Institutional Repositories in the Linked Open Data Cloud

Miller, Grant
School of Computing, College of Science, Engineering & Technology, University of South Africa, Gauteng, South Africa
grantmiller1@acm.org

Pretorius, Laurette
School of Computing, College of Science, Engineering & Technology, University of South Africa, Gauteng, South Africa
laurette@acm.org

Abstract

We believe that the wealth of digital humanity research resources that is available in Institutional Repositories in Southern Africa is largely overlooked because it is not machine explorable as defined by (Berners-Lee 2006). We argue that extracting research from Institutional Repositories, which (Berners-Lee 2006) describes as One-Star Linked Data, and then enriching this research and storing it as tuples in a triple store (RDF data store) will expose the research to a greater audience; make it machine explorable; and integrate it with a broader network of Linked Data, known as the Linked Open Data Cloud (LODC).

Our approach is to utilize software building blocks licensed under Free (Libre) Open Source Software (FLOSS) licences and to create a pipeline that can potentially transform any Institutional Repository into into a triple store, which can be published on the Internet as part of the LODC.

Our transformation process embraces under-resourced institutions by utilizing FLOSS components and utilizing digital resources already published on the internet in order to promote the research from One-Star Linked data to Five-Star Linked Data in (Berners-Lee 2006) taxonomy of linked data.

Keywords: Knowledge Graph, Scholarly Knowledge, Open Access, Linked Open Data, Resource Description Framework

1 Introduction

Our multidisciplinary research takes place at the intersection of these core concepts: Digital Humanities (DH), Open Access Institutional Repositories (OAIR), Knowledge Graphs (KG) and the LODC.

In its broadest sense DH research consists of two main components: data in the form of digital assets and electronic resources on the one hand, and semantic and Semantic Web technologies that may be used together to explore these data, on the other.

Semantic technologies are a fairly diverse family of technologies that have been in existence for a long time and seek to help derive meaning from information. Examples are natural language processing (NLP), data mining, artificial intelligence (AI), category tagging, and semantic search. Semantic Web technologies are a family of very specific technology standards from the World Wide Web Consortium (W3C) that are designed to describe and relate data on the Web and inside enterprises. These standards include a flexible data model (RDF), schema and ontology languages for describing concepts and relationships (RDFS and OWL), a query language (SPARQL), a rules language (RIF), a language for marking up data inside Web pages (RDFa), etc. [1] We specifically take the view that a lack of language technologies limits the DH research available for that language. Our opinion is that the availability of both electronic resources and language tools are essential to promote novel research in that it allows completely new questions to be asked. So, for successful DH research the following core aspects must be available:

1. machine understandable data
2. the language tools for asking innovative questions about this data

We take a high-level view when considering the OAIR, and we think of them as being digital repositories of research. The research available in these

repositories is as diverse as the faculties, and researchers affiliated with the institution. (Suber 2012) states that 'Open access (OA) literature is digital, online, free of charge, and free of most copyright and licensing restrictions.' Most of the University Institutional Repositories have at least partial OA resources, which typically depends on their level of commitment to the OA philosophy. The OA portion of these repositories is appealing to us because they do not require any special permissions to access; the licensing is open which allows us to enhance and re-publish. We believe it important to respect the copyright and licensing model that the research was published under. The portion of the Institutional Repositories that has been closed, in that it requires a valid login to access or there is a restrictive licensing model in place, is therefore omitted from our research.

The UNISA Institutional Repository (UIR) is one such repository that has both OA and closed resources. We utilized the OA portion of the UIR as the basis for a corpus that could be used for experimentation. We refer to this corpus as OAUIR, which refers to the open access portion of the UNISA Institutional Repository.

A key goal of our research was to ensure that our output would be machine searchable. Underpinning the concept of machine searching is firstly the idea that the data must be 'understandable' by a machine before it can be traversed, searched and explored. While traditional data stores, such as relational databases can be used to store this information, there is a far more flexible, concise and standardized technology to store data for machine processing. Linked Data is a concept formulated by Tim Berners-Lee who is responsible for inventing the Internet. (Berners-Lee 2006) proposed the concept of Linked Data (LD), where he defines the fundamental rules for linking data in a machine understandable way on the Internet. The following list summarizes Berners-Lee's rules:

1. Identify with URIs: This rule states the need to uniquely identify data elements by using a Uniform Resource Identifier (URI).
2. Use HTTP URIs: This rule states that the URI's must use the HyperText Transfer Protocol (HTTP) which is the standard protocol used for the Internet.
3. Serve information on the web against a URI: This rule identifies the need to have both the ontologies available and the actual data sets on the Internet using URIs.
4. Make links everywhere: This rule states the need to connect information over the Internet.

(W3C 2014) have evolved this definition slightly to include Internationalized Resource Identifier (IRI) which extends the URI concept to include encoding to include all Unicode characters but maintains the syntax defined in RFC3987-2005 by (Duerst & Suignard 2005). This extension allows for global adoption of IRI's because of the vast language support provided by Unicode, (Unicode-org 2020). Although, we should mention that there are still unsupported languages and that Unicode does not currently support universal text encoding, (Unicode-org 2022).

At the intersection of LD and OA we find Linked Open Data (LOD) which is a unification of these terms, providing linked data that is open access. We discuss this in more depth in the Section 3.

(W3C 2014) refined Berners-Lee's concepts in defining the Resource Description Framework (RDF), which states "An RDF triple consists of three components:

- the subject, which is an IRI or a blank node
- the predicate, which is an IRI
- the object, which is an IRI, a literal or a blank node"

This concept of an RDF is visualized in Figure 1 and an example is shown in Figure 2.

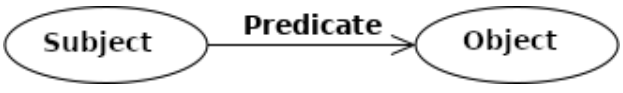

Figure 1: RDF Triple (W3C 2014)

(W3C 2014) further describes RDF triples as "The core structure of the abstract syntax is a set of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link." This definition of subject-predicate-object may also be referred to as an RDF triple because of the three entities that constitute the RDF definition. Leading on from this, we refer to a collection of triples as a triple store. These RDF triple stores are purpose built data stores for Linked Data.

'A KG, also known as a semantic network, represents a network of real-world entities—i.e. objects, events, situations, or concepts—and illustrates the relationship between them. This information is usually stored in a graph database and visualized as a graph structure, prompting the term knowledge "graph." A KG is made up of three main components: nodes, edges, and labels. Any object, place, or person can be a node. An edge defines the relationship between the nodes' (IBM 2021). For example, a node could be a person, like Berners-Lee, and an object like the Internet. An edge would be to categorize the relationship as invention, between Berners-Lee and Internet (see Figure 2). As in Figure 1, these nodes connected by a directed edge is analogous with the definition of an RDF triple.

'The heart of the KG is a knowledge model – a collection of interlinked descriptions of concepts, entities, relationships and events where:

- Descriptions have formal semantics that allow both people and computers to process them in an efficient and unambiguous manner;
- Descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities related to it;
- Diverse data is connected and described by semantic metadata according to the knowledge model.' (Ontotext.com 2018)

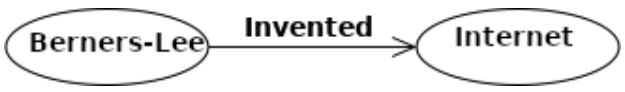

Figure 2: RDF Triple Example

Therefore in this article, a RDF triple store may be described as a KG.

KGs that utilize the RDF open standard, can be queried using SPARQL, which is another W3C open standard. (W3C 2013) states that SPARQL is 'a set of specifications that provide languages and protocols to query and manipulate RDF graph content on the Web or in an RDF store'. One of the key strengths of the SPARQL query definition is to allow queries across distributed RDF triple stores, also known as federated queries.

So, at the intersection of DH, OAIR, KGs and LD lies the ability to query these machine readable data in more sophisticated ways. Indeed, performing sophisticated queries using machine understandable data is at the heart of successful DH. In this article we present a pipeline that converts an OAIR into a KG based on FLOSS, so that it is reusable in any under-resourced environment.

The structure of this article is as follows: Section 2 briefly covers related work about KGs. Section 3 introduces the LODC. Section 4 constitutes the body of the paper and presents the transformation pipeline from one-star LOD to five-star LOD. In Section 5 we discuss our pipeline results and Section 6 concludes the article and discusses future work.

2 Related Work

While extensive research on KGs have been done in recent years, it falls outside the scope of this article to provide and extensive overview. Well-known examples are Google as the first major search engine to adopt the KG for search (Singhal 2012), Bing who mentioned that their popular search engine Bing[2] uses a KG for search (Microsoft.com 2017) and Amazon disclosed (Flint 2021) that Amazon Alexa[3] was built using a KG. It should be noted that these implementations (Google Search,

Microsoft Bing and Amazon Alexa) of KGs allow for public searching, however, the underlying KGs (triple stores) that power these search engines remain proprietary and hence closed.

The Semantic Scholar[4] and Microsoft Academic KG[5] are examples of academic search engines based on a KG that is useful for querying academic articles. The Microsoft Academic KG (Färber 2019) and (Färber & Ao 2022) is available for SPARQL queries[6] and thus it may be considered a part of the LODC. Unfortunately, this is not true of the Semantic Scholar, which does not expose their underlying KG using a SPARQL endpoint and therefore cannot be considered a part of the LODC.

Examples of KGs applied to library resources such as institutional repositories are (Sadeghi et al. 2017), (Zhang 2019) and (Jin & Sandberg 2019), which further supports our approach.

3 Linked Open Data Cloud

The LODC is a collection of KGs built on well-defined open standards, such as the W3C RDF standard, and adheres to the OA philosophy. (Berners-Lee 2006) continued his definition of Linked Data and described this five-level taxonomy, which we refer to as the (Berners-Lee 2006) 5-star classification of linked open data:

> One-star (⋆) LOD refers to data in any format that is published on the in Internet with an open license. The use of an open license qualifies this data as Open Data.
>
> Two-star (⋆⋆) LOD refers to machine-readable structured data. For example, using excel instead of an image scan of a table.
>
> Three-star (⋆ ⋆ ⋆) refers to machine-readable structured data in a non-proprietary format. For example, using a Comma Separated Values (CSV) format instead of the Microsoft proprietary format.
>
> Four-star (⋆ ⋆ ⋆⋆) LOD refers to all of the above and includes using open standards from the W3C (RDF and SPARQL) to identify things, so that people can point to your data.
>
> Five-star (⋆ ⋆ ⋆ ⋆ ⋆) LOD refers to all of the above and the data is linked to other people's LOD to provide context.

Figure 3 shows this linked data classification visually.

It is the Five-star (⋆ ⋆ ⋆ ⋆ ⋆) LOD that sets the standard for the LODC, which is the assemblage of openly published KGs that are interconnected on the Internet and provide the greatest value for machine searchability. To begin to grasp the vastness of this cloud of data (McCrae 2021) provides a small visualization, shown in Figure 4, and this only shows KGs that have been submitted and adhere to these guidelines:

- There must be resolvable http:// (or https://) URIs.
- They must resolve, with or without content negotiation, to RDF data in one of the popular RDF formats (RDFa, RDF/XML, Turtle, N-Triples).
- The dataset must contain at least 1000 triples.
- The dataset must be connected via RDF links to a dataset that is already in the diagram. This means, either your dataset must use URIs from the other dataset, or vice versa. We arbitrarily require at least 50 links.
- Access of the entire dataset must be possible via RDF crawling, via an RDF dump, or via a SPARQL endpoint.

The diagram of the LODC, shown in Figure 3,

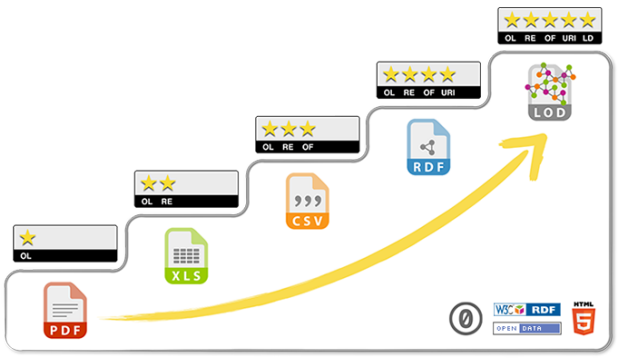

Figure 3: (Berners-Lee 2012) 5-star taxonomy

contains 1301 KGs with 16283 links between these KGs, as of May 2021, (McCrae 2021). This is by no means a comprehensive view of the LODC, but rather a sampling that allows us to begin to grasp the breadth and depth of the LODC.

Underpinning the LODC KGs are the vocabularies that have been used to define them. (Vandenbussche et al. 2017) curates one view of these Linked Open Vocabularies (LOV's) as shown in Figure 5, which contains 777 vocabularies. In the center of the LOV diagram, we find a well-used ontology, the Dublin Core terms ontology (dcterms)[7], which was one of the initial ontologies to gain adoption on the Internet. The Dublin Core onotologies are licensed under a Creative Commons licence[8], and therefore available for sharing and adaptation as long as there is attribution and any redistribution should be under the same Creative Commons license.

We view ontologies as providing the vocabulary with which we can describe data, hence utilizing a well-adopted ontology allows us to connect with a broader group of researchers. More formally, (W3C 2015) states "Vocabularies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms."

(W3C 2015) also declares that "The trend is to use the word 'ontology' for more complex, and possibly quite formal collection of terms, whereas 'vocabulary' is used when such strict formalism is not necessarily used or only in a very loose sense. Vocabularies are the basic building blocks for inference techniques on the Semantic Web."

Our decision to use the Dublin Core as our primary ontology was based on these reasons. Namely, the widespread adoption and the Creative Commons licensing.

We now discuss how these various concepts were consolidated into our transformation pipeline.

4 Transformation pipeline

Conceptually, the transformation of the available Open Data in the OAUIR was our starting point, and this data needed to be processed and enriched through (Berners-Lee 2006) classification system to produce the final result. The pictorial view of this conversion process is shown in Figure 6.

Before we review the documents discovered on the OAUIR and discuss their classification, it is important to note that we discovered several image files (JPEG, PNG and related) that fall below the One-star classification. These resources were excluded from deep analysis due to our limited resources, and we selected to focus on the bulk of the available resources.

We made use of a message queue (RabbitMQ[9] to loosely couple several microservices and create a transformation pipeline. This collection of microservices using messages that contained the state of the process allowed us to develop stateless dedicated microservices to perform specific pipeline tasks. The utility of stateless microservices is that they enable horizontal scalability.

As we developed our pipeline, we discovered that we did need some shared state, especially to reduce processing time when enriching resources. We elected

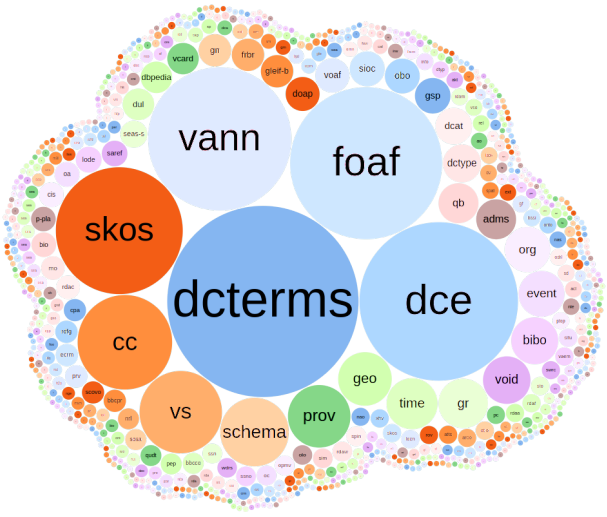

Figure 4: (Vandenbussche et al. 2017) Linked Open Vocabularies

to utilize Redis[10], licensed under the three clause BSD license[11], as our in-memory key value store for storing state, such as a key word with its definition. Our use of this in-memory data store to look up a word with its definition prevented us from calling publicly available API's that were slow to respond. The public internet API's often took several seconds to reply, and were always rate limited. We developed code to monitor and adhere to the calling rates of the external API's. Using a cache to store these results helped us stay within the bounds of the maximum calling frequency of these APIs. This lead to other challenges, such as the need to queue these requests. Another approach which we also used was to have a preferred service and an alternative service. The microservice would use the preferred service until it utilized the allowed quota, and then would automatically switch to the secondary service. When the primary service became available again, the service would switch to using it. This worked well, except when both primary and secondary had exhausted their quotas. When this happened we had to pause processing for a day, to receive fresh quotas, before resuming.

In our under-resourced environment, which utilized a shared physical volume for the data sinks (Fuseki triple store and Elasticsearch cluster) and consequently experienced 100% utilization frequently, causing blocking of writes at a disk level. The best way to overcome this would have been to utilize separate disks for each of the data sinks and as each of these is a service in its own right would be to separate these out of the virtual environment where the core enrichment pipeline runs. Ideally our entire pipeline should be co-located in the same region and data center in a cloud environment to reduce network latency. Unfortunately, we were not able to test our theory of this system scaling to reduce disk write contention due to the project's constrained resources. Hence, we adopted a pragmatic approach of retrying failed write operations. This was useful to understand that the bottleneck at a disk write level caused our thread to block until the write was complete (or had failed). This delay in writing caused our thread pool to be saturated for longer, which meant that our Task Queue could not be processed. We could therefore not read from the Message Queue because there was no capacity available in the Task Queue. This meant that the service publishing to the Message Queue eventually received an error from the Message Bus that messages could not be accepted because of the downstream queue being filled to capacity. This caused us to introduce the Binary Exponential Backoff algorithm to decrease the frequency of retries, using an exponentially increasing delay, when publishing a message from this upstream service. (Goodman et al. 1988) describes the binary exponential backoff algorithm in the context of networking packets, which is closely related to our system's messages. This build up of messages because of a processing bottleneck is usually referred to as 'back pressure'. There were multiple causes for back pressure in our prototype, including the use of slow API calls and shared disks. (Tassiulas & Ephremides 1990) first described the concept of 'back pressure routing algorithm' in which they describe how to route around the bottlenecks (or queues providing back pressure). Enhancing the prototype system to better resolve back pressure is reserved for future work. The challenges supplied by the very real back pressure in our pipeline challenged us to mitigate it by using increased service resilience, however, we believe that more could be done.

4.1 One-star (⋆) LOD

We discovered the initial data from the OAUIR through an adapted web-crawling engine using FLOSS Java core technologies. It was during this process that we discovered the resources were primarily associated with PDF documents. "The goal of PDF is to enable users to exchange and view electronic documents easily and reliably, independent of the environment in which they were created or the environment in which they are viewed or printed."(ISO 2008)

"PDF/E (ISO 24517) provides a mechanism for representing engineering documents and exchange of

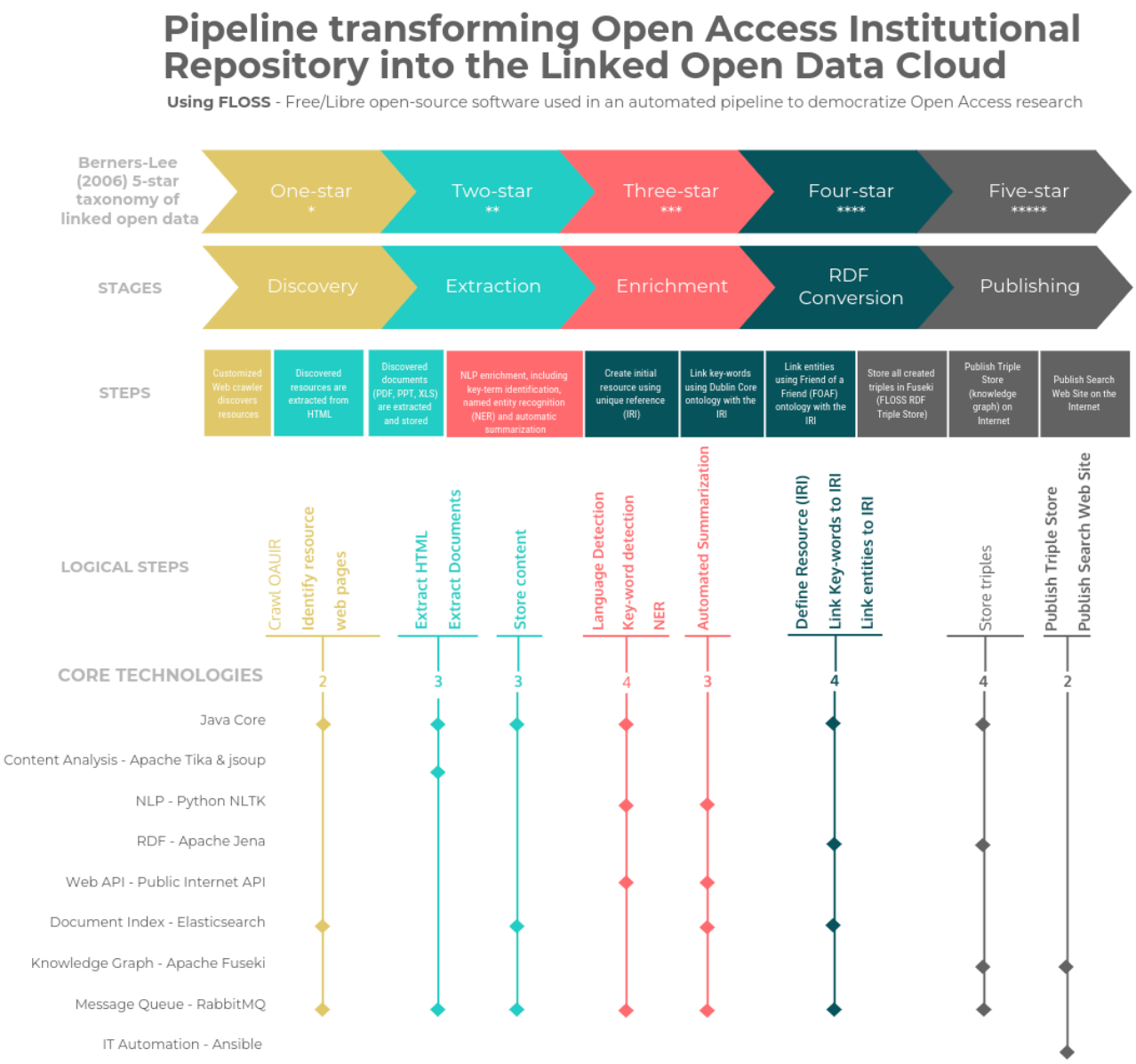

Figure 5: Transformation Pipeline

engineering data. As major corporations, government agencies, and educational institutions streamline their operations by replacing paper-based workflow with electronic exchange of information, the impact and opportunity for the application of PDF will continue to grow at a rapid pace. PDF, together with software for creating, viewing, printing and processing PDF files in a variety of ways, fulfils a set of requirements for electronic documents including:

- preservation of document fidelity independent of the device, platform, and software,
- merging of content from diverse sources—Web sites, word processing and spreadsheet programs, scanned documents, photos, and graphics—into one self-contained document while maintaining the integrity of all original source documents,
- collaborative editing of documents from multiple locations or platforms,
- digital signatures to certify authenticity,

- security and permissions to allow the creator to retain control of the document and associated rights,
- accessibility of content to those with disabilities,
- extraction and reuse of content for use with other file formats and applications, and
- electronic forms to gather data and integrate it with business systems."(ISO 2008)

Our view of this PDF definition is that it provides the ideal file format for human exchange of electronic documents, and it is unsurprising that the majority of the open access resources available on the OAUIR have an associated PDF document.

Therefore, it was critical that our transformation pipeline make provision for PDF files, and to this end we explored several alternatives before selecting Apache Tika, (Apache Software Foundation 2022) described as 'The Apache Tika™ toolkit, which detects and extracts metadata and text from over a thousand different file types (such as PPT, XLS, and PDF). All of these file types can be parsed through a single interface, making Tika useful for search engine indexing, content analysis, translation, and much more.' In addition to providing the content analysis and conversion of numerous document types, Apache Tika is licensed using Apache 2.0[12], and this is good because it is not restrictive for our research.

4.2 Two-star (⋆⋆) LOD

Each web page that we discovered, included metadata regarding the resource that we needed to extract. Central to this discovery was the identification of a unique URL, or IRI. The IRI then forms the primary reference for all future steps in the pipeline regarding this resource.

We utilized the FLOSS jsoup library[13], licensed under the MIT Licence[14] to extract information out of the HTML documents. This extracted metadata, including links to associated documents, was then packaged into a JSON object and shared on the message queue.

Our manual analysis of the attached documents in the OAUIR revealed some file types that were machine readable and structured data, which includes Microsoft PowerPoint (PPT) and Microsoft Excel (XLS) files. However, their proprietary format means that they were restricted to Two-star LOD.

Again, our choice of Apache Tika™ allowed us to parse and extract data from these files.

The next microservice to receive the message, which included links to the resource's documents was then responsible to download, parse and extract the text from each associated document. This document content and metadata was then stored in Elasticsearch.

Hence we converted the proprietary file formats into a plain text format using Apache Tika™ as a part of our conversion process. These files were stored in an Elasticsearch Index, so that they could be searched using a very powerful search engine.

(Elasticsearch 2022) explains this tool as 'Elasticsearch is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured. Elasticsearch is built on Apache Lucene and was first released in 2010 by Elasticsearch N.V. (now known as Elastic). Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of free and open tools for data ingestion, enrichment, storage, analysis, and visualization.' Hence, we utilized Apache Tika to do the initial text extraction from both One-star and Two-star data sources on the OAUIR; then we stored the extracted data into Elasticsearch, which creates indexes based on this raw extracted text. Our Elasticsearch indexes, built on this bare data, provided for very deep text search, which supported our enrichment process.

4.3 Three-star (⋆ ⋆ ⋆) LOD

In our further analysis of the OAUIR, we discovered no CSV files published on the OAUIR. The intermediate conversion of proprietary file formats (PDF, XLS and PPT) were not of interest to us for publishing because our goal was to publish an RDF triple store of the OAUIR.

However, we can think of the documents stored in Elasticsearch as being three-star LOD because they are in an open format (plain text) that is machine searchable. Our use of Elasticsearch to store our documents and then utilize Elasticsearch indexes effectively gave us a tool to search for words (or phrases) across all of the documents in the Elasticsearch cluster. These Elasticsearch indexes link to the specific words in documents that we uniquely identified with an IRI, consequently we were able to perform this search to find related research documents even when there was no obvious connection.

4.4 Four-star (⋆ ⋆ ⋆⋆) LOD

The OAUIR does not host a triple store that can be queried using SPARQL, however, there was some metadata relating to each resource was embedded in the HTML page that hosted the resource, regardless of the attached file type(s). This embedded metadata was human curated and greatly assisted us with mapping this metadata to RDF triples. We extracted this metadata from the HTML page using jsoup, as discussed above, and it provided consistent and valid data that we were able to map to RDF triples.

We utilized another FLOSS library, Apache Jena[15] licensed under the Apache 2.0 open source licence[16] for the management of RDF data in our pipeline.

This data extraction from both the HTML web pages, as well as from the attached documents and then their transformation into a collection of RDF triples was the focus of our research project.

Our transformation of data consisted of the following key aspects for each web page that our customized web-crawler discovered:

- Each OAUIR resource web page was considered to be a unique web reference, which fulfils the requirement for an IRI. This IRI may be considered as the unique address we used to associate all of the related RDF triples.
- Use the extracted metadata from the OAUIR resource web page and convert this to a set of triples using the Dublin Core ontology and associate it with the unique IRI (discovered above).
- Use the full-text from the extracted document for discovery of additional keywords. Link these keywords with the OAUIR resource web page by utilizing the IRI discovered above.
- Enrichment of these resources, identified by their IRI, then occurred as follows:
  - Apache Tika™ language detection was used on the extracted document and returns an ISO 639 code[17] describing the detected language. This detection includes a confidence indicator regarding the language returned. When thinking of the official South African languages[18], we note that this language detection tool only supports English[19]. This is a good example of where FLOSS software language tools do not support under-resourced languages.

    Consequently, we restricted our pipeline to English, as the other languages identified in the OAUIR have limited language tools available. English has mature language tools available and provides a best case for our automated transformation pipeline. We elected to defer multi-language support for future research and focused this research on building a working pipeline prototype.
  - Using Python NLP Toolkit[20], we standardized the input to ASCII from Unicode; removed the standard En-

glish stopwords; used the (Porter 1980) Stemmer to standardize words; extracted named entities; and then tokenized the Abstracts associated with each resource. This allowed us to create RDF triples for any named entities associated with the resource; as well as summarizing the resource abstract into a format which could be tweeted. This NLP process also allowed the extraction of key words that could be associated with the resource.

- The enrichment process then reviewed both the list of named entities and key words and looked up a common definition using the following strategy:
  * Check if the word had been looked up previously, if so then use the definition previously stored.
  * Utilize DBpedia Spotlight[21] to find if the word (either the named entity or keyword) appears in DBpedia and use this definition. If discovered, we store the word with its definition in an in-memory data store for faster lookup when next needed.
  * If the word was not available in DBPedia, then utilize a freely available Datamuse API[22] to lookup the meaning of the term. Next, we store the word with definition in the in-memory data store.
- Enriched terms were then linked to original resource (IRI) and stored as a set of RDF triples.

4.5 Five-star (★ ★ ★ ★ ★) LOD

The highest level Five-star LOD requires that this RDF triple store is published on the Internet with interlinks to other triple stores.

We utilized Ansible[23], an IT automation technology, to deploy our triple store and website to our Internet hosted server. The search website is available on `https://polysemous.org` domain.

This was our ultimate goal and we published the RDF triple store on a public web server with a simple search interface[24]. However, the triple store is not publicly exposed for SPARQL querying on the Internet at this time.

We have produced prototype software to extract and transform the OAUIR to a linked data triple store or KG and we believe that there may well be other OAIRs that can benefit from this work.

5 Results

As of April 2022, the OAUIR contains 22309 valid resources (unique IRI's), however at the time of the creation of the RDF triple store there were only 14388 valid resources (unique IRI's). From the discovery of 14388 IRI's, we constructed an enriched RDF triple store containing 720542 unique triples using our prototype software in 2018.

We extracted 78389 unique keywords (including named entities) from the OAUIR using our prototype software. The top twenty keywords that are most referenced are shown in Figure 7.

From these top keywords, our lookup strategy found definitions for most of the general terms. However, domain specific terms, such as 'United Party', 'National Party', 'Apartheid', 'ODL' and 'Chordophones' were not found. This is because the knowledge base for the tools that we used (Spotlight and Datamuse) are broad and not specific to the South African context. The probability of finding such triple stores specific to the South African context is low, and would probably require us defining a new triple store before proceeding. ODL, which means 'Online Distance Learning' and Chordophone which means 'a stringed musical instrument' were also not found because they are specific research domains. In future this can be resolved by finding LOD sources that are domain specific and allow these precise research terms to be linked to existing definitions. (Yarowsky 1992) states that there are significant NLP challenges when disambiguating terms, which we must overcome before find-

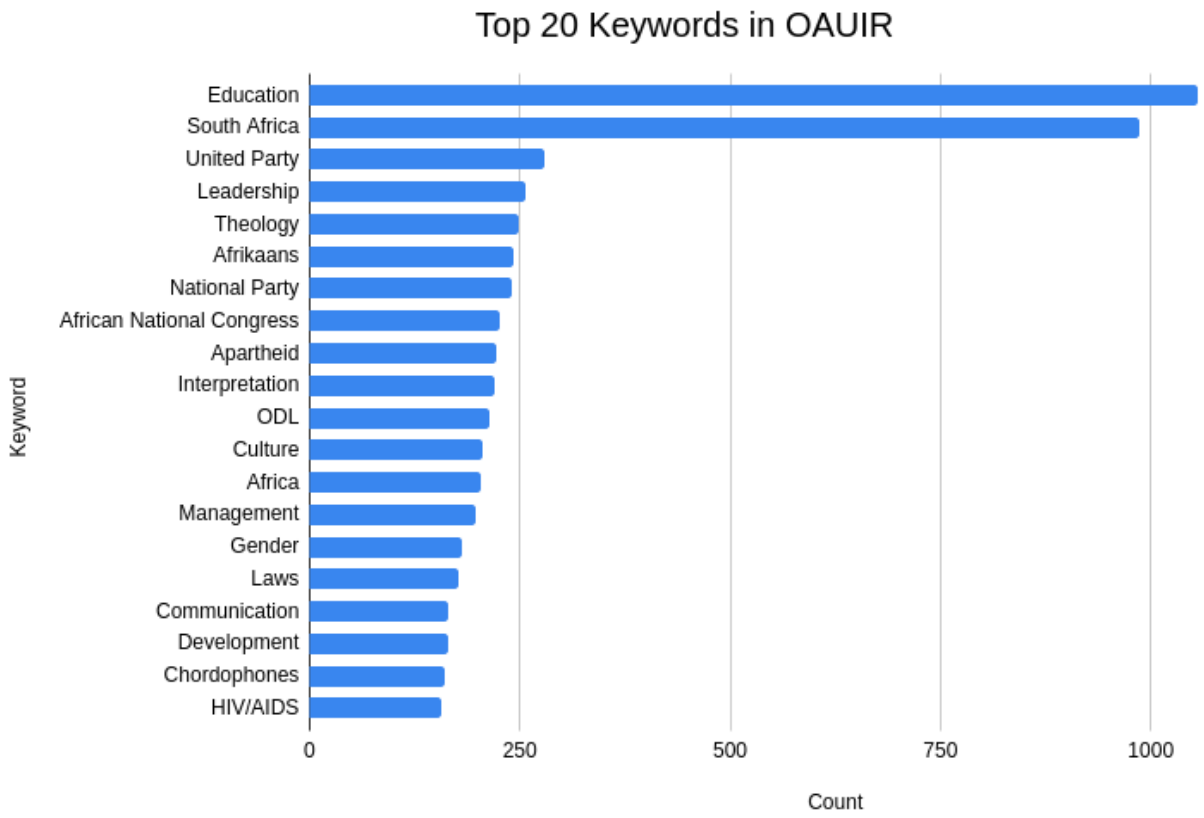

Figure 6: Top 20 Keywords in the OAUIR

ing the appropriate definition and linking it correctly.

(Dewey 1894) provided a classification and subject index cataloguing system that has been well-adopted by librarians since its inception. It is not surprising that many of the resources available on the OAUIR were tagged with a Dewey Decimal Classification (DDC) number. Only about a third of the available resources were classified with a DDC and when available, this was added to the resource's (IRI) list of attributes as an RDF triple using the DDC term from the Dublin Core ontology[25]. Of the resources (IRIs) that were classified using DDC terms, the top twenty DDC subjects are shown in Figure 8. Using the DDC sorting scheme we discovered three hundred and fifty four (354) distinct subjects. This gives a sense of the diversity of the OAUIR.

Our simple search web site offers a search based on keywords, for example, searching for Apartheid[26] gives the output shown in Figure 9. The list of available resources are shown in the left to middle of the screen, and these can be clicked on and expanded to show the abstract, keywords, links to the abstract and full-text. Clicking on any of the keywords will start a new search using the selected keyword. On the right are a list of topics extracted from the accumulated results. Hovering over any of the topics, will show the definition of the term as discovered using our API resources (described in Section 4.4 above). Clicking any of these links will open a new web page and show the definition of the word. Where possible, the link to DBpedia is used, however, if no DBpedia link was found then a link to Wikipedia[27] is used. (Head et al. 2021) explore a similar idea of explaining concepts using an augmented user interface, however, their focus is on Mathematical and Scientific formulae.

Opening one of the links to a resource on the OAUIR, we would find a web page such as the one

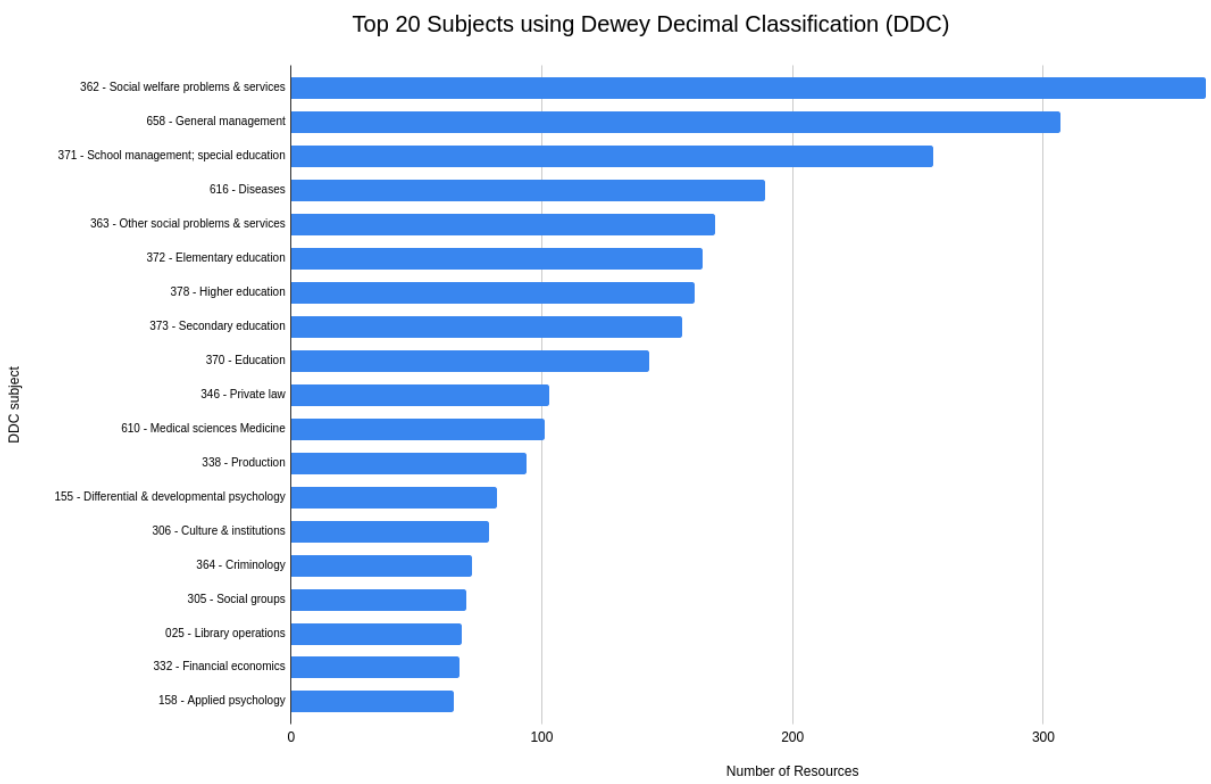

Figure 7: Top 20 Dewey Decimal Classifications in the OAUIR

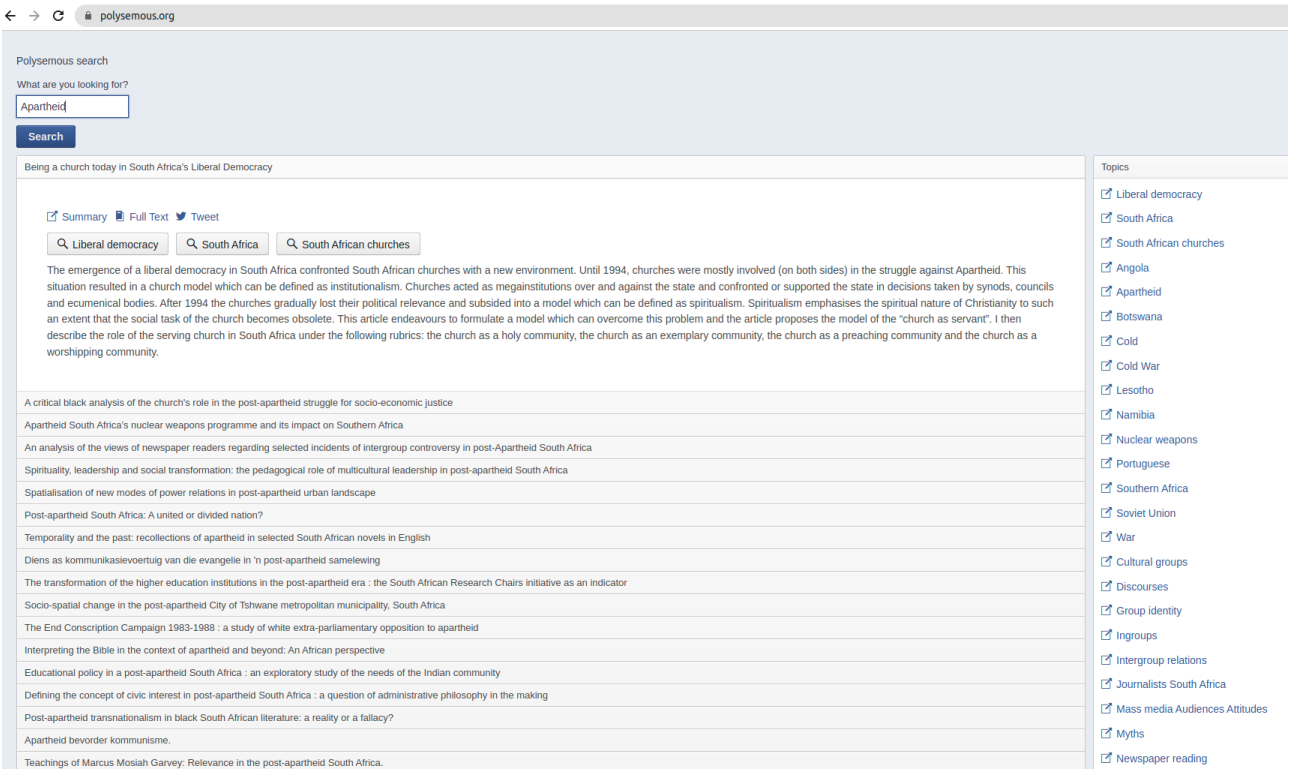

Figure 8: Search for Apartheid on the prototype web site

shown in Figure 10.

Our prototype software discovered this web page, extracted the data, enriched the data and encoded it in RDF (n3 format)[28] as shown in Figure 11. Note the use of the 'Friend of a Friend' (FOAF) vocabulary[29] to reference that the authors are acquainted. We would have preferred using an ORCID[30], however, the discovery and disambiguation of authors was outside the scope this initial research. (Subramanian et al. 2021) explore the disambiguation of author names using their S2AND model[31], available under the Apache 2.0 license, which offers an interesting starting point to deepen our implementation.

We also used the Dublin Core™ Ontology[32] to reference concepts such as title, subjects, contributors, abstract, publisher, date accepted and related concepts.

Referring back to the example search shown in Figure 9, the simplified SPARQL query against our triple store is shown in Figure 12. This SPARQL query returns a unique list of subjects (IRI's) where the Dublin Core subject matches the search query and limits the results to only the top twenty resources.

Our work in transforming the OAUIR repository in an automated way, was to create a pipeline using FLOSS technologies that could process all available resources without human intervention. The use of open source technologies allows the solution to be used in under-resourced environments. The purpose of our research was to connect the OAUIR with the LODC and provide an interface for searching this KG to demonstrate the power of the linked open data in the semantic web.

6 Conclusion and Future Work

We showed that an OAIR can be converted into a KG by demonstrating our prototype software that discovers, extracts, enriches resources available on the OAUIR and then publishes the resultant KG on the Internet with a rudimentary search interface.

We acknowledge that our work can be further enhanced. Utilizing a South African engine for detecting languages would be a good enhancement. Another enhancement would be the use of South African specific KGs to query and find definitions for words which are not included in the more generic triple stores. Similarly specialist domains such as education terms and musical instruments would also benefit from more specific triple stores. The discovery, or creation if they do not exist, of these domain specific resources remains future work. Additionally we recognise that the availability of the DDC terms in our KG can also be extended by connecting to an appropriate library classification triple store.

The use of Apache Tika™ to extract text from PDF documents was only successful when the PDF was encoded as text. Unfortunately, all styling information is lost during this process, such as chapter titles and headings. (Neumann et al. 2021) has done work in deepening text extraction from PDF files, specifically attempting to retain the structure of the document using annotations. An additional limitation of Apache Tika™ is the PDF is a collection of images, one example of such a resource is a series of handwritten letters that have been scanned as images and inserted into PDF files. In this case, Apache Tika™ is not able to extract the image as text. The conversion of handwritten text images to text and the extraction of images to text is reserved for future work. (Rajesh et al. 2021) shows an accuracy of 86.6% when converting handwriting scanned as images to text using a Convolutional Neural Network (CNN) and this shows that this conversion is possible.

We earlier noted that English language tools are mature, and we utilized these for the conversion of the bulk of the resources from the OAUIR. However, there are resources available in the other indigenous languages of South Africa that intrigue us. These languages are typically under-resourced and the availability of the necessary language tools are scarce. Discovering available language tools, or building the tools for these languages remains fu-

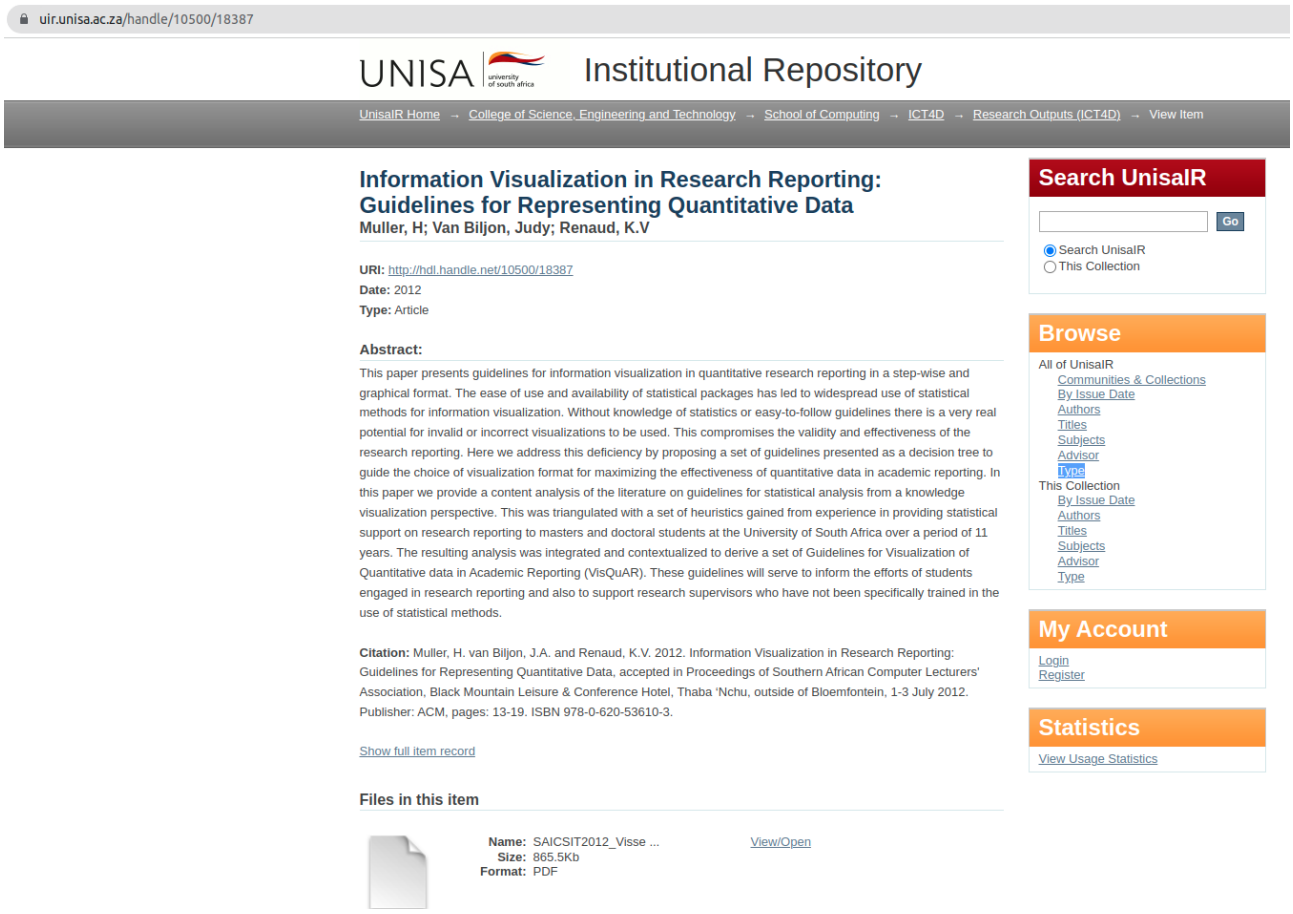

Figure 9: Example of a typical One-Star LOD OAUIR resource

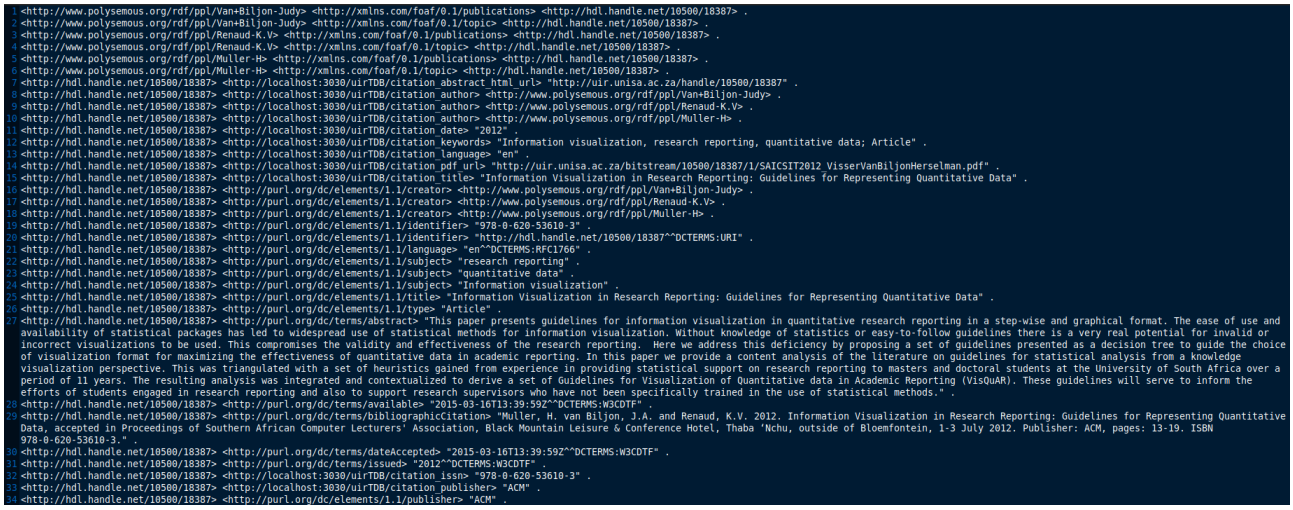

Figure 10: Example of a typical Four-Star LOD OAUIR resource

ture work.

We believe the deepening of our author disambiguation and connection with an accredited source of unique author identification will be a valuable enhancement for our research. Specifically, leveraging

research such as (Subramanian et al. 2021) for author disambiguation and use of a data source such as ORCID[33] for uniquely identifying authors will be advantageous.

From a technology perspective, we recognise our pipeline worked satisfactorily, however, more could be done to manage the back pressure caused by shared resources. Shared disk and the use of public internet API resources are shared resources that operate in seconds, rather than milliseconds.

We also recall (Flint 2021) stating that Amazon Alexa uses a KG, and we believe that a similar Artificial Intelligence (AI) implementation using our KG would be a useful future extension.

Our FLOSS based pipeline will be available for use by other organizations to transform their OAIRs into LODC resources.

Notes

[1] https://cambridgesemantics.com/blog/semantic-university/comparing-semantic-technologies

[2] https://www.bing.com

[3] https://www.amazon.com/b?ie=UTF8&node=21576558011

[4] https://www.semanticscholar.org/

[5] https://makg.org/

[6] https://makg.org/sparql-endpoint/

[7] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[8] https://creativecommons.org/licenses/by/4.0/

[9] https://www.rabbitmq.com/

```
SELECT DISTINCT ?x
WHERE{
    ?x <http://purl.org/dc/elements/1.1/subject> ?object
    FILTER regex(?object, "Apartheid", "i" )
}
LIMIT 20
```

Figure 11: A simple SPARQL query

[10] https://redis.io/

[11] https://redis.io/docs/about/license/

[12] https://www.apache.org/licenses/LICENSE-2.0

[13] https://jsoup.org/

[14] https://jsoup.org/license

[15] https://jena.apache.org/

[16] https://www.apache.org/licenses/LICENSE-2.0

[17] https://www.iso.org/iso-639-language-codes.html

[18] https://southafrica-info.com/arts-culture/11-languages-south-africa/

[19] https://svn.apache.org/repos/asf/tika/branches/1.2/tika-core/src/main/resources/org/apache/tika/language/tika.language.properties

[20] https://www.nltk.org/

[21] https://www.dbpedia-spotlight.org/

[22] https://www.datamuse.com/api/

[23] https://www.ansible.com/

[24] https://polysemous.org

[25] https://dbpedia.org/ontology/dcc

[26] https://polysemous.org

[27] https://en.wikipedia.org/

[28] https://www.w3.org/TeamSubmission/n3/

[29] http://xmlns.com/foaf/spec/

[30] https://orcid.org/

[31] https://github.com/allenai/S2AND/

[32] https://www.dublincore.org/specifications/dublin-core/

[33] https://orcid.org/

Acknowledgements

We are grateful to the UNISA Library for allowing us access to their IR so that we could perform this research.

References

Apache Software Foundation, T. (2022), 'Apache tika - a content analysis toolkit'.
URL: *https://tika.apache.org/*

Berners-Lee, T. (2006), 'Linked data'.
URL: *http://www.w3.org/DesignIssues/LinkedData.html*

Berners-Lee, T. (2012), '5 star open data'.
URL: *https://5stardata.info/en*

Dewey, M. (1894), *Decimal Classification and Relativ Index for Libraries: Clippings, Notes, Etc*, Library bureau.

Duerst, M. & Suignard, M. (2005), 'Rfc3987'.
URL: *https://datatracker.ietf.org/doc/html/rfc3987*

Elasticsearch (2022), 'What is elasticsearch?'.
URL: *https://www.elastic.co/what-is/elasticsearch*

Färber, M. (2019), The microsoft academic knowledge graph: a linked data source with 8 billion triples of scholarly data, *in* 'International semantic web conference', Springer, pp. 113–129.

Färber, M. & Ao, L. (2022), 'The microsoft academic knowledge graph enhanced: Author name disambiguation, publication classification, and embeddings', *Quantitative Science Studies* **3**(1), 51–98.

Flint, E. (2021), 'Announcing alexa entities (beta): Create more intelligent and engaging skills with easy access to alexa's knowledge'.
URL: *https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2021/02/alexa-entities-beta*

Goodman, J., Greenberg, A. G., Madras, N. & March, P. (1988), 'Stability of binary exponential backoff', *Journal of the ACM (JACM)* **35**(3), 579–602.

Head, A., Lo, K., Kang, D., Fok, R., Skjonsberg, S., Weld, D. S. & Hearst, M. A. (2021), Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols, *in* 'Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems', pp. 1–18.

IBM (2021), 'Knowledge graph'.
URL: *https://www.ibm.com/cloud/learn/knowledge-graph*

ISO (2008), 'Iso 32000-1:2008(en) document management — portable document format — part 1: Pdf 1.7'.
URL: *https://www.iso.org/obp/ui\#iso:std:iso:32000:-1:ed-1:v1:en*

Jin, Q. & Sandberg, J. (2019), 'Crafting linked open data to enhance the discoverability of institutional repositories on the web', *Qualitative and Quantitative Methods in Libraries* **7**(4), 595–606.

McCrae, J. P. (2021), 'Linked open data cloud'.
URL: *https://lod-cloud.net*

Microsoft.com (2017), 'Bring rich knowledge of people, places, things and local businesses to your apps'.
URL: *https://bit.ly/3wHRM56*

Neumann, M., Shen, Z. & Skjonsberg, S. (2021), 'Pawls: Pdf annotation with labels and structure', *arXiv preprint arXiv:2101.10281* .

Ontotext.com (2018), 'What is a knowledge graph?'.
URL: *https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph*

Porter, M. F. (1980), 'An algorithm for suffix stripping', *Program: electronic library and information systems* **14**(3), 130–137.

Rajesh, B., Jain, P., Javed, M. & Doermann, D. (2021), Hh-compwordnet: Holistic handwritten word recognition in the compressed domain, *in* '2021 Data Compression Conference (DCC)', IEEE, pp. 362–362.

Sadeghi, A., Lange, C., Vidal, M.-E. & Auer, S. (2017), Integration of scholarly communication metadata using knowledge graphs, *in* 'International Conference on Theory and Practice of Digital Libraries', Springer, pp. 328–341.

Singhal, A. (2012), 'Introducing the knowledge graph: things, not strings'.
URL: *https://bit.ly/3lvTUrf*

Suber, P. (2012), *Open access*, The MIT Press.

Subramanian, S., King, D., Downey, D. & Feldman, S. (2021), S2and: A benchmark and evaluation system for author name disambiguation, *in* '2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)', IEEE, pp. 170–179.

Tassiulas, L. & Ephremides, A. (1990), Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, *in* '29th IEEE Conference on Decision and Control', IEEE, pp. 2130–2132.

Unicode-org (2020), 'Scripts and languages'.
URL: *https://unicode-org.github.io/cldr-staging/charts/37/supplemental/scripts_and_languages.html*

Unicode-org (2022), 'As yet unsupported scripts'.
URL: *https://unicode.org/standard/unsupported.html*

Vandenbussche, P.-Y., Atemezing, G. A., Poveda-Villalón, M. & Vatant, B. (2017), 'Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web', *Semantic Web* **8**(3), 437–452.

W3C, S. W. G. (2013), 'Sparql 1.1 overview'.
URL: *https://www.w3.org/TR/sparql11-overview/*

W3C, T. W. W. W. C. (2014), 'Rdf 1.1 concepts and abstract syntax'.
URL: *https://www.w3.org/TR/rdf11-concepts/*

W3C, T. W. W. W. C. (2015), 'Vocabularies'.
URL: *https://www.w3.org/standards/semanticweb/ontology*

Yarowsky, D. (1992), Word-sense disambiguation using statistical models of roget's categories trained on large corpora, *in* 'Proceedings of the 14th Conference on Computational Linguistics - Volume 2', COLING '92, Association for Computational Linguistics, USA, p. 454–460.
URL: *https://doi.org/10.3115/992133.992140*

Zhang, L. (2019), 'Describe library resources with knowledge graph'.
URL: *http://library.ifla.org/id/eprint/2753/1/s02-2019-zhang-en.pdf*