedded with specific templates or within galleries. The `MediaExtractor` parses the respective markup and extracts the file information.

Second, the `MediaExtractor` checks for every file if an equivalent file exists on Wikimedia Commons. An equivalent file can be found under the same URL as on Wikipedia if one replaces `en.wikipedia.org` (in case of the English Wikipedia) with `commons.wikimedia.org`. Files not located on Wikimedia Commons are often not free and therefore discarded.

In a third step the `MediaExtractor` generates triples for every file that it finds on Wikimedia Commons with the newly introduced property `dbo:mediaItem`. The object of the triple is the URI of the file resource in DBpedia Commons. For every image file (ending in .svg, .jpeg, .jpg, .png, .gif), the `MediaExtractor` creates an additional triple with the property `rdf:type` and the object `dbo:Image`. Audio files can be unambiguously identified if the file name ends with one of the following suffixes: .flac, .wav, .midi, .kar, .opus, .spx. Additionally, we consider files ending with .ogg as audio files because until now 92.6% of all .ogg files uploaded to Wikimedia Commons are classified as sound files [Me16c]. The `MediaExtractor` adds a triple with the property `rdf:type` and the object `dbo:Sound` for each file. Video files are identified by the `MediaExtractor` based on the .webm suffix. This file type can also be used to describe audio files, but until now it has only been used for video data [Me16c]. Video files do not have a corresponding `rdf:type`, thus the `MediaExtractor` does not add any additional triples.

4.2 Enhancing the Connection Between DBpedia and DBpedia Commons

We improved the connection to DBpedia Commons by adding further multimedia references to the DBpedia resources in subsection 4.1. Typically, a page on Wikimedia Commons contains more images than the ones that are also used in Wikipedia pages. These files are all extracted to DBpedia Commons. Consumers of a DBpedia resource cannot easily consume this information because the DBpedia resource does not link the resource on DBpedia commons. The same holds true for consumers of DBpedia Commmons. For example, there is currently no link between the DBpedia resource `https://en.wikipedia.org/wiki/Eurasian_blue_tit` and the resource on DBpedia Commons `https://commons.wikimedia.org/wiki/Cyanistes_caeruleus` and vice versa.

Some Wikipedia pages link to a page with the same subject on Wikimedia Commons. This page is independently extracted to a resource in DBpedia Commons. Wikipedia pages create this link to Wikimedia Commons with a `{{Commons}}` template. We created the `CommonsResourceExtractor` as an extractor in the Extraction Framework that detects this template, which always contains a property specifiying the name of the page on Wikimedia Commons. For example, the Wikipedia page for the Blue Tit contains the template `{{Commons | Cyanistes_caeruleus}}`. This means that there exists a page with the URL `https://commons.wikimedia.org/wiki/Cyanistes_caeruleus` on Wikimedia Commons. Therefore, the framework creates a DBpedia Commons resource with the URI `http://commons.dbpedia.org/resource/Cyanistes_caeruleus` for this page. Hence, the `CommonsResourceExtractor` creates a triple with the property `owl:sameAs` and the object `http://commons.dbpedia.org/resource/Cyanistes_caeruleus`.

Conversely, some pages on Wikimedia Commons contain a `{{VN}}` template that allows to establish a link to Wikipedia pages with the same subject. This template lists the vernacular names for the entity described on the page. The `DBpediaResourceExtractor` uses this template to link resources on DBpedia Commons to their counterparts in other DBpedia instances. It reads each of the language properties (such as "de" or "en"), their respective value contains the possible name of the corresponding page in the Wikipedia instance in that language. For example the German equivalent for the page `https://commons.wikimedia.org/wiki/Cyanistes_caeruleus` would be `https://de.wikipedia.org/wiki/Blaumeise` and the English would be `https://en.wikipedia.org/wiki/Eurasian_blue_tit`. Therefore, the language specific Extraction Frameworks create resources with the URIs `http://de.dbpedia.org/page/Blaumeise` and `http://dbpedia.org/page/Eurasian_blue_tit`. The `DBpediaResourceExtractor` creates one triple for every language with the property `owl:sameAs` and the respective DBpedia resource as an object. At the moment, we only create such links for the English, German and French DBpedia.

5 Evaluation

The change from OAI-PMH has two advantages. First, the Extraction Framework does no longer need a local MediaWiki mirror as shown in Figure 1a. This saves a significant amount of memory as the mirror needs to contain all pages that the public Wikipedia also contains (for example over 5,000,000 pages for the English Wikipedia plus redirects). Second, the elimination of the mirror and the smaller event size of RCStream events allows for a faster event transfer. For our evaluation, we were not able to compare our approach with a running OAI-PMH version because the Wikimedia Foundation discontinued OAI-PMH in March 2016. Usually, more than a minute passed between the point where a user edited a page and the point where the Extraction Framework received the corresponding change event when using OAI-PMH [He09]. Conversely, the Extraction Framework receives change events almost instantaneously when using RCStream. However, RCStream events do not contain the page content which has to be downloaded separately in order to perform the extraction.

In this section, we evaluate the performance of the extraction (subsection 5.1) and examine the enhanced linkage between DBpedia and DBpedia Commons (subsection 5.2).

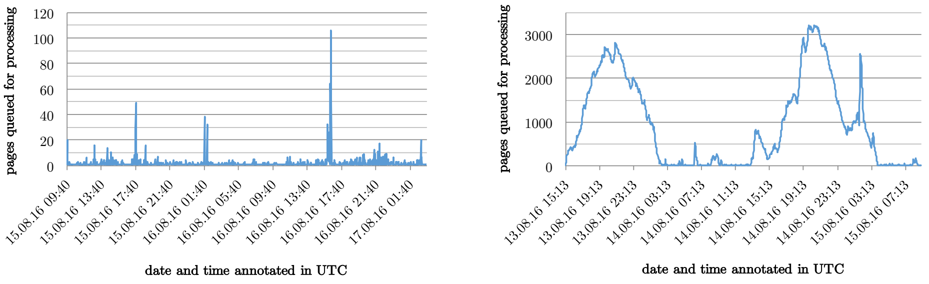
5.1 Performance Evaluation

Every DBpedia Live instance uses the Extraction Framework to create changesets. In our measures, we exclude the import of changesets into the triple store because this is a periodical task that can last zero seconds up to five minutes. When processing an event, we assume that the dominant factor in processing is the time needed for network traffic (such as querying the Wikipedia API for the page content). Therefore, when we discuss the efficiency of some workflow or optimization, we focus on network traffic.

Contrary to OAI-PMH, RCStream events contain only the affected page's title but not its ID. When we first integrated a RCStream-based change mechanism, we initially transformed the title to the page ID using the Wikipedia API. As introduced before, network traffic dominates the time needed to process these events. Therefore, we changed the Extraction Framework to also support events that only provide the page title. Thereby, we bisected the network traffic. Furthermore, we filter out events that are not of type *new* or *edit*. This allows us to discard 40% of all incoming events before they cause any network traffic. In the case of running DBpedia Live for the English Wikipedia, further 25% of events can be filtered out by only processing relevant pages, which are in the *Main*, *Template* or *Category* namespace. In total, the network traffic gets reduced to 22%.

When running DBpedia Live we need to ensure that events are faster processed than they arrive. Otherwise, the queue of items to be processed would grow until the system runs out of memory. During our tests, we measured the length of the internal queue that the DBpedia Extraction Framework queues events before it processes them. We conducted our measures inside a Docker image[7]. The Docker host runs Ubuntu 16.04 LTS inside a Hyper-V Virtual Machine on Windows Server 2012 R2. The physical system has two Intel Xeon E5-2630 v3 processors with a base speed of 2.40 GHz each and a total of 64 GB main memory. The Ubuntu VM has eight exclusive virtual processors (equals 25% of the total system resources), up to 16 GB RAM and was not running any other non-system task besides the specified Docker container. The available internet connection allows up to 150 Mbit/s and has an average ping of 26ms (based on 5,000 requests) to the Wikipedia servers. Other services and devices use up to 10% of this connection. We ran three configurations of the Extraction Framework for the English Wikipedia in this environment (see also Table 1):

[7] `https://github.com/semmul2016group4/docker_for_commons_live_extraction` — Dockerfile for our tests



(a) Extraction using the `RCStreamFeeder` without the `MediaExtractor`.

(b) Extraction using the `RCStreamFeeder` and the `MediaExtractor`.

Fig. 2: Pages queued for processing over time.

1. *without MediaExtractor / AllPagesFeeder.* The extraction framework processes events online. Rare peaks can be observed (see Figure 2a) that are probably caused by increased activity on Wikipedia.

2. *with MediaExtractor.* Using the `MediaExtractor` slows the processing of individual events down. If an increased amount of events occurs, the framework requires significantly longer to process the queue. On average it takes 15*min* until the changes of an event are adopted.

3. *with AllPagesFeeder and MediaExtractor.* The `AllPagesFeeder` pushes all pages (about 5.4 Million articles and 7.7 Million redirects[8]) once. The Framework then processes approximately 153 items per minute. Consequently, it would take about 66 days to process all pages in the English Wikipedia. Redirects take on average 28*ms*, articles 360. One drawback is, that new change events will be processed after all items of the `AllPagesFeeder`. Different priorities are a possible workaround.

Configuration	**queue after initialization**	**max queue**	**avg delay**	**avg processing time**
`RCStreamFeeder`	20	107	2.58*s*	229*ms*
additional `MediaExtractor`	20	3, 220	911*s*	757*ms*
additional `AllPagesFeeder`	13.1*M* (5.4*M* articles / 7.7*M* redirects)	13.1*M*	-	360*ms* / 28*ms*

Tab. 1: Key performance measurements of the Extraction Framework

The throughput can be improved by processing multiple items in parallel. When going for this solution, one must take care to combine the parallel request towards the Wikipedia API to stay compliant with its usage guidelines[9].

[8] `https://stats.wikimedia.org/EN/TablesWikipediaEN.htm`

[9] `https://www.mediawiki.org/wiki/API:Etiquette` — API etiquette for Wikipedia APIs

5.2 Enhanced Linkage

In this subsection, we measure how effective our means to link additional multimedia have been. We examine how many images the `MediaExtractor` adds to the extraction of a Wikipedia page. We also measure how many `sameAs` relationships the `CommmonsResourceExtractor` and the `DBpediaResourceExtractor` add.

The `MediaExtractor` extracts additional links between DBpedia resources and multimedia resources on DBpedia Commons. The number of newly created links entirely depend on the page content in Wikipedia. Some pages such as the Eurasian Blue Tit[10] include many multimedia files (image, audio and video files). The `MediaExtractor` extracts most of them. Other pages like Oriolidae[11] only contain one image, which is already considered in the current version of the DBpedia Extraction Framework, thus no additional multimedia files are linked. However, even in this case, the `MediaExtractor` adds a link to the file resource of the file in DBpedia Commons which was unavailable before (in addition to the file link to Wikimedia Commons). Consumers of the multimedia resources can also use these links as a starting point to explore DBpedia Commons for further multimedia data. Currently, the `MediaExtractor` does not extract files in some special galleries. This can be further improved, so we can ensure that every available multimedia information is parsed.

We measured the number of Wikipedia pages that the `CommonsResourceExtractor` extracts a `sameAs` triple for. To do so, we used the Wikipedia API to determine how many pages contain a `{{Commons}}` template. In a first step, we took a sample of the alphabetically first 5,000,000 pages whose titles come after the letter 'A'. For each of these pages we queried the Wikipedia API to get all templates used on this page. We count all pages that contain a template with the name "Template:Commons" and all pages that do not contain a template with the name "Template:Redirect template". The latter indicates that the page is a redirect page, which does not have any content. In our measures (on the 30th August 2016) we found out that the ratio of Commons templates on the English Wikipedia is $\frac{number\ of\ pages\ with\ commons\ template}{number\ of\ pages\ without\ redirect\ template} = \frac{11415}{4971671} \approx 0.23\%$. One reason for this low coverage is that many pages on Wikipedia reference a page in the `Category` namespace (such as the Tufted Titmouse[12]). Or they do not reference a page in Wikimedia Commons at all (such as the Grey Crested Tit[13]) even though a category page exists[14]. In this state users can hardly rely on this link to exist, so it would be essential to improve the percentage of sites the `CommonsResourceExtractor` extracts a link for.

We measured the approximate number of pages that the `DBpediaResourceExtractor` extracts at least one `sameAs` triple for. We took a similar approach to the measures for the `CommonsResourceExtractor`. We query for the alphabetically first 5,000,000 pages after the

[10] `https://en.wikipedia.org/wiki/Eurasian_blue_tit` — Wikipedia article about Eurasian blue tit
[11] `https://en.wikipedia.org/wiki/Old_World_oriole` — Wikipedia article about Oriolidae
[12] `https://en.wikipedia.org/wiki/Tufted_titmouse` — Wikipedia page of the Tufted Titmouse
[13] `https://en.wikipedia.org/wiki/Grey_crested_tit` — Wikipedia page of the Grey Crested Tit
[14] `https://commons.wikimedia.org/wiki/Category:Lophophanes_dichrous` — Grey Crested Tit in Commons

letter 'A'. However, there are only a total of 215,565 pages (as of 30th August 2016). We assume the number to be smaller because most information in Wikimedia Commons is stored in the `Category` namespace and not in the `Main` namespace [Me16a]. We calculated the relative number of pages that contain a `{{VN}}` template while excluding pages with a redirect template. This yields an approximate ratio of $\frac{number\ of\ pages\ with\ VN\ template}{number\ of\ pages\ without\ redirect\ template} = \frac{3521}{215565} \approx 1.63\%$ of pages that the `DBpediaResourceExtractor` can extract at least one `sameAs` triple for. The ratio is this small because the `{{VN}}` template used to find corresponding pages on Wikipedia is mostly used for animals and should be extended by using other templates to create a more reliable linkage between the instances.

6 Conclusion and Outlook

The Linked Open Data Cloud is a set of semantic data providers on the internet. DBpedia is part of the Linked Open Data Cloud that offers structured data from Wikipedia in a semantic format. We re-enabled DBpedia Live, an instance of DBpedia that continuously synchronizes with Wikipedia to serve the latest data and changes. We integrated the ability to consume RCStream with the DBpedia Extraction Framework. RCStream feeds the titles of changed pages into the Extraction Framework which then re-extracts the affected pages. The resulting triples are imported into DBpedia Live to replace outdated information.
We also extended the Extraction Framework to improve the accessibility of multimedia information. To do so, we added an extractor for DBpedia which detects most multimedia files on a Wikipedia page and extracts these. Therefor, we created a new property `dbo:mediaItem` which links to the file resource on DBpedia Commons for all files whose licenses allow the usage in open data. We additionally created two extractors that link resources from DBpedia Commons and other DBpedia instances and vice versa based on templates found in the articles in Wikipedia and Wikimedia Commons, respectively.

In its current state, the `MediaExtractor` creates a significant increase in network traffic. As a consequence, events take longer to process than it takes new events to arrive. Therefore, an important aspect of future work would be to find ways to enable faster processing. While the `AllPagesFeeder` properly initializes a new DBpedia instance and resynchronizes it after a downtime, it basically blocks the extraction for other feeders. Future work might find more suitable ways to integrate the `AllPagesFeeder`. One possible approach could be to filter out redirect pages when resynchronizing after a downtime since such pages will not create any meaningful triples. Currently, multimedia files in category pages on Wikimedia Commons are not extracted and are therefore not available in DBpedia Commons. One could think of extracting this data for DBpedia Commons and improving the `CommonsResourceExtractor` by also taking category pages into account. This will increase the coverage of Wikipedia pages for which a link to a DBpedia Commons resource can be extracted. In addition, more templates (such as the `{VN}` for animals) should be added to the `DBpediaResourceExtractor` to add additional links to corresponding Wikipedia articles from Wikimedia Commons pages.

7 Acknowledgement

This work is based on the seminar *Semantic Multimedia* at the Hasso Plattner Institute. The authors would like to thank Prof. Dr. Harald Sack and Magnus Knuth for their support.

References

[AL07] Auer, Sören; Lehmann, Jens: What have innsbruck and leipzig in common? extracting semantics from wiki content. In: European Semantic Web Conference. Springer, pp. 503–517, 2007.

[Au07] Auer, Sören; Bizer, Christian; Kobilarov, Georgi; Lehmann, Jens; Cyganiak, Richard; Ives, Zachary: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer, 2007.

[Ga11] García-Silva, Andrés; Jakob, Max; Mendes, Pablo N; Bizer, Christian: Multipedia: enriching DBpedia with multimedia information. In: Proceedings of the sixth international conference on Knowledge capture. ACM, pp. 137–144, 2011.

[He09] Hellmann, Sebastian; Stadler, Claus; Lehmann, Jens; Auer, Sören: DBpedia live extraction. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, pp. 1209–1223, 2009.

[Ko12] Kontokostas, Dimitris; Bratsas, Charalampos; Auer, Sören; Hellmann, Sebastian; Antoniou, Ioannis; Metakides, George: Internationalization of linked data: The case of the greek dbpedia edition. Web Semantics: Science, Services and Agents on the World Wide Web, 15:51–61, 2012.

[Le15] Lehmann, Jens; Isele, Robert; Jakob, Max; Jentzsch, Anja; Kontokostas, Dimitris; Mendes, Pablo N; Hellmann, Sebastian; Morsey, Mohamed; van Kleef, Patrick; Auer, Sören et al.: DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2):167–195, 2015.

[Me16a] MediaWiki: , Help:Namespaces, 2016. Online: current as of August 30th, 2016.

[Me16b] MediaWiki: , Manual:RCFeed, 2016. Online: current as of June 21st, 2017.

[Me16c] Mediawiki: , Media statistics, 2016. Online: current as of August 28th, 2016.

[Mo12] Morsey, Mohamed; Lehmann, Jens; Auer, Sören; Stadler, Claus; Hellmann, Sebastian: Dbpedia and the live extraction of structured data from wikipedia. Program, 46(2):157–181, 2012.

[Op15] Open Archives Initiative: , Open Archives Initiative Protocol for Metadata Harvesting, 2015. Online: current as of June 21st, 2017.

[SHK06] Suh, Sangweon; Halpin, Harry; Klein, Ewan: Extracting common sense knowledge from wikipedia. In: Proceedings of the Workshop on Web Content Mining with Human Language Technologies at ISWC. volume 6, 2006.

[Va15] Vaidya, Gaurav; Kontokostas, Dimitris; Knuth, Magnus; Lehmann, Jens; Hellmann, Sebastian: DBpedia commons: structured multimedia metadata from the wikimedia commons. In: International Semantic Web Conference. Springer, pp. 281–289, 2015.