Collaborative Project

FROCKG - Fact Checking for Large Enterprise Knowledge Graphs

Project Number: E FROCKG Start Date of Project: 2020/01/01 Duration: 36 months

Deliverable 1.1: Feasibility Study

Dissemination Level	Public
Due Date of Deliverable	Month 4, 2020/04/30
Actual Submission Date	Month 4, 2020/04/30
Work Package	WP1, Requirements Elicitation
Task	T1.1
Type	Report
Approval Status	Work in progress
Version	1.0
Number of Pages	35

Abstract:
This report describes first the use cases from the project partners concerning the fact checking challenges. Further, it analyses the state-of-the-art technologies in science and industry, and states the feasibility of the FROCKG project.

The information in this document reflects only the author's views and Eurostars is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.

Project by Eurostars.

History

Version	Date	Reason	Revised by
1.0	30/04/2020	Draft revised	Wolfgang Schell

Author List

Organization	Name	Contact Information
metaphacts GmbH	Peter Haase	ph@metaphacts.com
metaphacts GmbH	Wolfgang Schell	ws@metaphacts.com
metaphacts GmbH	Jeen Broekstra	jb@metaphacts.com
Paderborn University	Axel Ngonga	axel.ngonga@upb.de
Paderborn University	Michael Röder	michael.roeder@upb.de
Sirma AI EAD	Nikola Rusinov	nikola.rusinov@ontotext.com
Sirma AI EAD	Todor Primov	todor.primov@ontotext.com

Sirma AI EAD	Zlatina Marinova	zlatina.marinova@ontotext.com
Zazuko GmbH	Adrian Gschwend	adrian.gschwend@zazuko.com

Contents

Introduction

Enterprise knowledge graphs underpin business-critical decisions. Facts extracted from news streams on business entities influence financial markets and business decisions. The results of our project will enable companies to ensure the veracity of facts obtained from their data suppliers before taking business-critical decisions based thereon.

The first goal of the FROCKG project is to carry out a feasibility study to affirm the practicability of the approach. This study focuses especially on the scientific aspects of the FROCKG project to gather insights of the research effort needed to realise the FROCKG vision. This vision consists in providing Fact Checking for Large Enterprise Knowledge Graphs.

First, the partners provide use case descriptions which are used to drive high-level requirements. These use cases will be later described in deliverable D1.3 in depth and provide a specification for the development and research within the FROCKG project. Second, we present an analysis of the state-of-the-art technologies with respect to existing fact checking approaches, from an industry as well as academic perspective. The advantages and disadvantages of the existing solutions are discussed. Finally, we conclude with some feasibility statements on the scientific aspects of the project.

Use Case Specifications

In this section, we present preliminary use case descriptions from the industrial partners. These use cases aim to show that there is a need in the industry for improving fact checking and its use in authoring or exploration of information. A more exhaustive description will be reported in D1.3 in M6.

Pharma Fake News Detection

Characteristics and Market

In the pharmaceutical industry, the product to market process is long, research and development costs are high, and often the overall success is put under high risk. Thus, the results from clinical trials are often identified as key inflection points in the process around which investors and business as overall receive early indications on the chances for success of a drug product. The release of clinical trial results has meaningful effects on market value for biopharmaceutical companies. Several studies observe asymmetric market reactions - stocks are affected much greater and the drop persists longer due to reported underperformance, compared to an increase in stock value and the duration of keeping high rates due to reported positive events.

Creation and spreading of both “positive” and “negative” fake news about performance of clinical drug products is a common practice to manipulate the stock market and to achieve easy gains due to rapid drop or raise in the stock shares prices. Such attempts have variable success - due to the professional attitude of content editors in specialized investment monitoring web sites and timely reactions of affected companies. However, there are also cases in which the effect of spreading “deceptive articles” was significant due to the scale (and budget) of the fake news campaign and involvement of the affected company. In a such case, US Securities and Exchange Commission (SEC) identified Galena Biopharma (but also the mother company – CytRx and also other similar entities Immunocellular Therapeutics and Lion Biotechnologies) as involved in various alleged stock promotion activities resulting in substantial raise of in company stocks from $55,555 in August 12, 2013 to over $210,000 in mid-January, 2014, which is close to 4 fold increase. The company paid just $65,000 to MissionIR (owned by DreamTeam Group) to publish 13 articles on specialized sites like Seeking Alpha and Forbes, and to promote them through various marketing channels (including social media).

The above case is an example of a well planned and coordinated effort with substantial effect on the stocks for a particular entity. Sometimes even a single tweet is enough to have a significant, although short, effect on the stock’s fluctuations. In another SEC case, a Scottish trader was accused that with just a single tweet (using fake accounts mimicking market research company) he affected with 28% and 16% the stocks of two companies, then he took advantage of the short prices to get shares and profit from the rise afterwards.

Elicitation approach

The elicitation approach of Sirma AI’s use case is based on presentations and customer/partners interviews.

The presentations were mainly focused on formalization of the available unstructured knowledge about clinical trials (public clinical trials reports from CT.gov and EMA) into a corresponding semantic representation in RDF. The presented approach also includes analysis of unstructured content elements (e.g. study outcomes, reported adverse events, etc) , its semantic normalization and usage of the data for providing deeper business insights (e.g. competitors product pipelines, drug repurposing, etc). As a potential new application of the semantic normalization approach was identified usage of the generated knowledge graph for validation of different statements, including such from investment portals concerning pharma companies’ product development updates and other financial analysis. This validity of the use case was confirmed both by large pharma companies, like

AstraZeneva, Novartis and UCB, but also by various partners, large system integration companies in the pharma domain – Cognizant and Infosys.

Common pain points and customer characteristics

Based on our understanding of the use case, our experience in the domain and interviews with customers and partners we can identify the following key stakeholder roles and their associated common pain points:

Pharma companies: Identify quickly attempts for affecting share prices and take corrective actions

Stock Monitoring/Regulatory Agencies: Scale fake news monitoring to increase coverage and limit the effects on the global market

Individual investors: Avoid making wrong investment decision based on misleading information

All three different pain points can be summarized with “validate each pharma investment advice with source clinical trial data”.

Data and Infrastructure

The source clinical trial data is publicly available and released regularly in the form of a clinical trial report from national registries (like clinicaltrials.gov and EudraCT). Most of the investor’s portals provide free access. There is a limited number of specialized investment sites which provide access based on a subscription model. Due to the fact that people who plan to affect the financial market spreading misleading news are going to seek for a wider audience, access to paid content will be not crucial.

Clinical trials data is provided in semi-structured format that will require some level of normalization – some repositories provide the data in a structured format, e.g. XMLs (clinicaltrials.gov), some other repositories in unstructured documents, like PDFs.

Outlook and Gain

Sirma AI’s use case aims to provide a platform for validation of suspicious claims shared in posts in investors portals with official information from clinical trials repositories.

This will require semantic normalization of the clinical trials data and building of a concise knowledge graph out of validated information from the regulatory agencies. The system must support natural language processing of unstructured content (investor’s portals posts) in English language as this is the primary language used. The NLP process must be able to formalize the study outcomes mentioned/referred in a proper form that will allow the comparison with the information available in the knowledge graph. A mechanism for matching of each claim identified in the post with the formalized clinical trials data in the graph should be developed.

The major gain from the use case will be the limitation of the effect of spreading fake/misleading statements about the performance of a clinical drug, that might affect the shares of a particular biopharma company and thus resulting in significant losses in the majority of shareholders, just to provide profit for a few limited traders.

Cultural Heritage and Archiving

Characteristics and Market

Organizations such as galleries, libraries, archives, and museums (GLAM sector) collect information on historical and current events, people, places, as well as artifacts and works of art for research, presentation, preservation, and creating narratives.

Also, in the last decade, public archives started to publish their catalogs on the web. For many archives, this means providing a web interface on top of their AIS (Archival Information System) and providing read-only access to the public records they maintain. In 2014 a group of Swiss archives in collaboration with Zazuko started the aLOD[1] effort to publish raw data as well, not just a web interface or API to the back-end.

As this information is collected from sources of different quality, provenance and trustworthy, it is important to verify any facts gathered and cross-reference them with other known facts or pieces of information.

Researchers and curators work with existing and new sources of information, formulating assertions and hypotheses by annotating data. Data acquisition and curation often involves discussion and elaboration within the community of experts.

Cultural Heritage data is typically open and publicly accessible. Exchanging or referencing data from other sources is very common and desired.

Institutions in the Cultural Heritage space often work with projects limited in scope, time and funding. So any bigger effort has to be broken down into smaller, self-contained increments. Funding is typically provided by funding agencies, state-owned organizations or foundations interesting in supporting research and/or cultural institutions. Collaboration between institutions, researchers and organizations is encouraged and welcome, the outcome of such projects in terms of data, documentation, or reports is typically also public.

Archivists collect data from sources, provide annotations and store them in an organized way. aLOD is using the RDF technology stack to publish and interlink archival records. In the

[1] http://alod.ch/

initial PoC, aLOD was focussing on linking the same concepts between multiple archives. Within the project, it became clear that using public knowledge graphs like Wikidata might facilitate this process a lot, as it is by now a well-accepted player in the open data movement. A lot of information is curated and published within the Wikidata knowledge graph. In the domain of records management, entities uniquely identifying people, places, and events are of high interest.

Since 2012, with members from thirteen countries, the International Council on Archives has been developing the new standard for the description of records based on archival principles. This standard is called Records in Contexts (RiC)[2] and is built on the RDF technology stack. It is expected that this will become the dominant model of the future thus enabling additional archives to use RDF and Linked Data. Archives within the aLOD project are now exploring RiC to describe their public catalogs.

One of the most recent features added to the aLOD platform is to enable users to annotate archival records. Annotations can be done on concepts available in external knowledge bases like Wikidata. Annotations must be factual. At the time writing, they need to be validated by other users or specialists, which can be a tedious process. Within the FROCKG project, we want to support administrators in this process and ideally automate a big part of these validations.

Elicitation approach

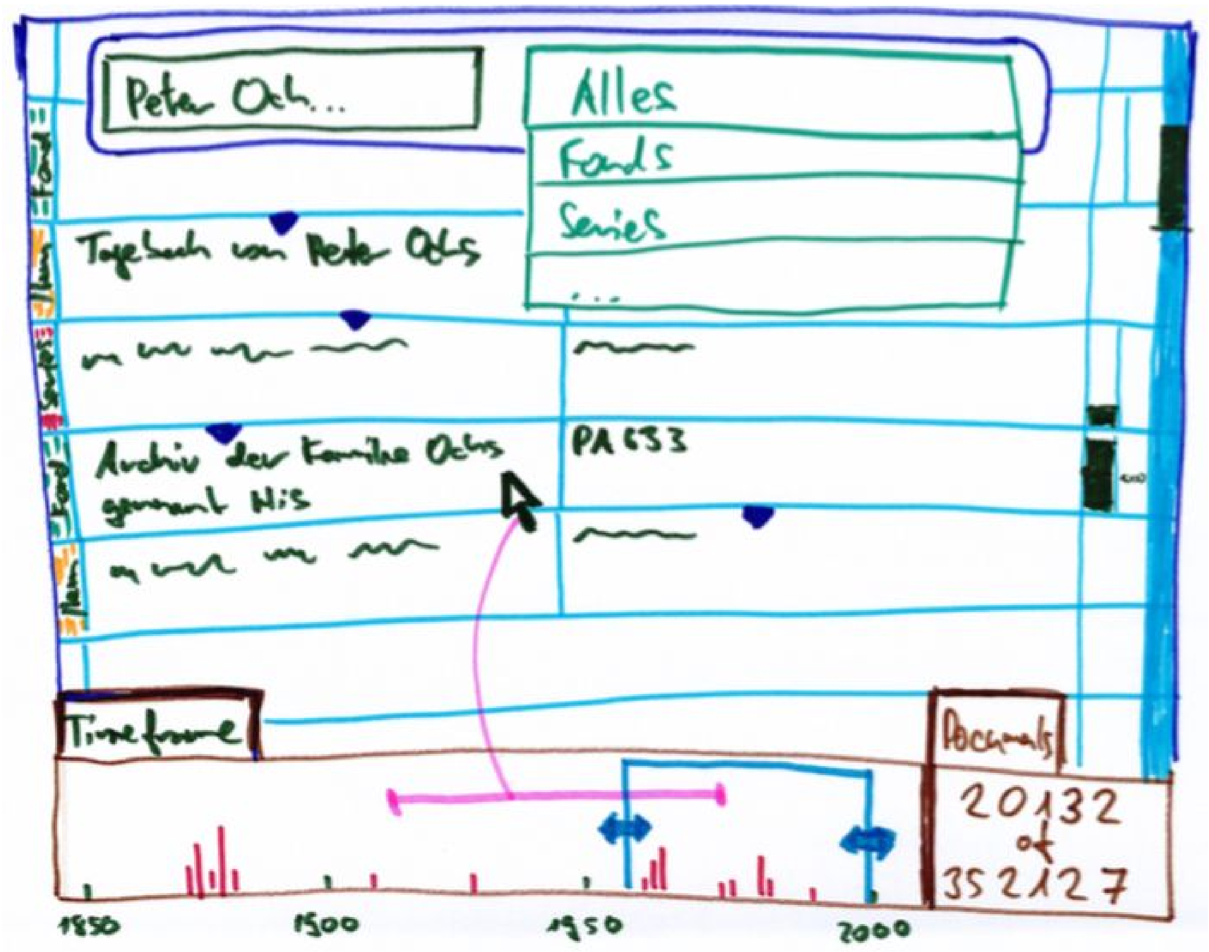
The elicitation process for the use case is based on previous collaboration with GLAM institutions on projects in the Cultural Heritage domain as well as on PoC developments within the aLOD platform.

Past and ongoing projects as well as related research projects already provide a lot of input and requirements which are based on the observed pain points as described below.

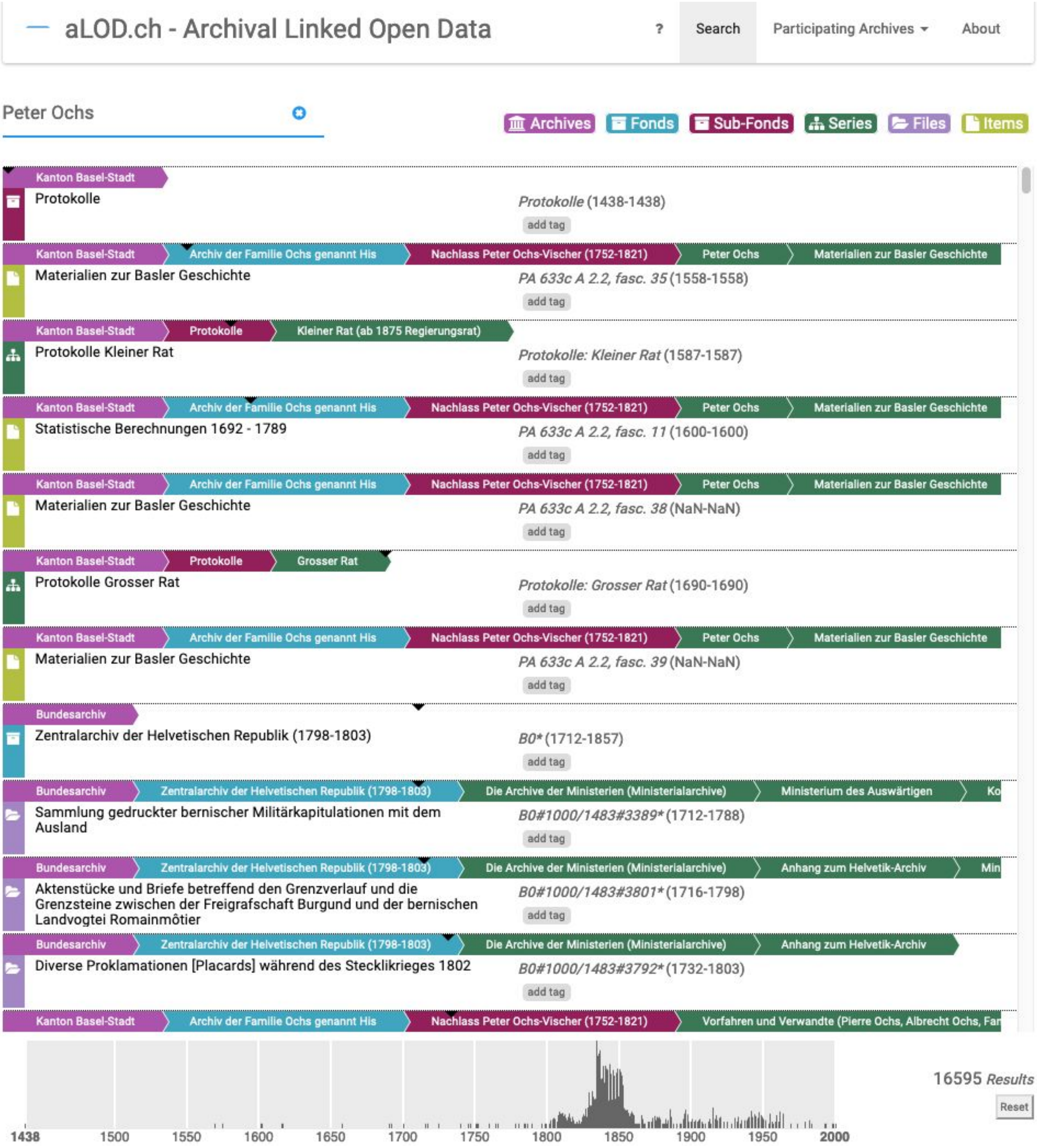
Mockups of potential user interfaces, and user experience provide a good base for discussion with partners and customers and help identify user interaction workflows and the requirements derived from them.

Example of a mockup:

[2] https://en.wikipedia.org/wiki/Records_in_Contexts



We will develop mockups to discuss the approach of annotation validation with the aLOD participants. The mockups are discussed within a small group of potential users in hands-on workshops. Once we have a working infrastructure, we will implement these UX ideas within the aLOD software stack and will validate the results of the FROCKG validation manually.



aLOD.ch - Archival Linked Open Data

Peter Och

Kanton Basel-Stadt
Protokolle
Protokolle (1438-1438)
add tag

Kanton Basel-Stadt | Archiv der Familie Ochs genannt His | Nachlass Peter Ochs-Vischer (1752-1821) | Peter Ochs | Materialien zur Basler Geschichte
Materialien zur Basler Geschichte
PA 633c A 2.2, fasc. 35 (1558-1558)
add tag

Kanton Basel-Stadt | Protokolle | Kleiner Rat (ab 1875 Regierungsrat)
Protokolle Kleiner Rat
Protokolle: Kleiner Rat (1587-1587)
add tag

Kanton Basel-Stadt | Archiv der Familie Ochs genannt His | Nachlass Peter Ochs-Vischer (1752-1821) | Peter Ochs | Materialien zur Basler Geschichte
Statistische Berechnungen 1692 - 1789
PA 633c A 2.2, fasc. 11 (1600-1600)
add tag

Kanton Basel-Stadt | Archiv der Familie Ochs genannt His | Nachlass Peter Ochs-Vischer (1752-1821) | Peter Ochs | Materialien zur Basler Geschichte
Materialien zur Basler Geschichte
PA 633c A 2.2, fasc. 38 (NaN-NaN)
add tag

Kanton Basel-Stadt | Protokolle | Grosser Rat
Protokolle Grosser Rat
Protokolle: Grosser Rat (1690-1690)
add tag

Kanton Basel-Stadt | Archiv der Familie Ochs genannt His | Nachlass Peter Ochs-Vischer (1752-1821) | Peter Ochs | Materialien zur Basler Geschichte
Materialien zur Basler Geschichte
PA 633c A 2.2, fasc. 39 (NaN-NaN)
add tag

Bundesarchiv
Zentralarchiv der Helvetischen Republik (1798-1803)
*B0** (1712-1857)
add tag

Bundesarchiv | Zentralarchiv der Helvetischen Republik (1798-1803) | Die Archive der Ministerien (Ministerialarchive) | Ministerium des Auswärtigen | Ko
Sammlung gedruckter bernischer Militärkapitulationen mit dem Ausland
*B0#1000/1483#3389** (1712-1788)
add tag

Bundesarchiv | Zentralarchiv der Helvetischen Republik (1798-1803) | Die Archive der Ministerien (Ministerialarchive) | Anhang zum Helvetik-Archiv | Min
Aktenstücke und Briefe betreffend den Grenzverlauf und die Grenzsteine zwischen der Freigrafschaft Burgund und der bernischen Landvogtei Romainmôtier
*B0#1000/1483#3801** (1716-1798)
add tag

Bundesarchiv | Zentralarchiv der Helvetischen Republik (1798-1803) | Die Archive der Ministerien (Ministerialarchive) | Anhang zum Helvetik-Archiv
Diverse Proklamationen [Placards] während des Stecklikrieges 1802
*B0#1000/1483#3792** (1732-1803)
add tag

Kanton Basel-Stadt | Archiv der Familie Ochs genannt His | Nachlass Peter Ochs-Vischer (1752-1821) | Vorfahren und Verwandte (Pierre Ochs, Albrecht Ochs, Far

16595 *Results*

Common pain points and customer characteristics

Enriching a knowledge graph by combining information from multiple sources facilitates the creation of a knowledge corpus that spans multiple topic areas and provides a comprehensive description of people, events, places, etc. These entities are of interest to researchers and curators, but also to publishers of data, information and narratives based on those, e.g. for consumption by the general public or interested individuals.

Validation and ensuring data quality are of high importance during the enrichment or integration of new data into knowledge graphs. Verifying facts or validating completeness, correctness and accuracy of data is often difficult due to limited automation, workflow and tool integration as well as user interfaces for these tasks. Similar issues impede authoring and annotation of information and metadata.

As entities in a given dataset typically have dataset-specific identifiers matching those entities to other datasets for referencing or inter-linking is an important task.

Cross-linking entities between data sets is at the same time highly desired and often difficult to automate when performing entity matching for huge amounts of data because of the need for disambiguation and ensuring data quality and veracity of connected information.

Cross-linking between entities is often done via use of Reference Data, i.e. data sets describing commonly used entities with labels, description, etc. Maintaining such a Reference Data Set requires additional functionality for lifecycle management, such as authoring, editing, curating, enriching entities in the data set.

Also, searching, cross-referencing and verifying multi-lingual data across data sets is a complex and manual task, as facts need to be compared and matched while translating between languages, data formats and vocabularies.

Provenance of information is an important aspect from which to derive trust and veracity of information. A well-established chain of provenance data and metadata is hence of high importance.

Data and Infrastructure

The GLAM sector has previously started a huge digitization effort in order to bring their collections into the digital age to allow easy access, sharing, searching, comparing, and cross-referencing of information.

Digitization comprises data and metadata about historical and current events, people, places, as well as artifacts and works of art. This (meta)data may be provided in various structured data formats and as well as unstructured data including photographs, scans, OCR'd extraction of documents, etc.

On top of this original data researchers and curators create annotations and narratives spanning own and external data sources.

Metadata is often already stored in Linked Data formats, i.e. RDF using established vocabularies. Also, a set of curated and widely used databases such as Wikidata[3], GeoNames[4], ULAN[5], etc. provide catalogs of entities for specific areas of interest. These data sets provide a common set of identifiers for entities which can be used to find and cross-reference data between data sets.

[3] https://www.wikidata.org/
[4] https://www.geonames.org/
[5] https://www.getty.edu/research/tools/vocabularies/ulan/

Within the aLOD project, data pipelines were established that are converting archival records from existing, non-RDF based Archival Information Systems (AIS).
For annotations, we use existing knowledge graphs like Wikidata which is already available in RDF as well. There are only a few annotations available right now, during the project we need to finalize the process for annotating archival records and crowd-source enough annotations for validation of FROCKG.

Galleries, Libraries, Archives, and Museums (GLAM institutions) typically store physical artifacts as well as corresponding digital representations in their own repositories. Data and metadata is more and more also available in publicly accessible formats and systems, so it can be accessed by external parties such as researchers, curators and publishers from other institutions and also the general public.
This allows integration of data sets from multiple sources and organizations into common data processing pipelines, e.g. for cross-referencing and enrichment with additional information.

Archive data is typically available in publicly accessible SPARQL endpoints. At the time of writing, there are between 5-10 million archival records available which amounts to around 70-100 million triples.

Outlook and Gain

Productizing the current manual curation process will help curators and researchers to create assertions and hypotheses and will improve both quality and velocity of such cross-referencing and fact checking.

This can be facilitated by high integration level of workflows and tool chains as well as automation, e.g. when extracting information, e.g. using Natural Language Processing (NLP) and Machine Learning (ML), and validating and verifying them based on rules (e.g. SHACL) and automated fact checking.
Additionally, rich editing capabilities provide high productivity and assist in authoring and annotating data as well as creating semantic narratives.

From other use-cases in the GLAM domain in Switzerland, we know that people are very motivated to support organizations in crowd-sourcing information related to the history of the places they live in. Some use-cases were so successful that taking care of this newly curated information started to draw a lot of resources that were not foreseen when the project started.

Fact-checking of these annotations could reduce this effort and save time and money for the archival institutions and at the same time add more value to the data itself.

Linked Open Data

Characteristics and Market

The amount of Linked Open Data available on the web grows continuously.[6] This data is readily available and used by companies to offer services. Since the datasets in the Linked Open Data Cloud can be published by any arbitrary body, it is crucial to be able to figure out whether a dataset that is available online contains true facts or spreads erroneous knowledge. Therefore, it is necessary to check Linked Open Datasets regarding their veracity before using them.

The aim of this use case is consequently to offer an easy-to-deploy Fact Checking platform and corresponding REST interfaces that can be used to check either single facts or complete datasets based on user-defined reference data (i.e., a reference corpus or a reference dataset). With this framework, we aim to demonstrate the usefulness of FROCKG technologies for consumers of Linked Open Data based on a subset of the Linked Open Data cloud by tackling two main challenges:

1. Checking the veracity of Linked Open Datasets (e.g., before using them as dataset in different algorithms)
2. Using Linked Open Datasets as reference knowledge base while
   a. Checking other knowledge bases
   b. Checking single facts (e.g., to support authoring tasks on the reference knowledge base)

Depending on the chosen datasets, their special features might have to be taken into account. For example, Wikidata is a dataset that is curated by a crowd-sourcing-based community. Its ambition is to be able to represent diverse views. It does not aim for global agreement on the ‘true’ data, since many facts are disputed or uncertain. Wikidata allows conflicting data to coexist and provides mechanisms to represent and manage this plurality. “Wikidata is not about truth. We look at what sources say and share it. It’s up to the user to decide what to believe. Wikidata is not about the truth, it’s about what sources say.”[7]
In addition to that, Wikidata gathers provenance data, i.e., facts published in primary sources, together with references to these sources. These references are directly part of the Wikidata data model. A further element of the data model is the notion of qualified statements. These qualifiers allow to express temporal validity (e.g. that a statement was true during a certain time period) or the level of certainty associated with a statement (e.g. whether a certain value was only guessed, estimated, or measured).

[6] https://lod-cloud.net/

[7] Denny Vrandecic, Markus Krötzsch: Wikidata: a free collaborative knowledgebase. Commun. ACM 57(10): 78-85 (2014)

Common pain points and customer characteristics

Not all available data has a high quality. To this end, services relying on this data may show a wrong behavior or low quality results. In addition to that, users may refuse to use all the available data since they are aware of some negative examples and can not distinguish between knowledge graphs that have a low veracity from those that have a high veracity.

Data and Infrastructure

The Linked Open Data cloud comprises structured knowledge graphs. For this use case, we will use a subset of these knowledge graphs as input for a veracity algorithm and/or as reference knowledge graphs. Depending on the chosen datasets, further, additional data sources (e.g., text corpora) might be of interest for this use case.

Outlook and Gain

The goal of the use case is twofold:

1. We would like to offer veracity values for the available knowledge graphs. This has two main effects. First, the users of the Linked Open Data can look up the veracity of single knowledge graphs. This leads to more trust of users into the Linked Open Data and, hence, enables the usage of this data. Such veracity scores might be even provided for single facts to enable users to use only a trustworthy part of a dataset. Second, the community is enabled to check datasets with a low veracity score and increase their veracity over time.
2. The authoring of datasets can be supported, e.g., by checking new statements while they are entered. The result of the checks could be used to assist the editor by pointing out possible errors or provide suggestions for values, as well as to provide references and evidence for the statements made by the author. In the context of collaborative authoring and discourse (e.g. to resolve disputes), a FROCKG fact checker could participate in the discourse by providing evidence, arguments in support or contradict a certain claim or fact.

State of the Art

In the following section, we give an overview of the the state of the art of knowledge graph techniques as relevant for the project. We start with an overview of knowledge graphs in general, then discuss in detail the state of the art for the validation of single facts as well as complete knowledge graphs, editing and curation, explanations, and knowledge extraction. We then describe state-of-the-art technology provided by the partners.

Knowledge Graphs

While the RDF data model and its first specifications have been around for more than two decades, the term Knowledge Graphs became popular in the last few years after Google started dubbing their knowledge base a knowledge graph.

What is a Knowledge Graph?

The knowledge graph (KG) represents a collection of interlinked descriptions of entities – real-world objects, events, situations or abstract concepts – where:

- Descriptions have a formal structure that allows both people and computers to process them in an efficient and unambiguous manner;
- Entity descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities, related to it.

Knowledge graphs combine characteristics of several data management paradigms and can be understood as a:

- Database, because the data can be queried via structured queries;
- Graph, because it can be analyzed as any other network data structure;
- Knowledge base, because the data in it bears formal semantics, which can be used to interpret the data and infer new facts.

When formal semantics are used to express and interpret the data of a knowledge graph, there are a number of representation and modeling instruments:

- Classes. Most often entity description contains a classification of the entity with respect to a class hierarchy. For instance, when dealing with general news or business information there could be classes Person, Organization and Location. Persons and organizations can have a common superclass Agent. Location usually has numerous sub-classes, e.g. Country, Populated place, City, etc. The notion of class is borrowed by the object-oriented design, where each entity should belong to exactly one class.

- Relationship types. The relationships between entities are usually tagged with types, which provide information about the nature of the relationship, e.g. friend, relative, competitor, etc. Relation types can also have formal definitions, e.g. domain and range definitions, or that parent-of is inverse relation of child-of, they both are special cases of relative-of, which is a symmetric relationship. Or defining that sub-region and subsidiary are transitive relationships.
- Categories. An entity can be associated with categories, which describe some aspect of its semantics, e.g. “Big four consultants” or “XIX century composers”. A book can belong simultaneously to all these categories: “Books about Africa”, “Bestseller”, “Books by Italian authors”, “Books for kids”, etc. Often the categories are described and ordered into taxonomy,
- Free text descriptions. It is possible to add ‘human-friendly text’ to further clarify design intentions for the entity and improve search.
- Ontologies. They serve as a formal definition between the developers of the knowledge graph and its users. A user could be another human being or a software application that wants to use the data in a reliable and precise way. It ensures a shared understanding of the data and its meanings.

What is not a Knowledge Graph

Not every RDF graph is a knowledge graph. For instance, a set of statistical data, e.g. the GDP data for countries, represented in RDF is not a KG. A graph representation of data is often useful, but it might be unnecessary to capture the semantic knowledge of the data. It might be sufficient for an application to just have a string ‘Italy’ associated with the string ‘GDP’ and a number ‘1.95 trillion’ without needing to define what countries are or what the ‘Gross Domestic Product’ of a country is. It’s the connections and the graph that make the KG, not the language used to represent the data.

Not every knowledge base is a knowledge graph. A key feature of a KG is that entity descriptions should be interlinked to one another. The definition of one entity includes another entity. This linking is how the graph forms. (e.g. A is B. B is C. C has D. A has D). Knowledge bases without formal structure and semantics, e.g. Q&A “knowledge base” about a software product, also do not represent a KG. It is possible to have an expert system that has a collection of data organized in a format that is not a graph but uses automated deductive processes such as a set of ‘if-then’ rules to facilitate analysis.

History of Knowledge Graphs

In 2010, Google acquired the company Metaweb and with it Freebase, described as an "open, shared database of the world's knowledge". What became the Google Knowledge Graph was initially built on top of Freebase, which in itself was shut down in 2016 and succeeded by Wikidata.

Wikidata was launched by Wikimedia Deutschland in 2012 to address the problem of diverting infobox data in multiple languages on Wikipedia pages. Wikidata is using its own data model internally but exposes the data as RDF and SPARQL since 2015. When Google announced that they will shut down Freebase, its knowledge base was migrated to Wikidata.

Wikidata can be considered the most popular knowledge graph available, many organizations contribute their data to Wikidata and make sure concepts have a Wikidata QID. Authority files like VIAF[8] are referenced in Wikidata entries, which makes Wikidata QIDs a very powerful identifier to explore additional data about an entity. Many of those referenced authority files are also available as RDF.

Before Wikidata, the main knowledge base for RDF based data was DBpedia. It aims to extract structured content from the information created in Wikipedia. It was first presented in 2007 and is available to this day. While part of its purpose was replaced by Wikidata, it still provides valuable information as machine-readable data. Unlike Wikidata it also provides extracted text fragments from Wikipedia like abstracts on a Wikipedia page.[9]

YAGO[10] is a Knowledge Graph that is built on existing datasets. In its latest version 4, it is largely built on Wikidata and WordNet but it is using a simpler taxonomy of schema.org. Classes are equipped with SHACL constraints. It is curated in a way that OWL-based reasoning is feasible.

The UniProt Knowledgebase (UniProtKB)[11] is the central hub for the collection of functional information on proteins, with accurate, consistent and rich annotation. UniProt is one of the largest public RDF knowledge graphs available, at the time writing there are roughly 60 billion RDF triples available in the public SPARQL endpoint. UniProt is a defacto standard in its domain and an excellent example of a more domain-specific knowledge graph.

Knowledge graphs came a long way. A decade ago it was mostly about research projects with little real-world use, nowadays companies want to build their corporate knowledge graphs behind firewalls to better harvest information from their data silos.

RDF is using global identifiers to link information among silos. This gives an interesting emergent property: self-assembling data structures. With global Ids and a graph database, the system takes care of the joins. Humans aren't writing code or queries to assemble the data. Building knowledge graphs was never easier.

[8] http://viaf.org/

[9] Ismayilov, A., Kontokostas, D., Auer, S., Lehmann, J., & Hellmann, S. (2018). Wikidata through the Eyes of DBpedia. *Semantic Web*, *9*(4), 493-503. See http://www.semantic-web-journal.net/system/files/swj1462.pdf

[10] Thomas Pellissier-Tanon, Gerhard Weikum, and Fabian Suchanek “YAGO 4: A Reason-able Knowledge Base”

[11] https://www.uniprot.org/help/uniprotkb

Fact Checking

Validation of single facts

Fact checking can be carried out manually or automatically. Manual fact checking is usually implemented using crowdsourcing and is the current state of the art for checking enterprise knowledge graphs.[12] End-user oriented knowledge graph data acquisition tools targeting specific use cases sometimes include user assistance techniques like autosuggestion and data validation.[13] However, the focus is again on formal correctness of data with respect to the expected structure and domain restrictions. To our knowledge, however, there are no tools that integrate fact checking using external sources into the manual knowledge acquisition process.

Automatic algorithms for fact checking can be broadly classified into two categories: (i) approaches that use unstructured textual sources and (ii) approaches that use structured information sources. Approaches of the first category use predefined templates to transform RDF facts into natural language.[14] The natural language representation is used to search for evidence in a reference text corpus. However, none of these approaches are able to determine whether a fact is likely to be false. In contrast to the solution FROCKG is aiming at, these approaches fail to explain why they assume a fact to be true, a feature of utmost importance for any practical use in enterprises.

The second category of algorithms relies on knowledge graphs as a source of evidence. Several approaches view a given knowledge graph as a labeled graph connecting nodes (entities) and edges (relations). Given an input triple (*s, p, o*), the goal is then to search for paths of length up to a given threshold *k* and use them to (in-)validate the given input triple. For instance, some approaches view a knowledge graph as an undirected network of paths. [15] The task is then to find the shortest paths that connect *s* and *o* and are semantically related to *p*. These approaches are unsupervised and do not require prior training data. However, they do not take into consideration the terminological information (in particular the

[12] Dong, Xin, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. "Knowledge vault: A web-scale approach to probabilistic knowledge fusion." In SIGKDD, pp. 601-610. ACM, 2014.

[13] Rafael S. Gonçalves, Martin J. O'Connor, Marcos Martínez-Romero, Attila L. Egyedi, Debra Willrett, John Graybeal, and Mark A. Musen "The CEDAR Workbench: An Ontology-Assisted Environment for Authoring Metadata that Describe Scientific Experiments". In ISWC 2017

[14] Gerber, Daniel, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. "Defacto—temporal and multilingual deep fact validation." Journal of Web Semantics (2015): 85-101.
Zafar Habeeb Syed, Michael Röder, and Axel-Cyrille Ngonga Ngomo: "FactCheck: Validating RDF Triples using Textual Evidence". In Proceedings of the International Conference on Information and Knowledge Management (CIKM), 2018.

[15] Shiralkar, P., Flammini, A., Menczer, F., Ciampaglia, G.L.: Finding streams in knowledge graphs to support fact checking. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 859–864. IEEE (2017)
Ciampaglia, G.L., Shiralkar, P., Rocha, L.M., Bollen, J., Menczer, F., Flammini, A.: Computational fact checking from knowledge networks. PloS one 10(6), e0128193 (2015)

semantics of RDFS) of the input knowledge graph while defining semantic proximity metrics. Other approaches view KBs as graphs and search for meta paths to extract features.[16] These features are then used to train a classification model to label unseen facts as true or false. However, these approaches require training data in the form of labeled meta paths and hence require significantly more human effort than the approach presented herein. In PredPath, the authors propose a novel method to automatically extract meta paths—called anchored predicate paths—given a set of labeled examples.[17] To achieve this goal, PredPath uses the rdf:type information contained in the input knowledge graph. However, the anchored predicate paths used for learning features are selected based on the type information of subject and object irrespective of the predicate connecting them. This means that they do not consider the domain, range and class subsumption provided by the RDFS schema of the given knowledge graph. Consequently, their ability to generalize over paths is limited. Additionally, PredPath requires labeled training data. Hence, porting it to previously unseen predicates is significantly more demanding than porting our approach, which is fully unsupervised. COPAAL overcomes the previously mentioned limitations of other knowledge-graph-based approaches by finding paths which corroborate a given triple fact. COPAAL makes use of the RDFS semantics of the Knowledge Graph and does not need training examples and does not require any supplementary effort to deploy to previously unseen relations.[18]

In addition to the previously mentioned fact checking approaches, other, closely related fields of research exist. Graph-based fact finders rely on a bipartite graph of facts and sources to determine how likely statements are to be true depending on how reliable the sources in which they were found are. A major drawback of these approaches is that they don't implement the search for evidence for a given fact but assume it as a given input. Hence, they can mainly be seen as a way to improve existing fact checking systems by using trustworthiness measures. In addition, we ran a preliminary study of these algorithms and showed that they cannot be applied to knowledge graphs because they are poorer in their performance than approaches designed for RDF graphs.

Another closely related field is link prediction. Several approaches encode the entities and relations in a KB using vector embeddings to predict missing triples in a knowledge graph.[19]

[16] Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment 4(11), 992–1003 (2011)
Zhao, M., Chow, T.W., Zhang, Z., Li, B.: Automatic image annotation via compact graph based semi-supervised learning. Knowledge-Based Systems 76, 148–165 (2015)
Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Machine learning 81(1), 53–67 (2010)

[17] Shi, B., Weninger, T.: Discriminative predicate path mining for fact checking in knowledge graphs. Knowledge-based systems 104, 123–133 (2016)

[18] Syed, Zafar Habeeb, Röder, Michael and Ngomo, Axel-Cyrille Ngonga. "Unsupervised Discovery of Corroborative Paths for Fact Validation." In The Semantic Web – ISWC 2019, 2019.

[19] Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI conference on artificial intelligence (2014)
Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. pp. 926–934 (2013)
Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for

Other approaches mine Horn rules that can be used for link prediction. However, link prediction addresses a related but different problem. Given a knowledge graph, these approaches compute a score function, which expresses how likely it is that any triple whose subject, predicate and object belong to the input graph should exist within the graph.[20] Fact validation approaches address a different but related goal: Given a graph *G* and a triple *t*, they aim to compute the likelihood that *t* is true. A core repercussion of these two different problem formulations are the runtimes and the applications of link prediction and fact checking. While fact validation algorithms are used in online scenarios, embedding-based algorithms are often used offline.

Validation of Knowledge Graphs

To the best of our knowledge, there exist no approaches for validating a complete knowledge graph. A straightforward solution would be to use the previously mentioned fact checking algorithms to check every fact within a knowledge graph. However, systems like DeFacto or FactCheck require up to 5 seconds for checking a single fact. For COPAAL, an average throughput of 21.02 triples per min has been reported. Since enterprise knowledge graphs can contain billions of facts, such an approach is not applicable.

Explanations

Explainability is a new field of research that became more important since the uptake of machine learning algorithms that make use of deep artificial neural networks or similar, complex approaches. Such machine learning approaches have the disadvantage that they act as black boxes, i.e., the user can not see how the algorithm comes to a conclusion. This is a large problem for the usage of these approaches in practice since 1) even domain experts can not see whether a classification model classified something in a wrong way and 2) the general acceptance of black box solutions that can not be understood by humans is low. To this end, several attempts have been made to explain the classification results of machine learning algorithms.[21] However, these approaches either need to be configured or their results need additional interpretation (e.g., heatmaps). In both cases, expert knowledge is needed. Since checking single facts can be seen as a classification task, the usage of these approaches for explainability could be used.

A different strategy is the usage of models that can be directly interpreted and explained. For example, the fact checking algorithms relying on reference knowledge graphs presented in

knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)

[20] Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing yago: scalable machine learning for linked data. In: Proceedings of the 21st international conference on World Wide Web. pp. 271–280. ACM (2012)

[21] Molnar, Christoph. "Interpretable machine learning." A Guide for Making Black Box Models Explainable (2018).
Friedman, Jerome H. “Greedy function approximation: A gradient boosting machine.” Annals of statistics (2001): 1189-1232. https://christophm.github.io/interpretable-ml-book/pdp.html#fnref27

Section 3.1.1 determine a set of paths supporting or refuting the given fact. These paths can be used for generating explanations by transforming the paths comprising RDF triples into natural language. This field of research has been previously exploited by a great number of publications, which are mainly based on template- or rule-based approaches.[22] The WebNLG challenge[23] provided a benchmark corpus of English texts verbalizing RDF triples in 15 different semantic domains. Among the participating models, the ones based on sequence-to-sequence artificial neural networks have shown promising results.[24] While these approaches rely on a sequence of triples, more recent approaches rely on a graph representation. Marcheggiani and Perez propose a structured data encoder based on a graph convolutional neural network, which directly exploits the graph structure.[25] Distiawan et al. present an LSTM-based approach that captures the global information of a reference knowledge graph by encoding the relationships both within a triple and between the triples.[26] Ferreira et al. introduced a systematic comparison of neural pipelines and end-to-end data-to-text approaches for the generation of text from RDF triples.[27] Although Marcheggiani et al. show that the linearisation of the input graph has several drawbacks, the authors implement an architecture based on a recurrent neural network, which shows superior results when compared to former architectures. Recently, Ribeiro et al. devise a unified graph attention network structure which investigates graph-to-text architectures that combine global and local graph representations to improve the fluency of generated texts.[28]

[22] Philipp Cimiano, Janna Lüker, David Nagel and Christina Unger. "Exploiting Ontology Lexica for Generating Natural Language Texts from RDF Data". In Proceedings of the 14th European Workshop on Natural Language Generation, 2013, pp. 10–19.
Daniel Duma and Ewan Klein. "Generating Natural Language from Linked Data: Unsupervised template extraction". In Proceedings of the ISWC 2013, pp. 89–94.
Basil Ell and Andreas Harth. "A language-independent method for the extraction of RDF verbalization templates". In Proceedings of the INLG, 2014, pp. 26–34.
Or Biran and Kathleen McKeown. "Discourse Planning with an N-gram Model of Relations". In Proceedings of the EMNLP, 2015, pp. 1973–1977.

[23] Emilie Colin, Claire Gardent, Yassine Mrabet, Shashi Narayan and Laura Perez-Beltrachini. "The webnlg challenge: Generating text from dbpedia data". In Proceedings of the 9th INLG conference, 2016. pp. 163–167.

[24] Amin Sleimi and Claire Gardent. "Generating Paraphrases from DBPedia using Deep Learning". In Proceedings of the WebNLG, 2016, pp. 54.
Yassine Mrabet, Pavlos Vougiouklis, Halil Kilicoglu, Claire Gardent, Dina Demner-Fushman, Jonathon Hare and Elena Simperl. "Aligning texts and knowledge bases with semantic sentence simplification". In Proceedings of the WebNLG, 2016.

[25] Diego Marcheggiani and Laura Perez. "Deep Graph Convolutional Encoders for Structured Data to Text Generation". In Proceedings of the 11th International Conference on Natural Language Generation, 2018.

[26] Bayu Distiawan, Jianzhong Qi, Rui Zhang and Wei Wang. "GTR-LSTM: A triple encoder for sentence generation from RDF data". In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1627–1637.

[27] Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg and Emiel Krahmer. "Neural data-to-text generation: A comparison between pipeline and end-to-end architectures".In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 552–562.

[28] Leonardo FR Ribeiro, Yue Zhang, Claire Gardent and Iryna Gurevych. "Modeling Global and Local Node Contexts for Text Generation from Knowledge Graphs". arXiv preprint arXiv:2001.11003, 2020.

However, to the best of our knowledge, there is no existing approach to generate explanations for the results of fact checking systems. A first attempt that goes into this direction has been done by Syed et al. by generating natural language representations of corroborative paths found within a reference graph while checking a given fact.[29]

Editing and Curation

Data Authoring

Data in a knowledge graph may be generated from various sources:

- automatic data ingestion via data integration,
- federation of external knowledge graphes or
- manual data authoring or enrichment.

Manual data authoring or editing is typically not done on RDF statement level, but rather via tooling to assist the editor when entering data, e.g., using a form-based approach. The available form fields typically depend on the type of the edited entity and may be derived from modelling information such as an ontology, SHACL shapes or other types of meta information.

Ideally, the meta information also provides information on constraints which allows validation before writing data to the knowledge graph which helps in ensuring consistency. Additionally, tooling may assist in finding related entities using lookup mechanisms to identify canonical identifiers or referenced entities, possibly from external reference data sets.

Form-based editing for generic data represented in a graph, is typically driven by metadata such as RDF Schema (RDFS)[30] and ontologies defined using the Web Ontology Language (OWL)[31]. For example, the metaphactory platform drives its editing form components by means of Field Definitions which can be partially generated from existing OWL property restrictions[32] or simple SHACL shapes[33]. The Eccenca form-based editing component follows a similar approach[34].

[29] Syed, Zafar Habeeb, Röder, Michael and Ngomo, Axel-Cyrille Ngonga. "COPAAL – An Interface for Explaining Facts using Corroborative Paths." In Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas), 2019.
[30] https://www.w3.org/TR/rdf-schema/
[31] https://www.w3.org/OWL/
[32] https://help.metaphacts.com/resource/Help:SemanticForm#fielddefinitions
[33] http://datashapes.org/forms.html
[34] https://documentation.eccenca.com/latest/explore/building-a-customized-user-interface

Editing Workflows

Editing workflows define processes and tooling to assist in editing with multiple involved people, organizations or automation processes. They may define roles and responsibilities between involved parties, e.g., approval processes and quality checks.
Workflows may be specified for content authoring as well as for defining configuration and meta artefacts.
Typically, such workflows consist of authoring and editing steps assigned to one or more persons with later stages of the workflow dedicated to editorial and approval stages. Besides human agents editing, approval, or action steps can also be performed by automated processes, e.g. by simple scripts, sophisticated rule engines or Artificial Intelligence (AI)-assisted services.
Content authoring consists of providing descriptive texts, narrations, definitions and other textual content or references. This is typically implemented with rich editing capabilities using advanced text editors.
Additional types of content editing or authoring consist of collecting entity data and providing annotations, typically using a form-based approach. This allows contributions through auto-suggestion based on structured content, e.g. from databases, knowledge graphs or other internal and external services. Forms for editing structured content are typically auto-generated based on schema information as described in the previous section on *Data Authoring*.

Validation against predefined constraints

Orthogonally to validation approaches that rely on establishing a comparative measure of confidence against text corpora or large structured data sources, the Shapes Constraint Language (SHACL)[35] is a formalism for checking integrity, consistency, and completeness of data against a predefined set of constraints, so-called *data shapes*. SHACL allows validation of knowledge graphs against these shapes and can be used to ascertain if data is complete and consistent, and to report on areas of interest in a knowledge graph.
In addition to the official specification maintained by a working group in the W3C some extensions are available such as DASH[36] (Data Shapes Vocabulary), a collection of reusable extensions to SHACL for a wide range of use cases.

Although several existing SHACL implementations exist (including Eclipse RDF4J[37], RDFUnit[38], rdf-validate-shacl[39], and Ontotext GraphDB[40]), there is little or no work done on applicability of these approaches in large-scale fact checking scenarios such as described in

[35] Shapes Constraint Language (SHACL). W3C Recommendation 20 July 2017. https://www.w3.org/TR/shacl/
[36] DASH: Data Shapes Vocabulary, http://datashapes.org/dash
[37] RDF4J: Validation with SHACL. Technical report. https://rdf4j.org/documentation/programming/shacl/
[38] RDFUnit: SHACL. Technical report. https://github.com/AKSW/RDFUnit/wiki/SHACL
[39] rdf-validate-shacl: JavaScript SHACL validation, https://github.com/zazuko/rdf-validate-shacl
[40] GraphDB: SHACL Validation. Technical report. http://graphdb.ontotext.com/documentation/free/shacl-validation.html

the FROCKG use cases. We will provide a consolidated approach where existing, improved and possibly new implementations of SHACL-based algorithms are integrated in the overall platform, and SHACL as a formalism is employed to maximum benefit for data validation at scale.

A secondary benefit of constraints in the form of SHACL shapes is that such constraints can be used to inform the user interface on how to present information, and which options to offer when information is edited. As a simple example, if a shape defines that a person can only have one name, but several addresses, the user interface can leverage this to make sure that an editing form for persons offers multiple fields for addresses but blocks entry of more than one name. Several existing approaches leverage SHACL shapes in this manner. The Schimatos project[41] semi-automatically generates web forms for knowledge graph editing from SHACL shapes. The metaphactory platform itself has support for leveraging SHACL shape data to drive UI components, both as textual forms and as part of its graphical editing components. However, these approaches are currently limited to simple use cases where form attributes correspond 1:1 to the underlying data and shapes. In more complex cases, where the UI is expected to present an abstracted view over the data, the current approaches often fall short. In addition, formulating SHACL shapes with the purpose of driving (multiple) views on the underlying data is itself complex and time-consuming. A focus area of research for FROCKG will therefore be to develop best practices and tools where standardized data constraints such as SHACL shapes can be successfully leveraged to quickly create user interface components for the viewing and editing of data from the knowledge graph.

Integrating validation and knowledge sources in a curation environment

A core objective of the FROCKG project is to not only provide components and algorithms for fact checking and validation, but to offer an integration approach that enables knowledge workers to effectively make use of these components and algorithms as part of their normal workflow when editing or curating content. To enable this, an integration platform will be needed that offers facilities to access these validation components and effectively combine, by means of federating and adapting, the disparate knowledge sources into a coherent experience.

Validation results such as produced by the algorithms described in the preceding sections need to be presented to knowledge workers in a context that enables them to be informed by those results, and in addition *act upon* those results. Depending on the specifics of the use case, there are several possible actions the FROCKG project aims to enable as part of its integration work.

As validation results of the form produced by any automated tooling are typically a measure of confidence based on corroboration (found by analysis of textual corpora or other, more structured data sources), the user needs to be informed of the confidence measure result itself for a given fact or set of facts, and the means by which the system arrived at it

[41] https://github.com/schimatos/schimatos.org

(so-called *explainability*). Enabling this will require user interface components that present the validation results as *annotations* on the underlying data, in a setting where context in terms of the large knowledge base as well as the actual result and its corroboration is clear. Typically, the user will need to be able to explore the facts themselves and their source and "neighboring" (that is, closely related) facts to obtain confidence that the validation result is (in)correct.

The user then needs to edit at two separate levels:

- At the meta level, they need to be able to accept or dismiss the result of the validation, thereby indicating that they either agree or disagree with its findings;
- At the data level, they need to be able to accept, correct, or otherwise annotate and curate the facts under scrutiny.

In a collaborative curation environment it will typically also be a requirement that users can leave their own annotations and notes on their decisions. Furthermore, in many use cases some form of auditability and preservation of edit history, and even options to reverse certain decisions, must be considered.

Various existing systems cater for some subset of these requirements, however to the best of our knowledge no single, open system provides a comprehensive user experience for this kind of knowledge graph curation at scale. The challenge will be to leverage existing platforms for integrated structured search and flexible UI component development, such as the metaphactory platform, to enable this experience.

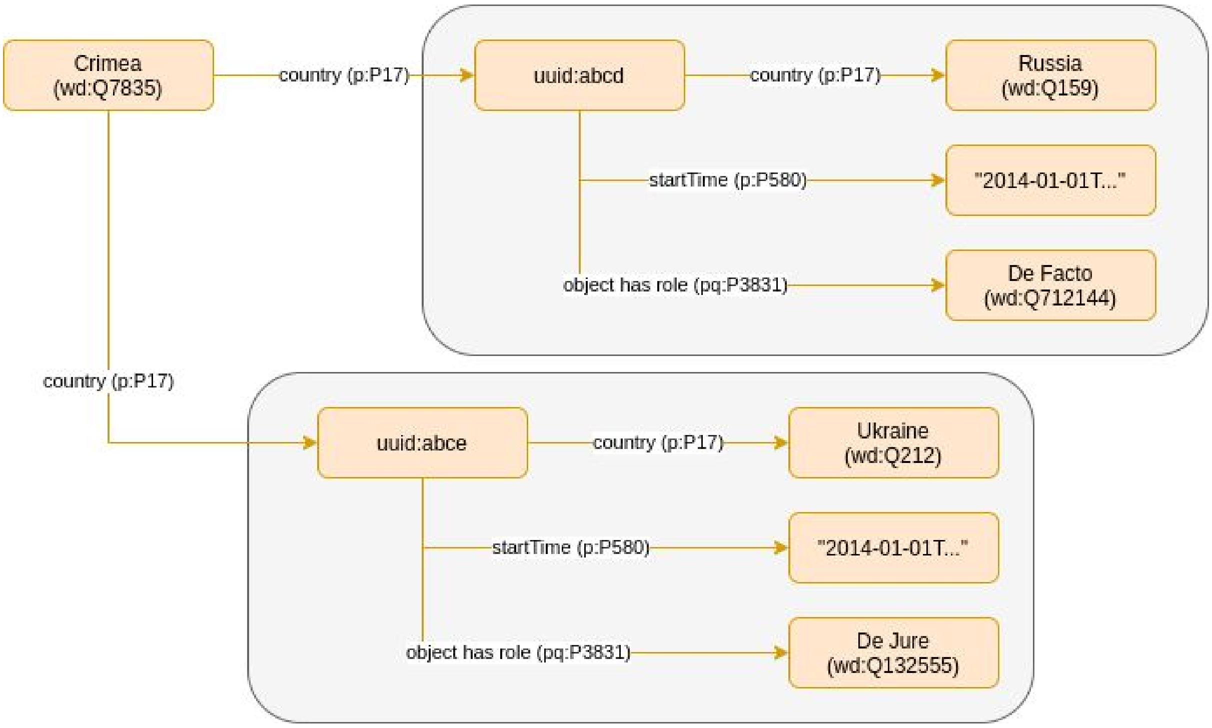
Annotating data

In order to be able to represent metadata such as curation results, auditing timelines, as well as fact checking assertions, a model for *annotation* of individual facts as well as groups of facts will be needed. This model will need to provide a consistent approach toward representing different kinds of annotations in a way that makes working with those annotations in an editing and curation user interface easy. Here, we discuss several current existing approaches for annotation.

Wikidata Qualifiers

The Wikidata project[42] is a massive community-maintained knowledge graph for machine-processable data, using RDF as its data model. As part of this effort, it was recognized early on that a solution to annotate certain data with *qualifiers* is necessary, to express provenance / source data. To express such metadata about facts, Wikidata uses an object-as-value approach, where, for any given statement, the value of that statement is turned into an object in its own right. This object can then be annotated with additional properties to indicate metadata about the fact.

[42] Vrandečić, Denny, and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase." *Communications of the ACM* 57.10 (2014): 78-85. See also https://www.wikidata.org/



As an example, to express the fact that the country to which the Crimea region belongs is in dispute, and that since 2014 it is said to belong to Ukraine “de jure” but to Russia “de facto”, Wikidata records two ‘country’ relations for the Crimea region. The values of those relations are objects themselves, which then get annotation attributes to establish the actual value (Russia vs Ukraine) as well as the start time and the role.

Web Annotation Framework

The Web Annotation Data Model[43] is a W3C Recommendation for a structured model and format for sharing and reuse of annotations. It is explicitly designed to be generic and focus on a reusable structure that can be applied for any particular annotation use case (whether that is provenance tracking, fact checking validation assertions, or any other kind of metadata).

At its core, the framework distinguishes the concept of an *annotation* which relates a *body* to a *target.* The target in this context is the entity being annotated, and the body is what is being said by the annotation about the target.

A simple example of annotation that links a comment to a particular page, is as follows:

```
<http://example.org/anno1> a oa:Annotation ;
    oa:hasBody <http://example.org/post1> ;
    oa:hasTarget <http://example.com/page1> ;
```

[43] Sanderson, Robert, Paolo Ciccarese, and Herbert Van de Sompel. "Designing the W3C open annotation data model." *Proceedings of the 5th Annual ACM Web Science Conference*. 2013. See also https://www.w3.org/TR/annotation-model/

```
    oa:motivatedBy oa:commenting ;
    dcterms:creator <http://example.org/person1> ;
    dcterms:created "2015-11-18T12:00:00Z" .
```

In this example the annotation indicates that post1 (the body) and page1 (the target) are related, and the supplied motivation indicates that post1 is a comment about page1. Furthermore, person1 is the creator of the annotation, and we have a timestamp to indicate when the annotation was first created.

On top of this core model, the framework introduces a vocabulary to express types of and relations between the various entities. The possible types of a resource are Dataset, Image, Video, Sound and Text—in other words a non-domain-specific, but representation-specific categorization of resources. In terms of other properties and relations, it introduces a basic vocabulary to express lifecycle information (e.g. created/modified timestamps), as well as ways to express intended audience, motivation and purpose, and rights/intellectual property information.

PROV

PROV[44] is a formal data model and vocabulary for tracking *provenance* of data. Its core model centres around the notions of Entities, Activities, and Agents. *Entities* are any kind of physical, digital or conceptual thing, in other words anything that potentially needs to be annotated with provenance information. *Activities* describe the actions and processes that create and manipulate entities. Finally, *Agents* describe the actors that are associated with a particular activity on a particular entity. In addition to these core concepts, PROV formalizes a model and vocabulary for tracking the relationships between them, including notions of derivation of entities (for example, to track various revisions of a particular document), time tracking of activities, and the roles that entities or agents play in any activity.

CRMInf

CRMInf[45] is an ontology for capturing argumentation and inference making in descriptive and empirical sciences. It is an extension of the well-known CIDOC-CRM[46] reference model widely used in the cultural heritage domain.

In particular, CRMInf introduces nomenclature to capture concepts such as Argumentation, Observations, Beliefs, Prediction, etc, and relationships to capture the argumentative structures between such concepts (e.g. a particular inference can be *motivated by* a particular belief). It is a vocabulary that is well-suited towards capturing scientific discourse in particular.

[44] Gil, Yolanda, et al. "PROV model primer." *W3C Working Group Note* (2013). See https://www.w3.org/TR/prov-overview/

[45] Stead, S., and M. Doerr. "CRMinf: The argumentation model. An extension of CIDOC-CRM to support argumentation." (2015). See also http://www.cidoc-crm.org/crminf/home-4 .

[46] Cidoc, Crm. "The CIDOC Conceptual Reference Model." *2003-10). http://cidoc.ics.forth.gr* (2003).

RDF*

RDF*[47] is an extension of the core RDF data model born from a recognition that annotation of individual facts requires a form of *reification:* in order to be able to say something *about* a fact, we have to be able to treat that fact itself as an object in its own right. RDF* introduces an extended syntax for RDF serialization formats such as Turtle. For example:

```
<<:p1 :name "Bob">> :confidence 0.9.
```

expresses the (meta)fact that we have 90% confidence that the name of resource `p1` is "Bob". The general form is still an RDF triple with a subject, predicate, and object, the extension RDF* introduces is that the subject value (enclosed in `<< >>`) is itself a triple.

RDF* is a community effort, currently enjoying significant industry uptake (for example in Eclipse RDF4J[48], Ontotext GraphDB[49], Stardog[50], Blazegraph[51], and AnzoGraph[52]). RDF* makes handling annotations on individual facts (a.k.a. "edge properties") at scale more efficient when compared to more traditional approaches towards modeling such annotations in RDF (using "classic" RDF reification, named graphs, or value objects). For example, in classic RDF reification, the annotation on the name of p1 would look like this:

```
[] a rdf:Statement;
   rdf:subject :p1;
   rdf:predicate :name;
   rdf:object "Bob" ;
   :confidence 0.9.
```

While this structure expresses the same information, it is significantly more verbose (introducing 4 extra triples to model the annotation of each individual fact), and therefore difficult to work with at scale. Other approaches offer less verbosity, but suffer from other drawbacks; for example the notion of *value objects* introduces a changed model for the relationship value. In this model, the annotation could look like this:

```
:p1 :name [ :value "Bob" ;
            :confidence 0.9 ].
```

A major drawback to such an approach is that incremental changes to the knowledge graph are hard. If the name relation has been originally designed to have a literal as object,

[47] Hartig, Olaf, and Bryan Thompson. "Foundations of an alternative approach to reification in RDF." *arXiv preprint arXiv:1406.3399* (2014).
[48] https://rdf4j.org/news/2020/05/07/rdf4j-3.2.0-released/
[49] http://graphdb.ontotext.com/documentation/9.2/free/devhub/rdf-sparql-star.html
[50] https://www.stardog.com/docs/#_edge_properties
[51] https://blog.blazegraph.com/?p=716
[52] https://www.cambridgesemantics.com/anzograph/

changing this later on to a value object to be able to add annotations means restructuring the entire knowledge graph as well as rewriting queries that might already exist to work with the data.

RDF* is itself not a vocabulary for expressing any particular type of annotation (such as time, probability or provenance). However, as shown previously, various existing and well-established vocabularies exist for the various annotation use cases, and such vocabularies can be used in combination with the RDF* model. In the course of the FROCKG project, we will establish practices to efficient modeling of annotations using the various existing vocabularies and modeling approaches such as named graphs and RDF*.

Incremental knowledge extraction

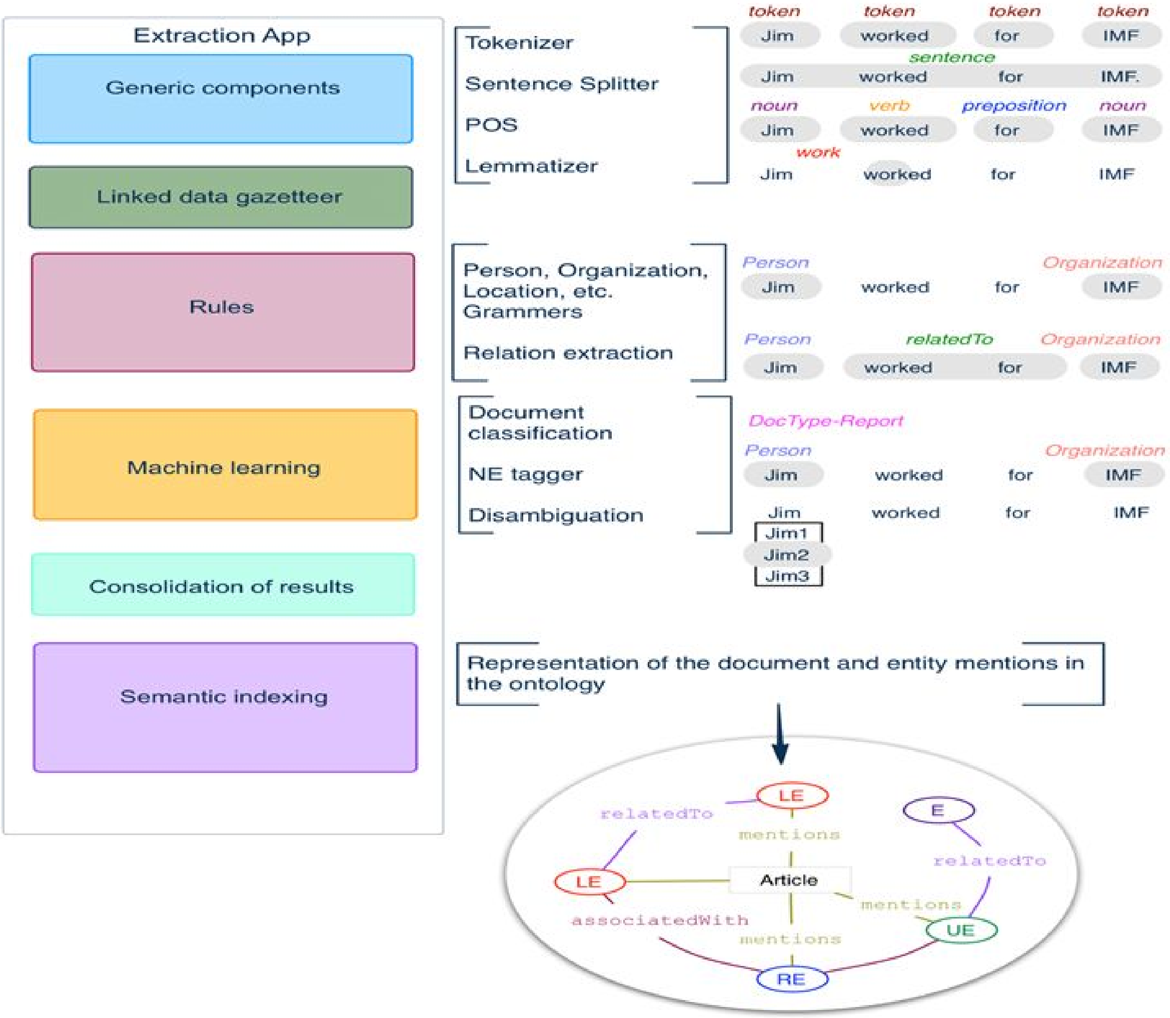
In the context of knowledge graph generation, knowledge extraction covers three key data categories - master data, meta data and unstructured content (mostly text, but also multimedia). Knowledge graph generation is a complex iterative process that usually requires integration of heterogeneous data. On the other hand, the source data can be quite dynamic. Thus the real value of a knowledge graph can be demonstrated when it does not just represent a snapshot of the current knowledge in the domain but is a live system with constant updates.

Master data

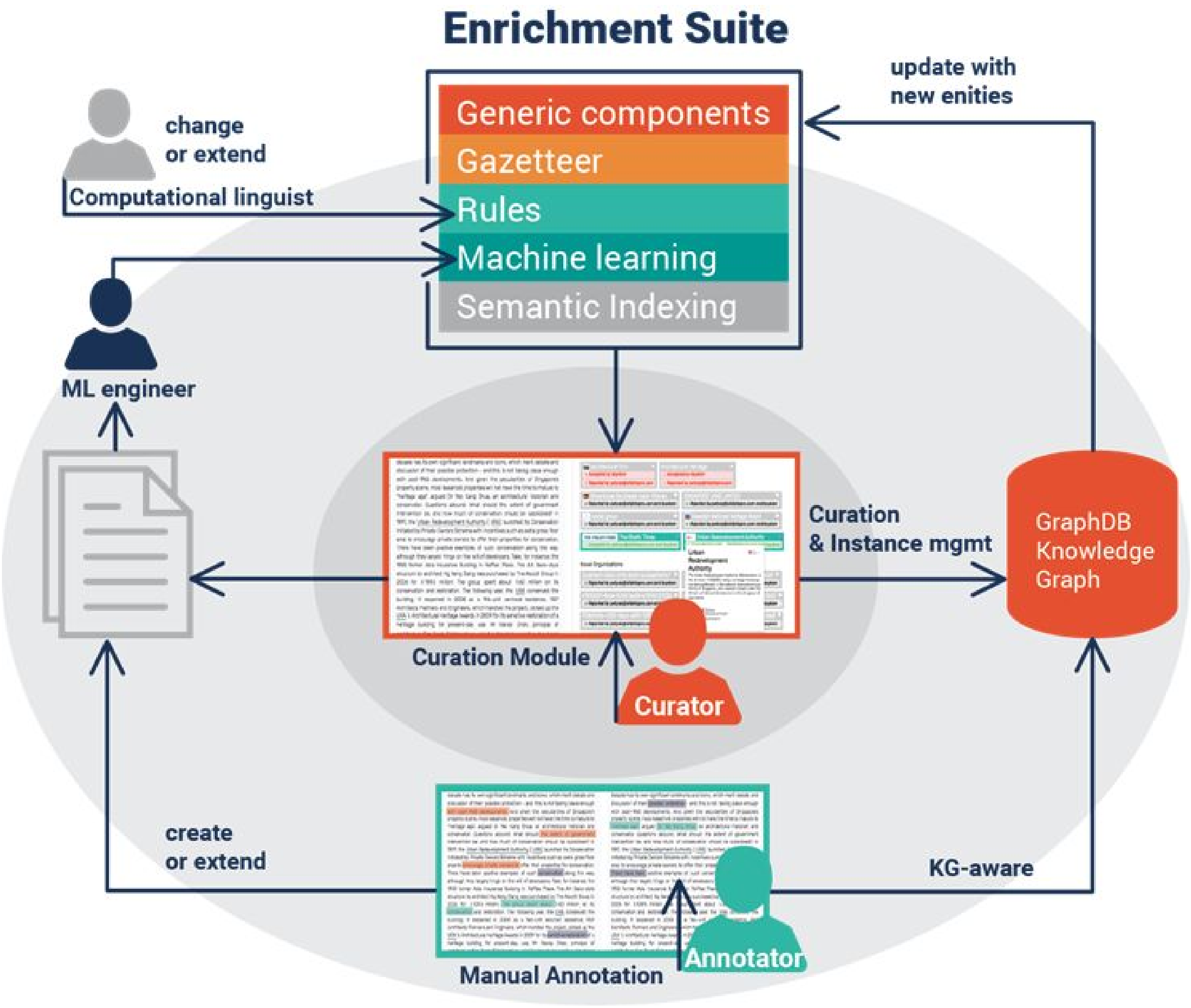
Master data is frequently used in knowledge graph generation and thus serves as a main source for information extraction. Although in general master data is well structured, it usually lacks semantic normalization and alignment with core ontologies, which will further allow simple (parent/child hierarchical relations) and even more complex inferencing based on ontology schema. A classical approach is to implement mapping of textual values to ontology concepts as part of the Extraction Transformation and Loading (ETL) process for each data set that needs to be included in the knowledge graph. This step will guarantee proper semantic normalization of the values and fusion with the ontological model that then can be applied on instance level.

Unstructured content

Knowledge extraction from unstructured content can be backed up by the instances and the relations between them expressed in the underlying knowledge graph. Besides the generic Natural Language Processing (NLP) components (tokenizers, sentence splitters, part-of-speech taggers, lemmatizers), there are other components (e.g. Gazetteers) which are tightly coupled with the knowledge graph and allow creation of semantic annotations within the content.



As a next step, the knowledge extraction process involves semantic fusion of the extracted data with the information within the graph. The resulting “new version” of the knowledge graph can be used to populate a new version of the NLP pipeline and to extend it with the newly extracted concepts and relations from the previous iteration.



Although knowledge graphs are constantly evolving (new instances and relations between them are provided by the primary data sources) there is always a need for manual extension of the knowledge graph with new instances based on the expert knowledge of the curators/annotators. The need for adding new instances is identified in manually processing/validation of new content in the cases when the knowledge graph and the associated NLP pipeline fail to identify concepts in the scope of the extraction process. Once these new instances become part of the knowledge graph (through a defined instance management process coupled with the manual annotation/validation of content), the associated NLP pipeline will be incrementally extended with the new instances and all newly processed content will be annotated with the new extended version of the pipeline.

Technology and Solutions provided by the partners

Paderborn University provides state of the art approaches for fact checking, namely FactCheck and COPAAL. In addition to that, tools necessary for the development and evaluation of fact checking and knowledge graph validation approaches can be provided, e.g., GERBIL KBC,[53] which can be used for the evaluation of newly developed approaches. In addition, libraries for transforming RDF data into natural language text can be provided.[54]

Metaphacts provides the metaphactory platform for knowledge graph management[55], based on which sector specific solutions are built. While the platform itself is generic, key sectors include finance, cultural heritage, enterprise information management and engineering / industrial applications. In particular, metaphactory offers components for dynamic integration of distributed structured and unstructured data sources through Ephedra[56], and a platform API for upstream and downstream integration with other tools. It has built-in support for the use of SHACL as a data validation mechanism. In addition, it provides advanced data visualization and authoring components, and a highly customizable semantic templating system that enables quick creation of case-specific navigation/search/browsing capabilities and integration of third party visual components.

Sirma AI contributes GraphDB[57]—a highly efficient, robust, and scalable RDF database. GraphDB and associate tooling streamlines the semantic data integration of heterogeneous data (linked open data (LOD) datasets, proprietary data sets and information extracted from unstructured content) into focused knowledge graphs. GraphDB implements the RDF4J framework interfaces, the W3C SPARQL Protocol specification, and supports all RDF serialization formats. GraphDB can perform semantic inferencing at large scale (billions of triples), allowing users to derive new semantic facts from existing facts. It handles massive loads, queries, and inferencing in real time. The database supports various usage scenarios because of it's enterprise level features such as cluster support, integration with external high-performance search applications (Lucene, Solr, and Elasticsearch) and support for RDF* and SPARQL*[58].

Most of the target data sets that will be used for building a knowledge graph are not available in RDF. The transformation of the proprietary formats into a corresponding RDF representation will be based on ETL tools and platforms (like TALEND Open Studio[59]). The result ETL scripts will be made available for the project.

Various components from Ontotext Dynamic Semantic Publishing (DSP)[60] will be required for the identification and processing of content relevant for the different use cases. This includes the DSP News feeder, which will need to be configured to ingest content from specific news portals, and Concept Extraction Service (CES) which is a customized Natural

[53] http://gerbil-kbc.aksw.org/gerbil/

[54] Ngomo, Axel-Cyrille Ngonga, Röder, Michael, Moussallem, Diego, Usbeck, Ricardo and Speck, René. "BENGAL: An Automatic Benchmark Generator for Entity Recognition and Linking." In Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018, 2018.

[55] Haase, P., Herzig, D. M., Kozlov, A., Nikolov, A., & Trame, J. (2019). metaphactory: A platform for knowledge graph management. *Semantic Web*, *10*(6), 1109-1125.

[56] Nikolov, A., Haase, P., Trame, J., & Kozlov, A. (2017, November). Ephedra: Efficiently combining RDF data and services using SPARQL federation. In *International Conference on Knowledge Engineering and the Semantic Web* (pp. 246-262). Springer, Cham.

[57] GraphDB product description: https://www.ontotext.com/products/graphdb/

[58]Foundations of an Alternative Approach to Reification in RDF https://arxiv.org/pdf/1406.3399.pdf

[59] https://www.talend.com/products/data-integration/data-integration-open-studio/

[60]Semantic Tagging: https://www.ontotext.com/solutions/semantic-tagging/

Language Processing (NLP) pipeline able to semantically annotate unstructured content with concepts from the knowledge graph.

Zazuko contributes tooling for data pipelining like its Expressive RDF Mapper XRM[61] and its streaming pipelining system barnard59[62]. This tooling is used in its contributed use-cases as well in the domain of archival records management.
Zazuko is one of the main initiators and contributors for the RDFJS W3C Community Working Group[63] and contributes and maintains various RDF JavaScript libraries. Among others, it provides a SHACL implementation for JavaScript. These libraries are used in HTML Web Components that are used in the domain of archival records management. The components will be used and extended for the FROCKG project.

Feasibility of the Fact Checking engine

Based on the use cases we will deduce required steps to confirm a feasible development. However, this section will not go into great detail due to the ongoing user requirement elicitation (See deliverable D1.3).

The general state of the art and more specifically, the expertise and components that the consortium partners bring to the project put the FROCKG project in a strong position to deliver a technically feasible solution to automated fact checking at scale for the various use cases.

However, it has to be taken into account that in open world scenarios, fact checking can only find supporting evidence, but not assume that unknown data implies that a fact is not true. This means that some high-level requirements as specified in the various use cases, in particular around the notion of “detecting false information”, are considered out of scope for FROCKG. In addition, not all requirements derived from the use cases are achievable within the project runtime. To ensure the feasibility and the success of the project, the important requirements, which represent core functionalities of the FROCKG project, will be prioritized over optional requirements.

Conclusions

The goals, visions and gains of the FROCKG project presented in the short use case descriptions name various technologies. These technologies have been foreseen or partially tackled with state-of-the-art approaches as pointed out in the previous sections. However, the combination, scale and enterprise relevancy of the use cases demand novel approaches towards the implementation of the FROCKG Fact Checking engine and related components. Thus, we will further define the architecture (deliverable D1.2) and the exact specifications and requirements (deliverable D1.3) to ensure a concise working goal.

[61] Expressive RDF Mapper XRM: https://zazuko.com/products/expressive-rdf-mapper/
[62] https://github.com/zazuko/barnard59
[63] RDF JavaScript Libraries: https://rdf.js.org/

Summarizing the preliminary approaches, the capabilities of the partners and the project goal, the FROCKG project is technically feasible.