

BACHELOR'S THESIS | LUND UNIVERSITY

Entity-based Search

Erik Zander

Department of Computer Science
Faculty of Engineering LTH

ISSN 1650-2884
LU-CS-EX 2018-22



Entity-based Search

(A take on the intelligent book)

Erik Zander

one@zandernet.net

July 8, 2018

Bachelor's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisors: Pierre Nugues, Pierre.Nugues@cs.lth.se
Markus Klang, Marcus.Klang@cs.lth.se

Examiner: Jacek Malec, Jacek.Malec@cs.lth.se

Abstract

This document describes a system to search entities in text, where we used named entity recognition to complement a traditional full text search. The named entities improve search by enabling a user to formulate queries with concepts and proper nouns and thus increase the precision of the search. Using concept and entity search, we can eliminate more easily name ambiguity and expand the search vocabulary to term variation. To carry this out, the application needs to have unique identifiers of concepts and names that are provided by Wikidata. We use these identifiers to annotate the documents, in our case a corpus of textbooks in Swedish. To annotate the documents with entities, the application uses external entity linkers through APIs. Additionally we can combine the search with the information available on the semantic web. In the future, we should be able to use the entities within the content to link content to content in the documents but also content to the web and create a product with these.

Keywords: Entity discovery and linking, Natural-language processing, Structured content, Wikidata, Semantic web

Acknowledgements

Thanks goes out to Pierre Nugues and Markus Klang for their help and input and access to the Langforia API. Thanks is also in place to Studentlitteratur for allowing the use of their content.

Contents

1	Introduction	7
1.1	Related Work	7
1.1.1	Papers and research	7
1.1.2	Semantic Scholar	8
1.1.3	AlphaSense	8
2	Approach	9
2.1	Background	9
2.1.1	Entity search	9
2.1.2	Semantic Web	10
2.1.3	Formats	10
2.1.4	Wikidata and DBpedia	12
3	Implementation	17
3.1	Overview	17
3.2	Enricher	17
3.2.1	Extractor	18
3.2.2	IBM Natural Language Processing API	18
3.2.3	Langforia	18
3.2.4	Wikidata connection	19
3.2.5	Insert data as RDFa	19
3.3	Search	19
3.3.1	Facets	20
3.3.2	Infobox	21
3.3.3	Search result	22
4	Evaluation	25
4.1	Evaluation strategy	25
4.2	Result	25
4.3	Discussion	27

5	Conclusions	31
5.1	Improvements	31
5.1.1	GUI	31
5.1.2	Facets and concept search	32
5.1.3	Search results	32
5.1.4	Concept tagging and information gathering	32
5.1.5	Speed	33
5.2	Summary	33
	Bibliography	37

Chapter 1

Introduction

The world is digitizing at an incredible speed and this affects how people expect information to be delivered. This thesis explores how to provide enhanced results when accessing and searching a large collection of documents.

1.1 Related Work

The idea to use some form of Natural-language processing (NLP) to make sense of content has been tried out in various places and some of the more related works to this project are presented in the following sections.

1.1.1 Papers and research

Silviu Cucerzan has made numerous contributions to the field of entity identification and linking. Cucerzan (2007) describes one of the first working methods for large scale entity disambiguation using data from Wikipedia.

Cucerzan later improved on this system and Cucerzan (2011) describes the improved system. The system focus on extracting entities for the whole document and disambiguate these globally for later finding the correct placement for the entity. This system was submitted to Text Analysis Conference 2011 conference.

In Rao et al. (2013), the authors provides a summary of the work in the field of entity linking. They also present a system for entity linking that use the max-margin ranking.

Sil et al. (2017) describe an Entity Discovery and Linking system were they discuss how to make entity discovery language independent.

1.1.2 Semantic Scholar

Semantic Scholar¹ is a search engine for academic papers, it uses machine learning to identify key data in papers. They aims to improve the discoverability of research papers. Semantic Scholar describes their goal as:

What if a cure for an intractable cancer is hidden within the results of thousands of clinical studies? We believe that in 20 years' time, AI will be able to connect the dots between studies to identify hypotheses and suggest experiments that would otherwise be missed. That's why we're building Semantic Scholar and making it free and open to researchers everywhere.

Currently Semantic scholar only works for papers written in English and has heavy focus on finding references between the papers and on citations.

1.1.3 AlphaSense

AlphaSense² is a commercial search application, it describes it self as “Using a blend of artificial intelligence (AI), plus advanced linguistic search and natural language processing algorithms” however as it is a commercial product little information is available.

¹<https://www.semanticscholar.org/>

²<https://www.alpha-sense.com/>

Chapter 2

Approach

2.1 Background

In this section, we give the needed background for understanding the project. We describe entity search, what it is and the benefits of it. The formats we used for the documents and data are also presented. We also describe Wikidata, DBpedia, and the semantic web.

2.1.1 Entity search

Nadeau and Sekine (2007) explained the term *named entity* as:

The term “Named Entity”, now widely used in Natural Language Processing, was coined for the Sixth Message Understanding Conference (MUC -6) (Grishman and Sundheim, 1996). At that time, MUC was focusing on Information Extraction (IE) tasks where structured information of company activities and defense related activities is extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions. Identifying references to these entities in text was recognized as one of the important sub-tasks of IE and was called “Named Entity Recognition and Classification (NERC)”.

The concept of entity-based search revolves around the idea to use identified and disambiguated named entities as the base of the search. This stands in contrast to the traditional “full text search” where a match is defined as matching the letters in the search string and the scope of the search is most of the text.

Entity-based search, on the other hand, uses the uniqueness of each entity. This needs a system to input the specific entity the user is looking for. The search results all relate to

the entity searched for but the relation can vary. The relation can be of varying degree of closeness from the specific entity to many steps away.

2.1.2 Semantic Web

Researchers started talking about the idea of a semantic web as early as the 1960s. Therefore, although the semantic web is a development and extension of the “World Wide Web”, there has been a coexisting thought on how to provide a web for data for a long time.

In early 2000s, the W3C worked on the standards for Resource Description Framework (RDF)¹ (see 2.1.3), Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL)², standards that are core components of today’s idea and implementation of the semantic web.

Allemang and Hendler (2011) is a good overview of what the semantic web is from a usability perspective, however we define it as follows:

The Semantic web is the idea to provide a machine readable web. This is guided by the specifications for RDF, RDFS and OWL. They are aimed to provide a very flexible representation of the data to increase adaption and of such the end result is heavily dependent on the publisher.

2.1.3 Formats

In this section, we give insight into the docbook and RDF formats that we use in the project.

Docbook

Docbook is a XML standard that provides a way of encoding the structure of a book/article or a set of books/articles. The docbook standard used in this thesis project is of version 5.1 and version 5.0. The standard for version 5.0 was first published by OASIS in November 2009³. One of the key features of version 5 is that it uses RELAX NG schemas in contrast to previous versions Document Type Definition (DTD).

As an example the excerpt consisting of the first paragraph (see below) of the book **Globaliseringens idéhistoria** by **Svante Nordin** is shown in Figure 2.1 in the docbook format.

Ingenstans finner man en så långvarig kulturell och politisk kontinuitet som i Kina. Det kinesiska bildskriftspråket med fullt igenkännbara om än givetvis inte oförändrade tecken går tillbaka åtminstone tre och ett halvt årtusende. Det kinesiska kejsardömet har en förmodligen lika lång historia. Reservationen hänger samman med svårigheten att skilja mellan legend och verklighet i Kinas äldsta historieskrivning.

The complete docbook standard can be found at DocBookTC (2009). In addition to this, Walsh (2010) wrote a useful book to build concrete implementations.

¹<https://www.w3.org/RDF/>

²<https://www.w3.org/OWL/>

³<http://docbook.org/specs/docbook-5.0-spec-os.html>

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <book xml:lang="sv" version="5.0"
3   xmlns="http://docbook.org/ns/docbook">
4   <?anchor xml:id="orgPage.1"/?>
5   <title>Globaliseringens idéhistoria</title>
6   <info>
7     <biblioid class="isbn"
8       >9789144004211</biblioid>
9     <biblioid class="pubsnumber"
10      >32497-01</biblioid>
11     <authorgroup>
12       <author>
13         <personname>Svante
14           Nordin</personname>
15       </author>
16     </authorgroup>
17   </info>
18   <?anchor xml:id="orgPage.2"/?>
19   <?anchor xml:id="orgPage.21"/?>
20   <chapter xml:id="isbn_9789144004211_ch_2"
21     label="2">
22     <title>De stora civilisationerna</title>
23     <section>
24       <title>Kina och Japan</title>
25       <section>
26         <title>Politik och religion i
27           Kina</title>
28         <para>Ingenstans finner man en så
29           långvarig kulturell och
30           politisk kontinuitet som i
31           Kina. Det kinesiska
32           bildskriftspråket med fullt
33           igenkännbara om än givetvis
34           inte oförändrade tecken går
35           tillbaka åtminstone tre och
36           ett halvt årtusende. Det
37           kinesiska kejsardömet har en
38           förmodligen lika lång
39           historia. Reservationen hänger
40           samman med svårigheten att
41           skilja mellan legend och
42           verklighet i Kinas äldsta
43           historieskrivning.</para>
44       </section>
45     </section>
46   </chapter>
47 </book>
48

```

Figure 2.1: Example of a DocBook encoding

RDF

The Resource Description Framework (RDF) ⁴ is an idea on how to represent data in a graph so that anyone can contribute and publish what they want. RDF also helps those with access the data to know what it is about. The idea revolves around the concept of a triple. This is data represented as Subject - Predicate - Object, as example I - has_name - Erik Zander.

There are many ways to encode the RDF information and different serialization formats. The most popular formats are N-Triples, Turtle, and RDF/XML. Given the example:

⁴<https://www.w3.org/RDF/>

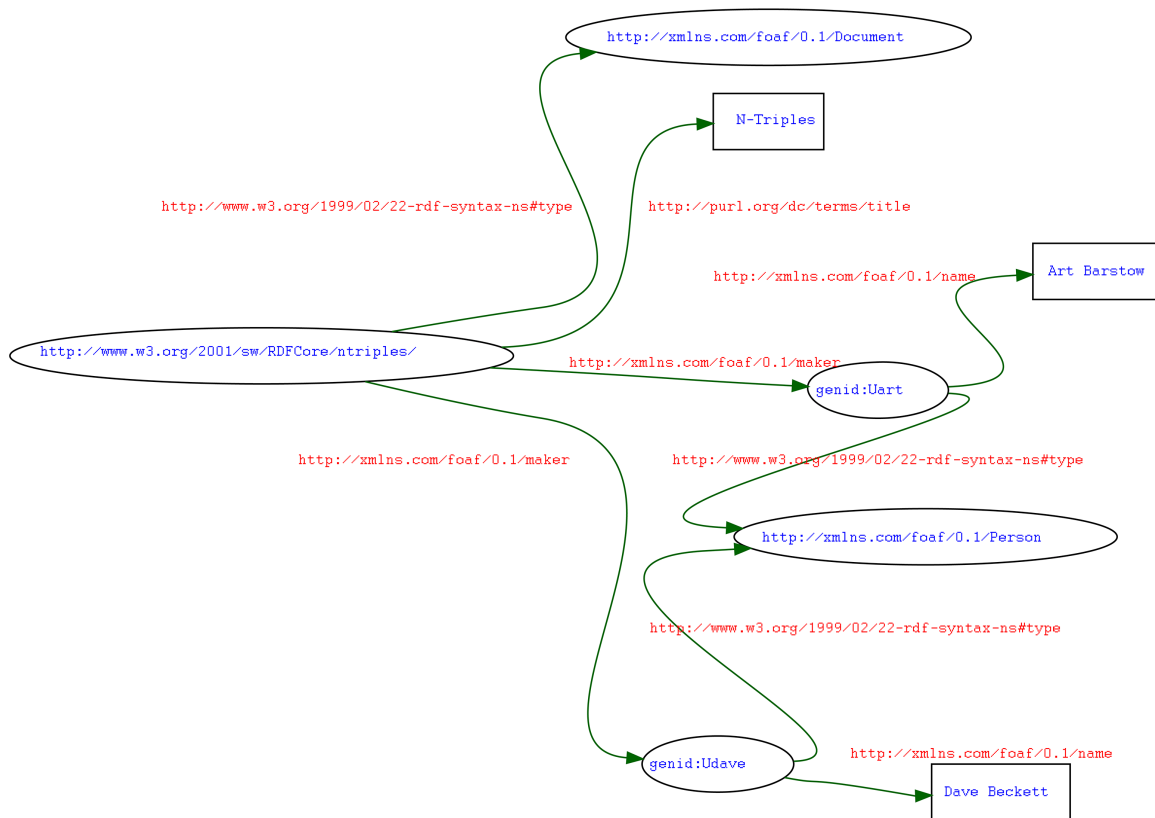


Figure 2.2: Information about a book as a graph

- the document has the title N-Triples
- the authors are Art Barstow and Dave Beckett
- both A.Barstow and D.Beckett are persons

The RDF-graph of this example can be seen in Figure 2.2 and Figure 2.3 shows the encoding.

In Figure 2.3, we can also see that, RDF uses the concept of Internationalized Resource Identifier (IRI)s to have a unified way of identifying what the subject-predicate-object is. IRIs are by design unique, and it is easy to produce and control IRIs. It is recommended that when creating IRIs this is done under a domain under the control of the creator. For example the IRIs created in this project all starts with `http://zandernet.net/namespaces/bsc/`. This way we have ensured that they will be internationally unique, as long as everyone follows the standard and recommendations. The publisher of information is responsible for keeping the IRIs unique within the domain. By having this structure, the uniqueness of each IRI is maintained without any central coordination.

2.1.4 Wikidata and DBpedia

In the project, we have used two large information resources, DBpedia and Wikidata. We present them in this section.

N-triples

```

<http://www.w3.org/2001/sw/RDFCore/ntriples/> $
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> $
<http://xmlns.com/foaf/0.1/Document> .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> $
<http://purl.org/dc/terms/title> "N-Triples"@en-US .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> $
<http://xmlns.com/foaf/0.1/maker> _:genid1 .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> $
<http://xmlns.com/foaf/0.1/maker> _:genid2 .
_:genid1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>$
<http://xmlns.com/foaf/0.1/Person> .
_:genid1 <http://xmlns.com/foaf/0.1/name> "Art Barstow" .
_:genid2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>$
<http://xmlns.com/foaf/0.1/Person> .
_:genid2 <http://xmlns.com/foaf/0.1/name> "Dave Beckett" .

```

The \$ indicate a line-break added for publication purposes N-triples does not allow line-breaks in a triple

Turtle

```

1 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
2 @prefix dc: <http://purl.org/dc/terms/> .
3
4 <http://www.w3.org/2001/sw/RDFCore/ntriples/>
5   a foaf:Document ;
6   dc:title "N-Triples"@en-US ;
7   foaf:maker [
8     a foaf:Person ;
9     foaf:name "Art Barstow"
10  ], [
11    a foaf:Person ;
12    foaf:name "Dave Beckett"
13  ] .

```

RDF/XML

```

1 <rdf:RDF xmlns="http://xmlns.com/foaf/0.1/"
2   xmlns:dc="http://purl.org/dc/terms/"
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
4   <Document rdf:about="http://www.w3.org/2001/sw/RDFCore/ntriples/">
5     <dc:title xml:lang="en-US">N-Triples</dc:title>
6     <maker>
7       <Person rdf:nodeID="art">
8         <name>Art Barstow</name>
9       </Person>
10    </maker>
11    <maker>
12      <Person rdf:nodeID="dave">
13        <name>Dave Beckett</name>
14      </Person>
15    </maker>
16  </Document>
17 </rdf:RDF>

```

Figure 2.3: RDF Serialization

Wikidata

Wikidata started out as a link cloud to link the different languages of Wikipedia articles to one identifier. The community has since added information to this identifier. As the information is crowd sourced and there is no responsible body, the data quality is very dependent on knowledge and interest of the contributing community. As of writing this (February 2018) wikidata has 43,827,650 total items.(wmflabs, 2018)

Wikidata has the property that it uses its own scheme for IRIs of entities. This lets Wikidata talk about entities and properties unrelated to Wikimedia projects.

DBpedia

DBpedia is a collection of statements and datapoints that together make up a knowledge graph. DBpedia web-page describes DBpedia as follows:

DBpedia is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects. This structured information resembles an Open Knowledge Graph (OKG) which is available for everyone on the Web. A knowledge graph is a special kind of database which stores knowledge in a machine-readable form and provides a means for information to be collected, organised, shared, searched and utilised. Google uses a similar approach to create those knowledge cards during search. We hope that this work will make it easier for the huge amount of information in Wikimedia projects to be used in some new interesting ways.

Figure 2.4 shows an example of what information is available in DBpedia. The aim is to do this in a controlled way so that data quality can be enforced. To help with the project, DBpedia is governed by The DBpedia Association founded in 2014.

The goal of the DBpedia Association is to professionalize DBpedia to surpass size and quality of closed, commercial providers (The DBpedia Association, 2018). The size of DBpedia is on the dataset download page⁵ described as follows

Altogether the DBpedia 2016-10 release consists of 13 billion (2016-04: 11.5 billion) pieces of information (RDF triples) out of which 1.7 billion (2016-04: 1.6 billion) were extracted from the English edition of Wikipedia, 6.6 billion (2016-04: 6 billion) were extracted from other language editions and 4.8 billion (2016-04: 4 billion) from Wikipedia Commons and Wikidata.

DBpedia uses the Wikimedia Project identifiers as IRIs for entities however there's an ongoing effort to introduce an independent IRI scheme to be able to talk about entities not on Wikipedia. Similar and related project is also to unify information from different languages.

⁵<http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10>

About Douglas Adams

Douglas Noel Adams (11 March 1952 – 11 May 2001) was an English author, scriptwriter, essayist, humorist, satirist and dramatist. Adams is best known as the author of *The Hitchhiker's Guide to the Galaxy*, which originated in 1978 as a BBC radio comedy before developing into a "trilogy" of five books that sold more than 15 million copies in his lifetime and generated a television series, several stage plays, comics, a computer game, and in 2005 a feature film. Adams's contribution to UK radio is commemorated in The Radio Academy's Hall of Fame.

Property Value

Property	Value
<code>dbo:abstract</code>	<ul style="list-style-type: none"> Douglas Noel Adams (11 March 1952 – 11 May 2001) was an English author, scriptwriter, essayist, humorist, satirist and dramatist. Adams is best known as the author of <i>The Hitchhiker's Guide to the Galaxy</i>, which originated in 1978 as a BBC radio comedy before developing into a "trilogy" of five books that sold more than 15 million copies in his lifetime and generated a television series, several stage plays, comics, a computer game, and in 2005 a feature film. Adams's contribution to UK radio is commemorated in The Radio Academy's Hall of Fame. Adams also wrote Dirk Gently's Holistic Detective Agency (1987) and <i>The Long Dark Tea-Time of the Soul</i> (1988), and co-wrote <i>The Meaning of Liff</i> (1983), <i>The Deeper Meaning of Liff</i> (1990), <i>Last Chance to See</i> (1990), and three stories for the television series <i>Doctor Who</i>; he also served as script editor for the show's seventeenth season in 1979. A posthumous collection of his works, including an unfinished novel, was published as <i>The Salmon of Doubt</i> in 2002. Adams was known as an advocate for environmentalism and conservation, as a lover of fast cars, cameras, technological innovation and the Apple Macintosh, and as a "devout atheist". ^(en)
<code>dbo:almaMater</code>	<ul style="list-style-type: none"> <code>en:St_John's_College,_Cambridge</code>
<code>dbo:birthDate</code>	<ul style="list-style-type: none"> 1952-03-11 ^(add date) 1952-3-11
<code>dbo:birthName</code>	<ul style="list-style-type: none"> Douglas Noel Adams ^(en)
<code>dbo:birthPlace</code>	<ul style="list-style-type: none"> <code>en:Cambridge</code>
<code>dbo:deathDate</code>	<ul style="list-style-type: none"> 2001-05-11 ^(add date) 2001-5-11
<code>dbo:deathPlace</code>	<ul style="list-style-type: none"> <code>en:Montecito,_California</code> <code>en:Santa_Barbara,_California</code>
<code>dbo:restingPlace</code>	<ul style="list-style-type: none"> <code>en:Highgate_Cemetery</code>
<code>dbo:thumbnail</code>	<ul style="list-style-type: none"> <code>wiki commons:Special:FilePath/Douglas_adams_portrait_cropped.jpg?width=300</code>
<code>dbo:wikiPageExternalLink</code>	<ul style="list-style-type: none"> <code>http://douglasadams.com/</code> <code>http://www.douglasadams.com/</code> <code>http://howladay.org/</code> <code>http://www.biola.org/people/douglasadams/</code> <code>http://www.biola.org/podcast/WDNA</code> <code>http://www.bookslut.com/nonfiction/2004_05_002057.php</code> <code>http://www.vintagemacworld.com/vitx.html</code>
<code>http://purl.org/linguistics/gold/hypermym</code>	<ul style="list-style-type: none"> <code>en:Writer</code>
<code>rstype</code>	<ul style="list-style-type: none"> <code>owl:Thing</code> <code>foaf:Person</code> <code>dbo:Person</code> <code>owl:Agent</code> <code>owl:NaturalPerson</code> <code>wikidata:Q215627</code> <code>wikidata:Q24229398</code> <code>wikidata:Q36180</code> <code>wikidata:Q5</code> <code>dbo:Agent</code> <code>dbo:Writer</code> <code>schema:Person</code> <code>yago:WikicatPeopleEducatedAtBrentwoodSchool(Essex)</code> <code>yago:WikicatPeopleFromCambridge</code> <code>umbel-ec:Artist</code> <code>umbel-ec:PersonWithOccupation</code> <code>umbel-ec:Writer</code> <code>yago:WikicatInteractiveFictionWriters</code> <code>yago:WikicatSatirists</code> <code>yago:WikicatScienceFictionBibliographies</code> <code>yago:WikicatScienceFictionWriters</code> <code>yago:Abstraction100002137</code> <code>yago:Advocate109774783</code> <code>yago:Alumnus109786338</code> <code>yago:Articulator109811712</code> <code>yago:Atheist109820044</code> <code>yago:Authority109824361</code>

Figure 2.4: DBpedia information about Douglas Adams

Chapter 3

Implementation

3.1 Overview

The application consists of following main components (also shown in Figure 3.1):

- A document ingestion engine (found in **EXT**) that takes docbook XML documents and run them through steps for enriching and ingestion;
The engine makes use of the Langforia API(**Algoritm**) and a SPARQL endpoint for wikidata (**Wiki db**) for some of the tasks;
- A Marklogic¹ database server (with a database containing xml documents **Bok XML** and graph data **Graf**) that helps with the search and indexing;
- Enriching queries that bring in additional data into the search results (triggerd by the ingestion engine **EXT**);
- A search application that runs in the browser.

3.2 Enricher

The enricher (part of the ingestion engine) is responsible for the collection of data, annotation into the documents and populating the graph database with semantic triples. The enricher starts from the assumption that the documents that are to be enriched already have been ingested into the database. They are therefore accessible by the database absolute URL.

¹<http://www.marklogic.com>

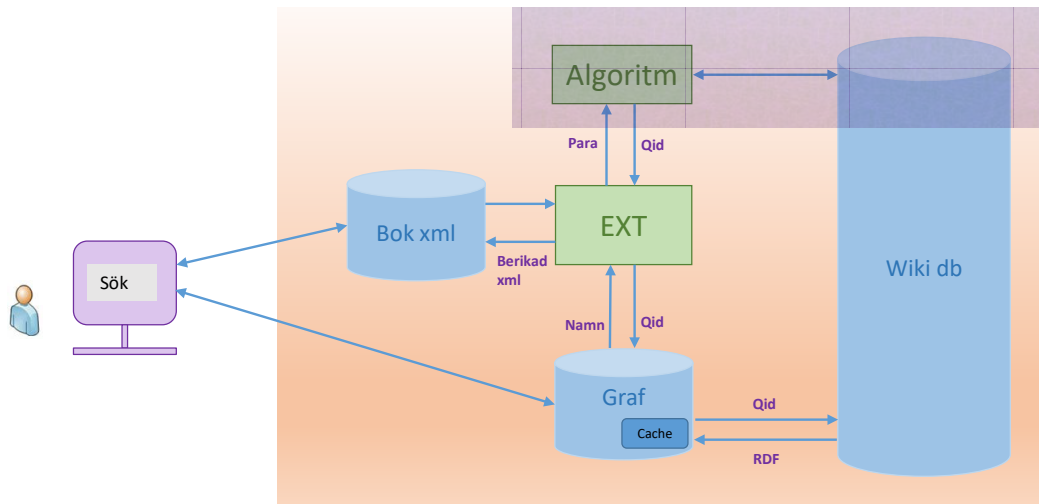


Figure 3.1: An overview of the enricher and how it fits into the application

3.2.1 Extractor

The extractor (also a ingestion engine part) has the capability to use two APIs for extracting information about the text. These are *IBM Natural Language Processing API* and *Langforia API*

In addition to extracting what the text is about, the extractor also connects to *Wiki-data.org* to extract information about the entities returned from the APIs. The extractor adds this information to the database. This way when the extractor makes a subsequent call for the same entity (identified by its Wikidata QId) the database can respond without querying the APIs.

The last thing the extractor does is inserting the data into the document and inserting this enriched document into the database.

3.2.2 IBM Natural Language Processing API

In the project, we have looked at IBM's NLP API. It was formerly know as Alchemy NLP API but IBM has since they bought renamed and integrated it into their portfolio of Artificial Intelligence and Machine Learning tools. The API has two different functions, one for named entity identification and one that delivers disambiguated concepts. The one that delivers the concepts is the one the application need. However it is not available for Swedish. As a result IBM API was never used more than in a trial run.

3.2.3 Langforia

In our application, we use the Langforia API for analyzing text. This API was developed by Markus Klang at LTH (Klang and Nugues, 2016). The extractor in the application sends the text to the API.

Our program consists of a loop, where, for each paragraph of a book:

1. We send the text of the paragraph to Langforia;
2. Langforia returns several layers of information. We extract the disambiguated entities;
3. The extractor then inserts them in the corresponding paragraph of the book.

Langforia uses Wikidata as knowledge graph and builds a graph of links from Wikipedia. From these graphs, it builds the model that is later used to identify and link the named entities. Langforia uses the page-ranks from Wikipedia and how many pages that links into a concept to determine the internal ranking of concepts. Langforia stores this information in an internal graph format.

When we send text to Langforia, it looks for mentions in the text. A mention is the term used for something that the algorithm considers a candidate to be an entity. These mentions may overlap and Langforia determines what mentions to keep using a model trained on links in Wikipedia pages. The model uses statistics as how many links referees the entity to determine what are good entities. As a second step, Langforia links the mentions to the entities in its knowledge-graph. These entities are linked to Wikidata. It then ranks the candidate entities using a local page rank. For more details on Langforia, see Klang and Nugues (2016).

3.2.4 Wikidata connection

As part of the enrichment process, the Wikidata connection queries the Wikidata SPARQL endpoint² for a predefined set of properties. The application does this for each of entities discovered by the Langforia API. The application stores the result in the database. For subsequent queries on the same entity, the application gets the data from the internal database. One of the properties extracted from Wikidata is the `RDFS:LABEL`³, Wikidata connector returns this to the calling method.

3.2.5 Insert data as RDFa

RDFa is a way to encode RDF information as attributes in a document. Here we describe the part of the enrichment engine that inserts the data as Resource Description Framework in Attributes (RDFa) in the Docbook document. As the `SUBJECTSET` element isn't used in the base content, the information is inserted as such under the `PARA` element; see Figure 3.2.

3.3 Search

The Search is the application that an end-user sees and that is the entry-point into the data on the normal day-to-day usage. The frontpage of the application can be seen in Figure 3.3.

²<https://query.wikidata.org/sparql>

³<http://www.w3.org/2000/01/rdf-schema#label>

```
1 <para>
2   <info xmlns:json="http://marklogic.com/xdmp/json/basic"
3     xmlns:zt="http://zandernet.net/namespaces/bsc/elements/transfer">
4     <subjectset prefix="zid: http://zandernet.net/namespaces/bsc/zid#"
5       resource="zid:urn:uuid:f0831681-c680-4724-b7e3-14689fe2a89a">
6       <subject>
7         <subjectterm property="http://purl.org/dc/terms/subject "
8           resource="http://www.wikidata.org/entity/wd:Q9235">
9           Friedrich Hegel
10        </subjectterm>
11       </subject>
12     </subjectset>
13   </info> Den förste av dessa är G.W.F. Hegel (1770-1831),
14   det tyska 1800-talets dominerande filosof och den
15   förste västerländske filosof som ställde historien och
16   dess problem i centrum för sitt tänkande.
17 </para>
```

Figure 3.2: An example of RDFa in docbook



Figure 3.3: Application frontpage

It runs on a Marklogic Server and is configured to use both full text indices and a specific field index to provide the ability to search for specific entities. These entities are used to provide the user with a fluent way of constraining the search by entity. How this is done is described under the following section 3.3.1.

3.3.1 Facets

The application implements an advanced form of filtering called faceted search. In the case of the application one facet corresponds to an entity. To find books with one specific facet the application relies on a field index. The server indexes the subjects of some of the RDFa triples stored in the documents. The subjects are stored as XML attributes. To specify the attributes in the documents we use XPath. XPath is a query language for selecting nodes from an XML document. The specific XPath for the attributes is:

```
//db:subjectterm
  [@property = "http://purl.org/dc/terms/subject"]
  / @resource
```

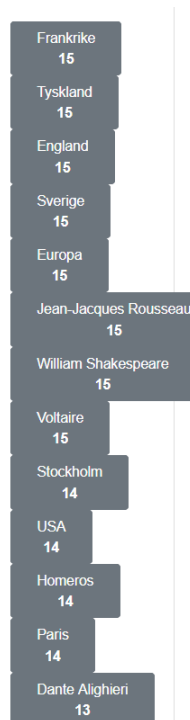



Figure 3.4: Concept based facets

This xpath looks for each `docbook:subjectterm` that has the attribute `property` with the value `http://purl.org/dc/terms/subject`. A `docbook:subjectterm` matching this criterion has a `rdf:subject` resource attribute. The value of this attribute is what is put into the index. As the server indexes these resources, it is possible to search for specific values of `rdf:subject`. We can then present the results in a sidebar in the form of facets. Given the example search:

```
voltaire kafka Subject:http://www.wikidata.org/entity/Q6527
```

This search is an *and search* of the text strings *voltaire*, *kafka* and the concept `wd:Q6527` that represents *Jean-Jacques Rousseau*.

The `wd:Q6527` is the compact form of the IRI `http://www.wikidata.org/entity/Q6527`, this IRI is the identifier for the concept in the domain of Wikidata. We refer to this as the QId and it consists of the prefix *wd* and the unique identifier *Q6527*. The prefix *wd* is just short for `http://www.wikidata.org/entity/`.

Figure 3.4 shows the facets corresponding to the search in the example.

3.3.2 Infobox

The infobox, see Figure 3.5, shows information about the entity in the search term (if multiple entities are present, it takes the first term and displays the information for that term). The information present in the infobox is taken from the graph database connected to the application server. This way, the information in the database can be controlled so that it is possible to use the Marklogic `xmdp:describe` command which is similar to a SPARQL `describe` but only returns directly related triples for that specific IRI. Take note

that the SPARQL standard does not specify how much information SPARQL returns, this is up to the implementation.



Figure 3.5: Example of the infobox

As the infobox uses the `xdmp:describe`, it takes all triples from the database that have the IRI from the search query as subject in the triple. It then looks up the predicate and makes a human readable text out of it. The look-up is handled by the wikidata API as the current data is only sourced from there. If needed, when the object is an IRI, another look-up is made for the object to provide a human readable result.

3.3.3 Search result

The complete result is both the facets, infobox, and the main search result that presents the documents that match the query. In the search result, relevant information is presented along with a snippet. As the documents in this project are books metadata such as ISBN, title and author are presented. Figure3.6 shows the full result.

Expasearch

Continue →
Give feedback →

Search Results

Sverige
259

Stockholm
233

USA
226

Europa
219

August Strindberg
212


Tyskland
195

England
188

Göteborg
181

Frankrike
175


anfallare
170



Drama i tre avsnitt

9789144095455

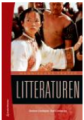
Gunnar Brandell



Nordens litteratur

9789144083322

Margareta Petersson
Rikard Schönström
Ulf Teleman




Människans texter

Litteraturen

9789144059822

Anders Lindqvist
Karl Lindqvist



Fråga föremålen

9789144089805

Anna Blom

Förord

August Strindberg vågar att mördra sin far men ändrar sig. Det är vad Strindberg skulle kalla en "ofullgånge intention", i stil med alla dem som förekommer i August Strindberg ... skriven på vers; hos Horatius blir detta ett påbud, vars makt ännu den unge Strindberg fick känna på, då han lämnade in prosaversionen av Mäster Olof till spelning på... August Strindberg ... den yngre eller Strindberg...

Kronologi

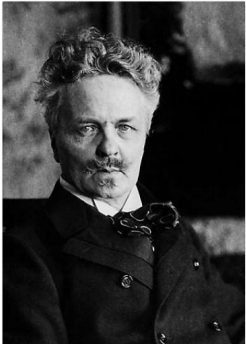
August Strindberg Denna period kallas "det moderna genombrottet" och omfattade författare som Georg Brandes, August Strindberg och Henrik Ibsen. Brandes talade om litteraturens betydelse för skapandet av en "Verdensborgerfølelse" ... August Strindberg ... med William Blake, Charles Baudelaire, Honoré de Balzac och August Strindberg i spetsen söka sig till hans korrespondenslära för...

Vad kännetecknar ett epos?

August Strindberg ... på 1800-talet skrevs hexameteros av exempelvis Esaias Tegnér, Johan Ludvig Runeberg – och August Strindberg. August Strindberg August Strindberg August Strindberg och Strindberg och tysken August Strindberg August Strindberg August Strindberg August Strindberg August Strindberg ... äkte Strindberg till Uppsala för att studera till läkare. Det blev inte mycket av...

Vad kan föremål säga om historisk tid?

August Strindberg August Strindberg August Strindberg August Strindberg August Strindberg ... av



August Strindberg

Commons-kategori (relaterad kategori) på August Strindberg

Commons utan prefixet: Category: August Strindberg

födelsedatum (datum dd en individ föds) 1849-01-22T00:00:00Z

dödsdatum (datum dd en individ oavider) 1912-05-14T00:00:00Z

[Learn more](#)

Figure 3.6: The full search result

Chapter 4

Evaluation

We conducted a qualitative evaluation of the system. We defined a search experiment and asked users to give their feedback on the system.

4.1 Evaluation strategy

To evaluate the application, we presented it to five users of various background and let each user perform a test of the application. The test has not been predetermined as to what they should search for or in what order to look at the different components. We have been available during the whole test to answer questions that have arisen. After the test, we asked the participants to fill in a form about their experience.

4.2 Result

In this section we present the result of the evaluation of the application. The results are in Swedish with translations. The questions asked are:

- Jag förstår vad applikationen skall vara bra för
 - Kommentarer om användning
- Jag tycker applikationen ser bra ut
 - Kommentarer om utseende
- Jag tycker applikationen verkar vara användbar för mig
 - Kommentarer om användbarhet

- Jag tycker konceptsökningen på sidan var användbar
 - Kommentarer om konceptsök
- Jag vet vad du menar med konceptsök
- Jag tycker informationsboxen bidrog till upplevelsen
 - Kommentarer om infobox
- Fler kommentarer
- Jag upplevde problem med
 - Lång tid för sökningen
 - Fel resultat när jag sökte
 - Annat ...
- Övriga problem

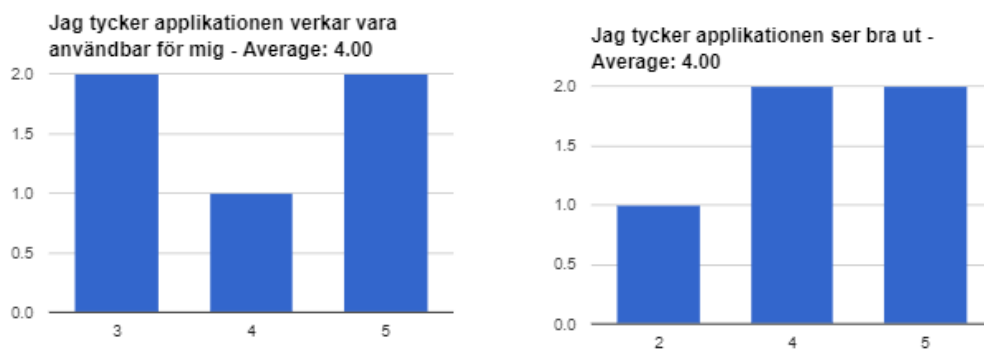


Figure 4.1: Left part: How useful is the application to me; Right part: I think the application looks good



Figure 4.2: Left part: I found the concept search useful; Right part: I know the meaning of concept search



Figure 4.3: Left part: I think the infobox enhance the experience;
Right part: I experienced problems with

In Figures 4.1 to 4.3, the result of the evaluation is presented in a graphical summary. If we look closer at this we see that the lowest score in any question is 2. In Figure 4.1 we can see that the average score for both “how useful the application is” and “how good the application looks” is 4 but that the distribution is different. In the right part of Figure 4.3, we can see that the search time was an issue that most users experienced. We follow up this and some reasoning about this in section 4.3.

Comments In the evaluation form, there was the ability to comment the specifics of the application. The comments made are presented with translation in Table 4.1.

4.3 Discussion

In the result, we can see that there is a general understanding of the search application and how it can be of benefit. There is however some confusion on the concept search. The general opinion is overall positive and the idea to search by concept instead of just a text string have been attractive during discussions after trying the application.

Looks and feels There was a common opinion that the application was clean and not cluttered. There are elements lacking in the application that would have been necessary for a production ready application, such as pagination of results and some indication on number of results.

An issue that was commonly reported was that the search was slow, sometimes up to several minutes. This is most likely caused by how the documents are represented in the database (as the whole document, not split into smaller chunks) in combination with using a filtered search so every hit has to be verified. In the section about improvements 5.1, this thought is followed up on.

Infobox When asked about the information box there were different opinions. Some thought it only partly contributed to the experience, while others wished it displayed much more and really liked it. We noticed that this was to some extent depending on the search terms and concepts explored, as this also regulates the information shown in the box. In

the application, we use an API to guess the correct Qid based on the search term. As this guess is independent of the concepts previously identified in the books, there can be a mismatch. A mismatch results in a very limited infobox as there is no information in the database for that concept. We lack information about these entities as we in application only have captured information for the concepts identified to be present in the books. In retrospect, this is not an optimal solution.

Concept search Regarding the concept search, there was, as stated, some confusion on the function some of the users used. Some users saw the facets as their primary result view. Other users found them to be of limited use. A note is that the facets usually sparked questions and requests for explanation about the idea of concepts. As we gave the explanation on how to use the application verbally, the margin of error of how well the intentions are understood is quite large here.

One obvious problem was the sorting and prioritization of the concepts in the facet box. The Langforia API in combination with the books have a high hit-rate for countries and some other specific concepts. This result in an all too common situation where the concepts listed are very broad and mostly a list of countries. This has to do with how the facet table is limiting to only the facets with a high count. In retrospect this is not the ideal ranking.

Search result In the project, we are aware that to study the quality of the search result, there is a high requirement on domain knowledge. However when we selected the users to test the application we did not have this as a criteria. Some users did anyhow some statements and were able to identify some strange results.

When we investigated these strange results, we concluded that it was due to that the first search is a text search. We made a trade-of so that the search is not limited to identified concepts that an auto-complete to concept could lead to. One solution could be to auto-complete if the concept was known but do a text if not, another alternative could be to give the user a choice of text or concept search.

Table 4.1: Comments

Kommentarer om utsende (About looks)	
Kan förbättras designmässigt samt förenkla sökandet med t.ex. att sökning börjar när användaren trycker enter.	The design could be improved and so could the searchexperience i.e. by starting the search when the user press enter
konceptsök kolumnen till vänster i bilden kändes lite störig pga olika långa fält	The concept column to the left in the picture felt a bit messy due to uneven length on the buttons
Kommentarer om användbarhet (About usability)	
Känns mer företagsinriktat alternativt användning vid djupgående informationssökning.	Feels more targeted at use in corporations alternative in deeper searches for information.
Kommentarer om konceptsök (About concept search)	
det känns som om jag finner svaret på min fråga där	Feels like I find the answer to my question there
Kommentarer om infobox (About infobox)	
Sökningarna tar väl mycket tid, man är van vid snabbare då man söker i andra databaser. Absolut gav det ett fylligare svar, extra info!	one is used to faster searches in other databases. Absolutely it provide a more elaborated answer, extra info!
Fler kommentarer (More comments)	
Mycket önskvärt att kunna komma in i texten kring det man sökt i de olika böckerna. hittade trots stavfel i sökningen, använder nog inte applikationen fullt ut. Fler infoboxar , eller fler flikar på infoboxen att välja på	It is very desirable to be able to get into (read) the texts that appears in the search result. Returned results even thou the search term was misspelled, I probably doesn't use the application fully. More infoboxes of tabs to choose from on the infobox
Övriga problem (Other problems)	
Söksättet ger mersmak men behöver utvecklas.	The search method gives you wanting more, but needs to be developed.

Chapter 5

Conclusions

This project has shown that the users like the idea of a concept-based search. When the users make a search that matches the right conditions the results are improved over string search. However, the users find that the idea of concepts is something new and needs an explanation.

We need to address the nature of entity identification and that there is a balancing act that has to be made between precision and how many “types of entities” that can be identified. This means that for the application to be as relevant as possible, we should ideally use an algorithm that is suited to the content. This could be that the application adapts the current algorithm with some parameters or that the application uses different algorithms for different types of content. If the application uses a relevant algorithm, it delivers the concepts that are both precise and relevant to the users.

The evaluation unearthed some issues in the application but we deemed that we could rectify those in a future development. We discuss this in the following improvements section.

5.1 Improvements

There are several improvements that could be made to the application to make it more production ready. In this section, we take a look at the different parts and what improvements that can be made.

5.1.1 GUI

The graphic interface is a bit rough with some buttons like those of the facets being of unequal length. And although it uses bootstrap¹ as a framework some problems with the

¹<https://getbootstrap.com/>

responsiveness of the design exists.

One suggestion made is that when applying a facet via the buttons the facet should not be concatenated to the search term but instead be presented as buttons after the search box. This way it would be easy to remove a facet from a search.

5.1.2 Facets and concept search

The facets and the concept search are the main part of the whole application so there is obviously quite many improvements that we could make. We find some of them in the evaluation.

Facets should be more related to the actual search term instead of relying on the count of occurrences in the document. It would also be neat if the facets were more related to each other. This way it would be possible to link related concepts in the GUI.

Concepts could be improved in several ways. There is one fundamental limitation on using machine learning algorithms for identifying concepts. This is that if the training for the algorithm is too general with data from every type of subject. It will not be as good as an algorithm trained on more specific subject, this makes a generally trained algorithm prone to produce some errors. However if the training is more specific only the concepts within the domain are identified. To solve this, it would be possible to be more specific in what documents are loaded and to combine this with training the algorithm with regard to the type of documents, but this requires more control of the Langforia API than was available in this project.

5.1.3 Search results

In the discussion part, we noticed that we could improve how easy and understandable the concept search is. We can improve multiple things. For example, we could improve the instructions on the idea of concepts. We could also hide some of the details i.e. displaying them as buttons with the prefix and name instead of the IRI.

One improvement to the search result would be the possibility to go deeper into the results. When we display the results, we should ideally show the section where the hit is and combine this with relevant information to the text. The application should display related sections so that text can be connected via concepts. Much of the needed information is already available in the current application database.

5.1.4 Concept tagging and information gathering

The enrichment process in the current application is triggered manually so the concepts in the application do not benefit of any updates to the API or wikidata. This should in a production version be automatic. As the RDF data used in the infobox is collected at the same time as the concept tagging the data is not updated with any regularity.

Infobox Regarding the infobox, we have thoughts about some ways to improve this. One might be to capture the data directly from the semantic web in every request. Another solution might be to increase the amount of data we capture and to capture data for concepts related to those identified in the book. Finally, we thought about a third solution that is a combination of the two previous. In this alternative, we would still use the database but whenever the application gets a request for a Qid not in the database, it makes a request to Wikidata about the missing information.

5.1.5 Speed

One frequent comment that came up during evaluation was that the search is slow. This has to do with several factors like how the snippets are calculated; how the documents are represented in the database; and how the indices are configured. To improve the speed documents could be split into smaller parts. This however has some drawbacks as *and searches* will only match the smaller parts. This will give more precise but not as many search matches something that could potentially be a very good thing if the division is done in a good way.

The indices can also be used in a much better way as the current application uses few indices and filters for the search. This means that the database has to read the documents from the hard drive and not only rely on in memory indices. If the indices were configured more optimally it would be possible to do an unfiltered search returning results without touching the disk.

The snippets currently are calculated with the built in function. This is a good but not the fastest. A simpler snippeting function could be faster.

5.2 Summary

We have explored the viability of a concept-based search and we can conclude that there is a demand and a possibility for it. We have seen that there is a great potential in the information available at the semantic web. We also see that if we would improve our application with this data, we would be able to create an application that really helped in the discovery of information. However, we also discovered the non-triviality of the problem. It is a challenge to create a search that is both fast and precise. In addition, we have as discussed earlier seen that the search needs to be adapted to the content so it will never meet the requirements of a general search.

Glossary

DTD Document Type Definition. 10, 35

EDL Entity Discovery and Linking. 7

IRI Internationalized Resource Identifier. 12, 14, 21, 22, 32

MUC Message Understanding Conference. 9

NERC Named Entity Recognition and Classification. 9

NLP Natural-language processing. 7, 18

OKG Open Knowledge Graph. 14

OWL Web Ontology Language. 10

QId Identifier for entities on Wikidata. 18, 21

RDF Resource Description Framework. 10–14, 19, 32

RDFa Resource Description Framework in Attributes. 5, 19, 20

RDFS Resource Description Framework Schema. 10

RELAX NG RELAX NG (REgular LAnguage for XML Next Generation) is a schema language for XML. RELAX NG is aimed to be both more powerful and easier to write than a DTD. 10

SPARQL SPARQL is a query language for RDF data. With SPARQL it is possible to query many different types of data sources.. 19, 22

TAC Text Analysis Conference. 7

XML Extensible Markup Language. 10, 17

Bibliography

- Allemang, D. and Hendler, J. (2011). *Semantic Web for the Working Ontologist, Second Edition: Effective Modeling in RDFS and OWL*. Morgan Kaufmann.
- Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*.
- Cucerzan, S. (2011). TAC entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of the Text Analysis Conference (TAC)*, Gaithersburg, Maryland.
- DocBookTC, O. (2009). Docbook-5.0 the docbook schema version 5.0. edited by norman walsh. 1 november 2009. Technical report, OASIS DocBook TC.
- Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Klang, M. and Nugues, P. (2016). Langforia: Language pipelines for annotating large collections of documents. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 74–78, Osaka, Japan. The COLING 2016 Organizing Committee.
- Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticæ Investigationes*, (30):3–26.
- Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*.
- Sil, A., Dinu, G., Kundu, G., and Florian, R. (2017). The IBM systems for entity discovery and linking at TAC 2017. In *Proceedings of the Text Analysis Conference (TAC)*, Gaithersburg, Maryland.

The DBpedia Association (2018). What is DBpedia. <http://blog.dbpedia.org/sample-page/>. Accessed: 2018-03-01.

Walsh, N. (2010). *DocBook 5: The definitive guide*. O'Reilly Media. Docbook 5.0.

wmflabs (2018). Wikidata statistic. <https://tools.wmflabs.org/wikidata-todo/stats.php>. Accessed: 2018-02-28.

EXAMENSARBETE Entity-based Search

A take on the intelligent book

STUDENT Erik Zander**HANDLEDARE** Pierre Nugues (LTH), Markus Klang (LTH)**EXAMINATOR** Jacek Malec (LTH)

Konceptbaserad sök

POPULÄRVETENSKAPLIG SAMMANFATTNING Erik Zander

Tänk dig att vi ska söka upp *Paris*, svaret kommer direkt - stad i Frankrike. Dock är det personen *Paris* vi söker. Vi prövar igen, *Paris personen*, vi får nu träff på den mytologiska personen *Paris* men vi sökte den fiktiva personen *Tom Paris* från Star Trek. Det är uppenbart att det är svårt att veta vad vi söker efter, för datorer har det varit omöjligt. Omöjligt är dock ett relativt begrepp när det gäller teknik. Detta projekt har jobbat med att ta aktuella forskningsresultat om språkteknik och skapa en sökmotorprototyp med hjälp av dessa.

Vår prototyp är en sökmotor där en användare kan söka bland 3500 böcker. Det vanliga fallet är att en sökning startar med en textsträng, sökningen letar bara efter att bokstäverna skall vara rätt. När sen resultatet presenteras, visas de koncept som sökningen har hittat i anslutning till sökordet. Ett koncept kan vara en person, ett land, en stad, osv. Om vi återgår till vår sökning om *Paris* kan till exempel sökmotorn presentera *Paris (staden)*, *Paris (grekisk mytologi)*, *Paris (Sci-fi karaktär)*. Vi ser att om vi får dessa alternativ presenterade kan vi som användare snabbare komma in på det vi letade efter. Det vi inte är intresserade av filtrerar vi bort.

Hur går då detta till? För att svara på det ger vi först en överblick vad som krävs för att det skall vara möjligt. Vi behöver veta vilka koncept en bok innehåller; vi behöver spara den informationen någonstans; vi vill kunna länka ut till online resurser som Wikipedia, alternativt till grafvarianter av Wikipedia så som Wikidata eller DBpedia.

För att veta vad en bok innehåller tar vi hjälp av ett API (Langforia) som är skrivet av Markus Klang på LTH. Till API:et skickar vi varje para-

graf i boken och får tillbaka vilka koncept den inne-håller. Vi sparar denna information i samma databas som vi har böckerna i. Mer precist sparar vi informationen i varje digital bok genom att vi berikar paragrafen.

När vi får informationen från Langforia får vi även ett så kallat Qid. Ett Qid är en identifierare på Wikidata.org och med den kan vi hitta mer information om ett specifikt koncept. Den kompletterande informationen kan till exempel vara när en person dog eller hur stor befolkning ett land har. I vår applikation visar vi denna information i en informationsruta till höger.

Vad skall det vara bra för? Genom att veta vilka koncept som texterna innehåller kan vi både ge en precisare och mer berikad upplevelse. Det krävs dock att applikationen arbetar med texterna innan och begränsningar i hur det fungerar med maskininlärning gör att vi inte kan jobba med alla typer av texter samtidigt. Bäst blir det därför om vi har ett material som vi vet ungefär vad det är men som vi behöver kunna söka bättre i. Har vi ett sådant material kan vi erbjuda en mycket bra upplevelse till användaren.