Deliverable 3.2: Technologies for MOVING data processing and visualisation v2.0

Iacopo Vagliano, Mohammad Abdel-Qader, Till Blume, Falk Böschen, Lukas Galke, Ahmed Saleh, Ansgar Scherp/ZBW
Vasileios Mezaris, Alexandros Pournaras, Christos Tzelepis/CERTH
Ilija Šimić, Cecilia di Sciascio, Vedran Sabol/KC
Aitor Apaolaza, Markel Vigo/UMAN
Tobias Backes, Peter Mutschke/GESIS

30/03/2018

Work Package 3: Data processing and data visualisation technology

TraininG towards a society of data-saVvy inforMation prOfessionals to enable open leadership INnovation

Horizon 2020 - INSO-4-2015

Research and Innovation Programme

Grant Agreement Number 693092

Dissemination level	PU
Contractual date of delivery	31/03/2018
Actual date of delivery	30/03/2018
Deliverable number	3.2
Deliverable name	Technologies for MOVING data processing and visualisation v2.0
File	`MOVING_D3.2_v1.0.tex`
Nature	Report
Status & version	Final & v1.0
Number of pages	113
WP contributing to the deliverable	3
Task responsible	ZBW
Other contributors	CERTH, KC, UMAN, GESIS
Author(s)	Iacopo Vagliano, Mohammad Abdel-Qader, Till Blume, Falk Böschen, Lukas Galke, Ahmed Saleh, Ansgar Scherp/ZBW Vasileios Mezaris, Alexandros Pournaras, Christos Tzelepis/CERTH Ilija Šimić, Cecilia di Sciascio, Vedran Sabol/KC Aitor Apaolaza, Markel Vigo/UMAN Tobias Backes, Peter Mutschke/GESIS
Quality Assessors	Franziska Günther
EC Project Officer	Hinano SPREAFICO
Keywords	Technologies, data acquisition, data processing, data visualisation, user data logging, common data model

Executive summary

This deliverable *D3.2 Technologies for MOVING data processing and visualisation v2.0* provides an update on the common data model as well as on the set of data acquisition, data processing, user logging and data visualisation components. The common data model has been modified to adapt to the new requirements and to better represent the document hosted by the MOVING platform (Section 2). Additionally a new service to validate and integrate data based on the common data model has been developed. Advances in data acquisition and data processing is described in Section 3 and comprises different techniques. This includes the improved version of three different crawlers for webpages and social media content (Section 3.1), a flexible schema-level index for Linked Open Data to harvest additional metadata (Section 3.2), an analysis on the evolution of vocabulary in knowledge graphs (Section 3.3) to investigate how these changes may affect the data harvesting, the disambiguation of string representations of authors (Section 3.4), duplicate detection (Section 3.5), the comparison of various information retrieval model that use only titles with traditional full-text based techniques (Section 3.6), the use of machine learning for multi-label classification relying only on titles (Section 3.7), improved techniques for video fragmentation, concept detection and video transcript analysis (Section 3.8) and a prototype of a scientific search engines which exploits not only the text but also the figures (Section 3.9). The changes in logging of user interaction data as well as the new features of the dashboard to test hypotheses against the logging data is described in Section 4. A set of new visualisations as well as their functional prototypes are described in Section 5. Finally, we summarise the main contribution for each task highlighting the achievements of year 2 in Section 6.

Table of contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Explanation
AD	Author Disambiguation
ADMS	Asset Description Metadata Schema
AP	Average Precision
API	Application Programming Interface
BFS	Best Feature Subset
BM25	Best Matching 25
BM25C	Best Matching 25 using Concept frequencies
BM25CT	Best Matching 25 using both Concept and Term frequencies
BTC	Billion Triple Challange
CF	Concept Frequency
CF-IDF	Concept Frequency - Inverse Document Frequency
CFS	Correlation-based Feature Selection
CiTO	Citation Typing Ontology
CQ	Complex Query
CSE	Complex Schema Element
CTF	Concept frequency combined with Term Frequency
C-DSSM	Convolutional Deep Structured Semantic Model
DCAT	Data Catalog Vocabulary
DCNN	Deep Convolutional Neural Network
DCG	Discounted Cumulative Gain
DIS	Data Integration Service
DSSM	Deep Structured Semantic Model
DyLDO	Dynamic Linked Data Observatory
EQR	Equivalence Relation
ESA	Explicit Semantic Analysis
FDC	Focused web-Domain Crawler
FIV	Fachinformationsverbund Internationale Beziehungen und Länderkunde
FLuID	Formal schema-Level Index model for the web of Data
FOAF	Friend Of A Friend
geom	Ontology for geometry
GN	Geonames Ontology
GSU	Gaussian Sample Uncertainty
GVF	Graph Visualisation Framework
HCF-IDF	Hierarchical Concept Frequency - Inverse Document Frequency
HOF	Histogram Of Flow
HOG	Histogram Of Gradient
HTML	HyperText Markup Language
ID	Identifier
IDCG	Ideal Discounted Cumulative Gain
IDF	Inverse Document Frequency
iff	if and only if
IR	Information Retrieval
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KG	Knowledge Graph
kNN	k-Nearest Neighbors
L2R	Learning to Rank
LOD	Linked Open Data
LOV	Linked Open Vocabularies
LR	Logistic Regression
LSVM	Linear Support Vector Machine
LSVM-GSU	Linear Support Vector Machine with Gaussian Sample Uncertainty
LSVM-iso	Linear Support Vector Machine for handling isotropic uncertainty
MAP	Mean Average Precision

Abbreviation	Explanation
MART	Multiple Additive Regression Trees
MeSH	Medical Subject Headings
MK	Metzler and Kanungo's features
MLP	Multi-Layer Perceptron
mo	Music Ontology
nDCG	Normalised Discounted Cumulative Gain
NER	Named Entity Recognizer
NLP	Natural Language Processing
oa	Open Annotation data model
OC	Object Cluster
OEM	Object Exchange Model
PAY	Paylod function
PC	Property Cluster
PDF	Portable Document Format
PLD	Pay Level Domain
POC	Property Object Cluster
POS	Part of Speech
PROV-O	PROVenance Ontology
PSVM	Power Support Vector Machine
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
rID	researcher´s ID
SD	Standard Deviation
SEC	Search-Engine-based web Crawler
SG	Schema Graph
SGD	Stochastic Gradient Descendant
SIFT	Scale-Invariant Feature Transform
SKOS	Simple Knowledge Organization System
SLI	Schema Level Index
SPARQL	SPARQL Protocol and RDF Query Language
SQ	Simple Query
SRT	transcripts
SSE	Simple Schema Element
SSM	Social Stream Manager
SSOAR	Social Science Open Access Repository
STW	Standard-Thesaurus Wirtschaft
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
SVM-GSU	Support Vector Machine with Gaussian Sample Uncertainty
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VIA	Video Analysis Service
voaf	Vocabulary Of A Friend
WebGL	Web Graphics Library
WevQuery	Web Event Query Tool
WoS	Web of Science
xkos	SKOS for representation of nomenclatures

1 Introduction

1.1 History of the document

Table 2: Document history.

Date	Version
19/01/2018	v0.1: first draft of the table of content
29/01/2018	v0.2: table of content ready for QA
31/01/2018	v0.3: table of content´s comments addressed
09/03/2018	v0.4: content ready for QA
31/03/2018	v1.0: final document

1.2 Purpose of the document

This document provides an update on the technological components developed for the MOVING platform with respect to what described into the previous deliverable D3.1: *Technologies for MOVING data processing and visualisation v1.0* (Blume et al., 2017). All technologies described in this document can be integrated in the final MOVING platform, but not necessarily have to. The upcoming D3.3: *Technologies for MOVING data processing and visualisation v3.0*, due in month 34 will further extend on this document to collect a final set of technologies well suited for the MOVING platform and its functional requirements elicited in D1.1: *User requirements and specification of the use cases* (Bienia et al., 2017).

1.3 Structure of the document

This document is structured into four main sections describing the different technologies. In Section 2, we describe the new version of the common data model. The data acquisition and the data processing components are described in Section 3, while the user logging components in Section 4. Finally, the data visualisation components are described in Section 5. An overview of the interaction between components developed in WP 3 as well as their interconnection to the MOVING web application developed in WP 4 is illustrated in Figure 1.

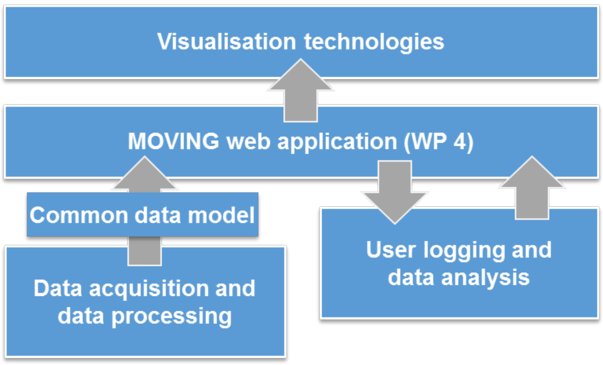

Visualisation technologies
MOVING web application (WP 4)
Common data model
Data acquisition and data processing
User logging and data analysis

Figure 1: Overview of the interaction between components developed in WP 3 and the MOVING web application.

2 Common data model

The MOVING platform provides access to a large variety of documents coming from different data sources. The common data model for MOVING lays the foundation for data integration in the project. The specific challenge is the integration of the variety of data sources like video lectures, publications, and metadata from professional publishers. After reviewing the user requirements coming from deliverable D1.1 (Bienia et al., 2017), we needed to revise the core attributes we identified in deliverable D3.1 (Blume et al., 2017). Therefore, we defined a new common data model v1.1. This update to the common data model was also used to introduce a number of fields to enhance further data processing.

The remainder of this section is organised as follows. In Section 2.1, we describe how we conceptually change the common data model to fit the new requirements. In Section 2.2, we review the user requirements coming from deliverable D1.1 (Bienia et al., 2017) with regard to the common data model. Finally, in Section 2.3 we briefly describe the Data Integration Service (DIS), which ensures that data introduced in the platform is compliant with the common data model.

2.1 Improved document representation

As previously outlined, modifications on the common data model were necessary. The new version of the common data model has three major characteristics:

1. separation of original metadata and generated metadata;
2. provision of core fields for all types of documents and additional specific fields for certain types of documents;
3. backwards compatibility with version v1.0.

The core of the common data model v1.1 is detailed in Listing 3 and Figure 58 in Appendix 7. In the new common data model, there is a trade-off between re-using fields as much as possible and semantic separation of content. Furthermore, we want to ensure a minimum quality level on the data. For example, storing every person in the same field may seem appealing to avoid redundancy and ease the disambiguation task, however, automatically extracted persons from the text have a higher probability of being of lower quality in contrast to manually created metadata from domain experts.

In Section 2.2, we provide details on the common data model's core. In addition to the core, some specific document types need highly specialised fields. In order to keep the core clean, we introduce single attributes to the document, which serve as a container for other attributes. These attributes, e. g., "lawSpecific_metadata", "videoSpecific_metadata", "fundingSpecific_metadata", or "temporal_metadata", contain the specialised fields and can be changed without affecting the core.

2.2 Updated requirements and changes in the model

Deliverable D1.1 provides the initial requirement analysis based on the project objectives, the inputs collected from the project partners, as well as the user groups targeted by the project and the specifications of the platform according to the documented requirements (Bienia et al., 2017). We reviewed each requirement with regard to the common data model. This means, we specifically investigated what data needs to be present in order to full-fill a requirement. Preparing the data model to store this data is a first, mandatory step to implement the required feature. For example, requirement *#TUD32: Listing of all databases* requires storing in the index the database from which the document is retrieved. Thus, we include *document.source* as core attribute which allows to implement the proper functionality in the front-end. However, just updating the common data model does not guarantee the satisfaction of the requirement. Typically further actions need to be taken.

An overview of core attributes in the common data model v1.1, a short description of the content, and a pointer to the corresponding user requirement that requires the existence of this attribute is given in Table 3. We highlighted the newly introduced fields. Please note, some of the changes are not directly motivated by a user requirement but rather from a data processing component. For most of the changes, we provide a supplementary explanation to the description in Table 3.

The most drastic change compared to the previous version is the strict separation of provided metadata and generated metadata. In principle, we are prepared to store persons, organisations, locations, and venues inside a single document. For example, we conceptually distinguish metadata related to persons (e.g. authors) and (extracted) entities of type person. Furthermore, each metadata person, metadata organisations, and metadata location can have a specific role inside this document, e. g., being the author or the editor. Furthermore, we now distinguish between source URLs and document URLs. Source URLs can contain machine readable information and allow a fine grained handling regarding data quality in contrast to the whole data source. With document URLs, we can expect a link which safely redirect users to a human-readable resource. Another important change is the separation of concepts, sectors, and subjects. Although their data structure is the same, the semantic meaning differs. Thus, we decided to store each in a separate field. Please see Table 3 for a detailed list of all changes and the full status of the new common data model.

Table 3: Overview of core attributes in the common data model v1.1, a short description of the content, and a pointer to the corresponding user requirement that implies the existence of this attribute. The highlighted rows indicate changes compared to the previous version of the common data model.

Attribute	Description	Requirement
document.identifier	Unique identifier for each document (same identifier for documents representing the same real world document)	#TUD053
document.sourceURLs	A valid URI pointing to the original source record, e.g., Springer link	#EY004
document.documentURLs	A valid URL pointing to the available fulltext/YouTube/...	#TUD071, #TUD128, #TUD102
document.title	The title of the document	#TUD039, #EY008
document.abstract	The abstract or a short description of the document	#TUD039, #EY008
document.fulltext	If available, store the full-text for data analysis. For human readable format, we have the link to the full-text hosted at an external provider (document.documentURLs)	#TUD039, #EY008, #TUD102
document.thumbnailURL	A link to a thumbnail if provided to enhance the visual experience when browsing the result list	
document.isPartOf	A document may be part of a parent document, e. g., a video in playlist or paragraph inside a law. We store the document.identifier as well as the position within the parent document	
document.hasParts	A document may have multiple child documents. We store the document.identifier as well as the position within the parent document	
document.metadata_persons	List of valid people appearing in the metadata, e.g. author. In the previous version, this was the author field *(see metadata_person below)*	#TUD040, #EY037, #EY016
document.metadata_organisations	List of valid organisations appearing in the metadata, e.g. authoring organisation *(see metadata_organisation below)*	#EY037, #EY016, #TUD040
document.metadata_locations	List of valid locations appearing in the metadata, e.g. where a specific law is applicable *(see metadata_location below)*	#EY009, #EY016
document.metadata_venue	*(see venue below)*	
document.startDate	ISO 8601	#TUD038, #EY036, #EY042
document.endDate	ISO 8601	#TUD038, #EY036, #EY042
document.source	The name of the data source (predefined list)	#TUD011, #TUD012, #TUD016, #TUD089, #EY007, #EY003

Table 3: Overview of core attributes in the common data model v1.1, a short description of the content, and a pointer to the corresponding user requirement that implies the existence of this attribute. The highlighted rows indicate changes compared to the previous version of the common data model.

Attribute	Description	Requirement
document.license	Name of the license or license description	#TUD113, #TUD112, #TUD118
document.openAccess	1 = open access, 0 = not	#TUD113, #TUD112, #TUD118
document.docType	The document type (predefined list)	#TUD035 #TUD090
document.language	ISO 639-1 language code	#EY012
document.concepts	*(see concept below)*	#TUD006, #TUD015, #EY043
document.sectors	Stores the industry types as sectors *(See sectors below)*	#EY049
document.subjects	Use to model the subject area/discipline of a document *(See subject below)*	#TUD041, #TUD050, #EY010
document.keywords	Simple keywords with no particular connection to a thesaurus	#TUD006, #TUD015
document.references	*(see reference below)*	#TUD042, #TUD051
document.entities	Extracted entities, which in contrast to metadata entities may be of lower quality *(see entity below)*	#EY016
document.external_statistics	Include externally generated statistics, e. g., click data or number of citations *(see external_statistic below)*	#TUD042, #TUD051, #TUD046
document.searchDomains	We differentiate between, e. g., research data, learning resources, project data. Crucial for the integration of working and training environment	
metadata_person.identifier	Unique identifier for each person (same identifier for persons representing the same real world person)	#TUD053 #TUD124
metadata_person.mentionID	MOVING unique ID of this appearance within a document *(needed for disambiguation)*	
metadata_person.URIs	URI pointing to an external representation/additional information	
metadata_person.name	Processed/Normalised name, e.g., "Name1 Name2, J. R. R." *(needed for disambiguation)*	
metadata_person.rawName	Name as displayed in source	
metadata_person.roles	List of possible roles in a predefined list	#EY037, #TUD048
metadata_person.email	A valid email address	#TUD125, #TUD126
metadata_person.affiliations	*(see organisation below)*	#TUD125, #TUD126
metadata_organisation.identifier	Unique identifier for each organisation (same identifier for organisations representing the same real world organisation)	#TUD053
metadata_organisation.mentionID	MOVING unique ID of this appearance within a document *(needed for disambiguation)*	
metadata_organisation.URIs	URI pointing to an external representation/additional information	
metadata_organisation.name	Processed/Normalised name *(needed for disambiguation)*	
metadata_organisation.rawName	Name as displayed in source	
metadata_organisation.location	*(see location below)*	#EY009

Table 3: Overview of core attributes in the common data model v1.1, a short description of the content, and a pointer to the corresponding user requirement that implies the existence of this attribute. The highlighted rows indicate changes compared to the previous version of the common data model.

Attribute	Description	Requirement
metadata_organisation.roles	List of possible organisation roles in a predefined list	#EY037, #TUD048
metadata_location.identifier	Unique identifier for each location (same identifier for locations representing the same real world location)	#TUD053
metadata_location.mentionID	MOVING unique ID of this appearance within a document *(needed for disambiguation)*	
metadata_location.URIs	URI pointing to an external representation/additional information	
metadata_location.name	Processed/Normalised name *(needed for disambiguation)*	#EY009
metadata_location.rawName	Name as displayed in source	
location.lat	Latitude with '.'	#EY009
location.lon	Longitude with '.'	#EY009
metadata_location.roles	List of possible location roles in a predefined list	#EY037, #TUD048
metadata_venue.identifier	Unique identifier for each venue (same identifier for venues representing the same real world venue)	#TUD053
metadata_venue.mentionID	MOVING unique ID of this appearance within a document *(needed for disambiguation)*	
metadata_venue.URIs	URI pointing to an external representation/additional information	
metadata_venue.name	Processed/Normalised name *(needed for disambiguation)*	#EY009
metadata_venue.rawName	Name as displayed in source	
metadata_venue.startDate	ISO 8601	
metadata_venue.endDate	ISO 8601	
metadata_venue.volume	Valid integer	
metadata_venue.issue	Valid integer	
metadata_venue.location	*(see location above)*	#EY009
concept.label	A human readable label explaining this concept	
concept.URL	A link to an external thesaurus	
concept.relevanceScore	Value between 0 and 1 determining the relevance for this particular document	#EY043
sector.label	A human readable label explaining this sector	
sector.URL	A link to an external thesaurus	
sector.relevanceScore	Value between 0 and 1 determining the relevance for this particular document	
subject.label	A human readable label explaining this subject	
subject.URL	A link to an external thesaurus	
subject.relevanceScore	Value between 0 and 1 determining the relevance for this particular document	
reference.identifier	Holds a document.identifier if document is in the MOVING database	
reference.rawText	The extracted text of this reference	
entity.identifier	Unique identifier (linking entities and metadata representing the same real world object)	
entity.mentionID	MOVING unique ID of this appearance within a document *(needed for disambiguation)*	
entity.URIs	URI pointing to an external representation/additional information	
entity.label	The text fragment extracted from the text representing the entity	

Table 3: Overview of core attributes in the common data model v1.1, a short description of the content, and a pointer to the corresponding user requirement that implies the existence of this attribute. The highlighted rows indicate changes compared to the previous version of the common data model.

Attribute	Description	Requirement
entity.confidence	Confidence score between 0 and 1 to indicate to which level of trust this entity was extracted correctly	
entity.type	A predefined list of possible entities we extract, e. g., person, organisation, location	
external_statistic.label	Name of this statistic, e. g., "Number of Clicks"	
external_statistic.value	The actual value, e. g., 300	

2.3 Validation and integration in the MOVING platform

All data indexed by the search engine[1] has to be converted into the common data model beforehand. We use the JSON format and name the fields as presented in the Appendix 7 (Listing 3). For certain fields some normalisation patterns need to be applied in order to ensure a smooth processing. For specific fields like `source` and `docType`, a list of all valid values has to be maintained to avoid ambiguity.

For internal uses, we store the document type with the highest level of detail possible. However, we do not always show these details to the user. We keep track of document types and how they have to be presented to the user. We also pre-defined, which type of document qualifies as research, learning, or funding material. All types of documents are stored in the search engine based on the common data model and are subject to validity and minimum quality checks before being indexed.

These quality checks are performed by the Data Integration Service (DIS). The DIS can be manually started to index one or more JSON files containing one or more objects. Each JSON object is validated according to the latest version of the `common-data-model.jsonschema` document. Valid JSON objects obtain a default ID, mapped to a search domain according to the provided `docType`, and are indexed by search engine to be retrieved. Each invalid JSON object is rejected and a detailed error report is provided in a log file. The service is also running in background waiting for HTTP POST requests containing the data of a single JSON object. The architecture of the DIS is depicted in Fig. 2.

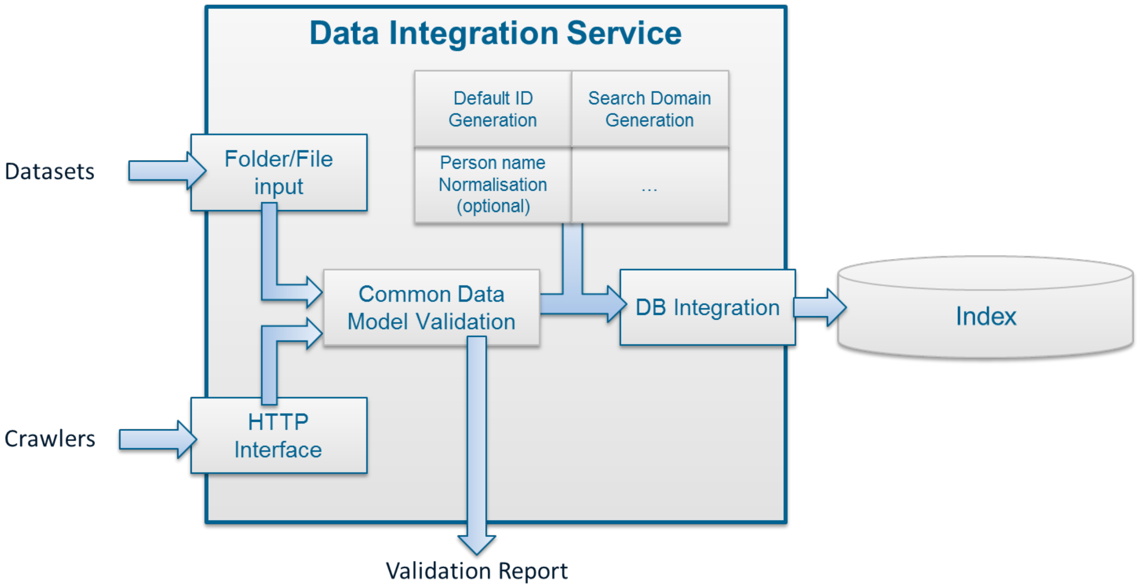


Figure 2: Basic functionality of the Data Integration Service (DIS).

[1] See deliverable D4.1: *Definition of platform architecture and software development configuration* (Gottfried, Grunewald, et al., 2017) for further details on the search engine.

3 Data acquisition and data processing

3.1 Crawling of social media, websites and videos

3.1.1 Problem statement

The internet has become the greatest source of information and is continuously expanding. The emergence of social media the last decade has even increased the volume of data available making its effective retrieval a challenging task. The typical way of accessing information on the internet is by employing search engines. Search engines rely on web crawlers to collect their data. Web crawlers are internet bots that systematically browse the web while adhering to some rules.

For the purpose of MOVING, we narrow down crawling by targeting user-defined topics and websites that serve the demands of the public administrators and young researchers who will engage with the platform. There are three separate crawlers, each focuses on different data sources: (1) the Social Stream Manager (SSM) targets social media, (2) the Search-Engine based Crawler (SEC) deals with web search engines (notably Google), and (3) the Focused web-Domain Crawler (FDC) handles specific websites.

3.1.2 Method description

The overall Crawler architecture is illustrated in Figure 3. The three crawlers communicate with the Data Integration Service that handles the document indexing as well as with the Video Analysis service that handles the video processing/annotation. MongoDB database is also used for storing the input (topics, domains) and for temporal data storage for the SSM.

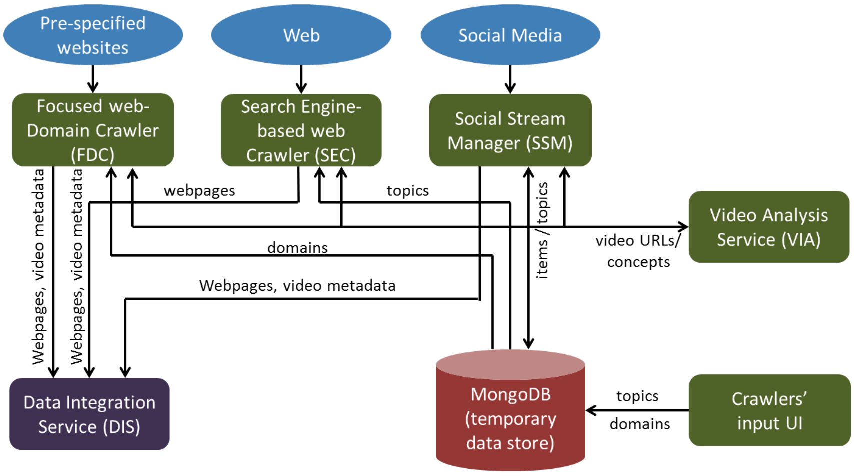
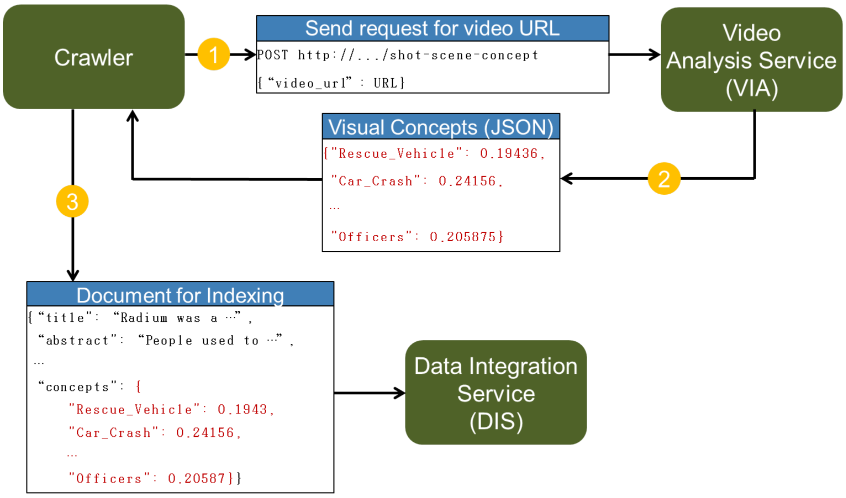


Figure 3: MOVING Crawler architecture.

Social media crawling - Social Stream Manager (SSM) As a core component for the SSM, we employ the stream manager, an open source software developed for the FP7 project Social Sensor[2] that can monitor several social media platforms and is described in detail in D3.1. The SSM functions as a wrapper for the stream manager, processing the data it collects in the MongoDB database. There are three collections of data processed by the SSM. The "WebPages", already outlined in D3.1, are webpage links extracted from the social media posts collected. "MediaItems" contain the multimedia links extracted from the social media posts and "Items" are all the social media posts collected. The HTML content from the WebPage collection is retrieved and field extraction is performed on it. Video processing is performed on "MediaItems" with video content. To accomplish this, we utilise the external Video Analysis Service (VIA), described comprehensively in D3.1. The sequence of events is illustrated in Figure 4. The crawler issues a processing request with the video URL and the service returns the visual concepts with the highest probability scores for the video along with those scores (1). The fields of the collections' metadata together with the extracted ones (concepts for videos, fulltext for HTML) form a document that complies with the common data model (2). The document is then sent to the Data Integration Service (DIS) which handles the indexing (3).

[2] http://socialsensor.eu/


Crawler
1
Send request for video URL
Video Analysis Service (VIA)
Visual Concepts (JSON)
2
3
Document for Indexing
Data Integration Service (DIS)

```
POST http://.../shot-scene-concept
{ “video_url” : URL}
```

```
{"Rescue_Vehicle": 0.19436,
 "Car_Crash": 0.24156,
 …
 "Officers": 0.205875}
```

```
{ “title”: “Radium was a …” ,
 “abstract”: “People used to …” ,
 …
 “concepts”: {
     "Rescue_Vehicle": 0.1943,
     "Car_Crash": 0.24156,
     …
     "Officers": 0.20587}}
```

Figure 4: Crawler communication with VIA.

Web search engine crawling - Search-Engine based Crawler (SEC) The Search-Engine based Crawler exploits the Google search engine API[3] to retrieve topic-related webpages as described in detail in D3.1. SEC has been updated to extract and handle embedded videos included in the crawled webpages. To achieve this we have adopted the open-source tool youtube-dl[4]. Youtube-dl can extract information about embedded videos in a webpage and download those videos. SEC uses youtube-dl only to extract information about embedded videos that potentially exist in a webpage. Youtube-dl is capable of extracting information such as title, thumbnail URL, description, tags, information on the user who uploaded the video, and of course the external URL link. SEC then issues REST requests to the VIA service for each one of the videos (if any) included in the page and retrieves the visual concept results. The fields mentioned above form the document that is sent to the DIS similarly to the SSM.

Crawling of specific domains - Focused web-Domain Crawler (FDC) The Focused web-Domain Crawler is responsible for the crawling of the user-defined web domains. FDC is based on the python framework Scrapy[5] and its basic function is outlined in D3.1. FDC has been updated to include "allow" and "block" patterns defined by the user. An "allow" pattern will force the crawler to fetch a URL if it matches this pattern. Negatively a "block" pattern will prevent the crawler from fetching URLs matching this pattern while allowing any other URL in the domain. Therefore the crawler can focus on specific paths of a website, thus conduct a more efficient crawling process. FDC has also inherited the SEC's functionality to handle video which was previously described.

3.1.3 Crawled data statistics

The use case partners' input was focused on the FDC. The quantitative results of the domains' crawling process is presented in Table 4. The crawling took place from November 2017 to February 2018 resulting in a total of 1,231,722 crawled pages. The domains are mostly websites with educational material and big corporations. The video crawling feature was not introduced at the time, so there are no videos crawled by the FDC. Nevertheless there are videos crawled by the SSM. For a test run in the same period the SSM collected 2214 documents, including 558 videos, for the topic "machine learning". Listing 1 shows an indexed document representing a crawled video. The "concepts" element contains the top visual concepts retrieved by the VIA for the video illustrated in Figure 5.

Listing 1: A document representing a video.

```
{
"_index" : "moving",
"_type" : "publication",
"_id" : "AWBbsaPyj1D7-w21s4o2",
"_score" : 9.573014,
```

[3] https://developers.google.com/custom-search/json-api/v1/overview
[4] https://github.com/rg3/youtube-dl
[5] https://scrapy.org/

```
"_source" : {
"identifier" : "AWBbsaPyj1D7-w21s4o2",
"sourceURLs" : [ "https://www.youtube.com/embed/_rdINNHLYaQ" ],
"title" : "Machine Learning: Solving Problems Big, Small, and Prickly",
"abstract" : "From helping farmers in Japan sort cucumbers to helping doctors in India diagnose eye disease, machine learning is changing the way people -- inside and outside of Google -- use code to solve problems and improve lives. \n\n\nGoogle engineers and researchers (in order of appearance): Maya Gupta, Jeff Dean, Jay Yagnik, Francoise Beaufays, John Giannandrea, Fernando Pereira, Dana Movshovitz-Attias, Rajat Monga, Lily Peng\n\n\nMore on Machine Learning at Google :\nBlog: http://research.google.com/pubs/ArtificialIntelligenceandMachineLearning.html\nG+ page: https://plus.google.com/+ResearchatGoogle\nTwitter: https://twitter.com/googleresearch\nMachine Learning and Deep Neural Nets Explained: https://www.youtube.com/watch?v=bHvf7Tagt18",
"thumbnailURL" : "https://i.ytimg.com/vi/_rdINNHLYaQ/hqdefault.jpg",
"startDate" : "2016-12-12",
"endDate" : "2016-12-12",
"source" : "SocialMediaWeb",
"docType" : "video",
"concepts" : [ {
"label" : "Vehicle",
"relevanceScore" : 0.6972194
}, {
"label" : "Computer_Screen",
"relevanceScore" : 0.7341284
}, {
"label" : "Landscape",
"relevanceScore" : 0.7667853
}, {
"label" : "Outdoor",
"relevanceScore" : 0.8165064
}, {
"label" : "Vegetation",
"relevanceScore" : 0.7103719
}, {
"label" : "Person",
"relevanceScore" : 0.7460899
}, {
"label" : "Indoor",
"relevanceScore" : 0.7064794
}, {
"label" : "Man_Made_Thing",
"relevanceScore" : 0.7280624
}, {
"label" : "Waterscape_Waterfront",
"relevanceScore" : 0.7501564
}, {
"label" : "Text",
"relevanceScore" : 0.7386587
} ],
"searchDomain" : [ "research" ]
}
}
```

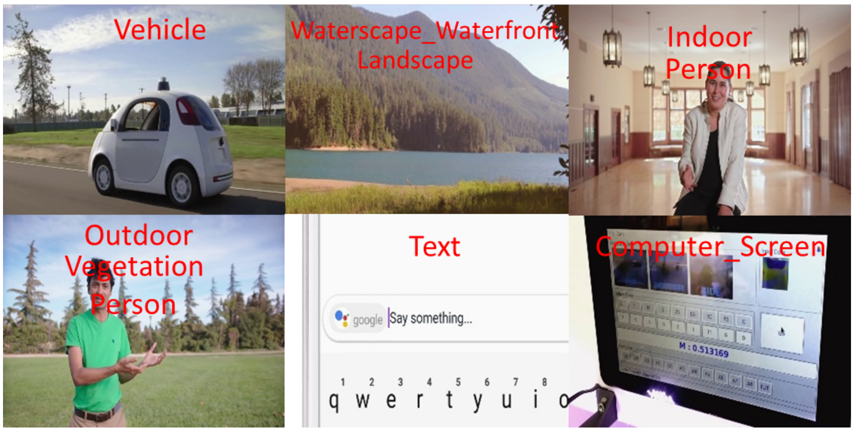


Figure 5: Concept-labeled video frames.

Table 4: Number of pages crawled by the FDC.

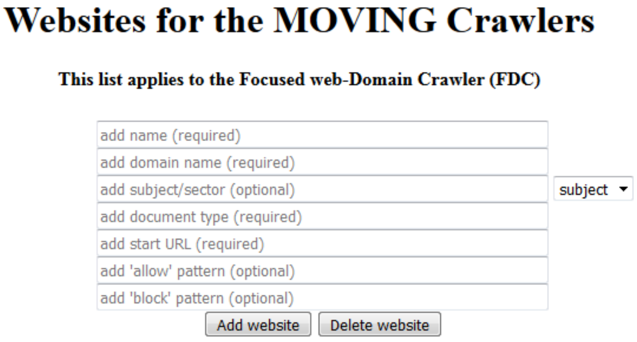
Domain	Start date	End date	Number of pages
academia.stackexchange.com	16/12/2017	12/01/2018	146,508
academicearth.org	15/12/2017	15/12/2017	526
allianz.com	24/01/2018	06/02/2018	58,798
bain.com	12/01/2018	19/01/2018	43,030
bayer.com	05/12/2018	23/01/2018	2,412
bcg.com	13/12/2017	16/12/2017	17,863
bdo.com	17/01/2018	24/01/2018	7,945
beiersdorf.com	12/01/2018	12/01/2018	740
bmw.com	16/12/2018	16/12/2018	181
canvas.net	16/12/2018	24/12/2017	26,380
class-central.com	24/12/2017	12/01/2018	130,064
coursera.org	28/12/2017	12/01/2018	106,914
curriki.org	13/01/2018	06/02/2018	110,255
daad.de	16/12/2017	28/12/2017	39,309
deloitte.com	13/12/2017	16/12/2017	12,415
deutsche-boerse.com	13/12/2017	15/12/2017	7,230
deutschepost.com	16/12/2017	16/12/2017	120
editage.com	15/12/2017	16/12/2017	2,175
edukatico.org	12/01/2018	24/01/2018	302,22
eur-lex.europa.eu	02/02/2018	06/02/2018	25,572
europa.eu	02/02/2018	06/02/2018	25,706
forrester.com	14/12/2017	12/01/2018	206,625
fresenius.com	12/01/2018	12/01/2018	160
freseniusmedicalcare.com	12/01/2018	13/01/2018	1,025
grad.ucla.edu	12/01/2018	24/01/2018	10,934
heidelbergcement.com	17/01/2018	17/01/2018	1,997
kpmg.com	12/01/2018	24/01/2018	43,570
mckinsey.com	12/01/2018	17/01/2018	23,705
oliverwyman.com	13/12/2017	14/12/2017	5,188
pwc.com	13/12/2017	16/12/2017	15,999
telekom.com	13/12/2017	12/01/2017	114,006
volkswagenag.com	02/02/2018	06/02/2018	14,148
Total			**1,231,722**

3.1.4 Implementation, APIs and integration

The crawlers run as separate system services. Instead of directly accessing Elasticsearch[6] to index the document as in the initial implementation, the crawlers rely on the intermediate Data Integration Service (DIS, see

[6] `https://www.elastic.co/products/elasticsearch`, see deliverables D4.1 (Gottfried, Grunewald, et al., 2017) and D4.2: *Initial responsive platform prototype, modules and common communication protocol* (Gottfried, Pournaras, et al., 2017) for more details on the MOVING platform's architecture

Section 2.3) to send the documents and allow it to handle the indexing. The maximum number of concurrent spiders [7] crawling and the minimum delay between the requests are configurable to allow the crawling process to be scaled up or down depending on the hardware and network resources available. The Crawlers' Input UI has also been updated to support new user-inserted attributes and settings for the FDC such as the sector/subject of the domain and the block/allow URL patterns (Figure 6).



Websites for the MOVING Crawlers

This list applies to the Focused web-Domain Crawler (FDC)

add name (required)
add domain name (required)
add subject/sector (optional) subject ▾
add document type (required)
add start URL (required)
add 'allow' pattern (optional)
add 'block' pattern (optional)

Add website | Delete website

Figure 6: FDC input fields.

3.2 FLuID: a flexible schema-level index model for Linked Data retrieval

3.2.1 Problem statement

The Web of Data, a global dataspace where providers can freely publish their content, is a valuable source of semantic, semi-structured knowledge. However, finding such knowledge is a challenging task since there is a vast amount of data available, which is of different kind and distributed over various data sources. Indexing can help to efficiently process huge amounts of data and, for many application scenarios, indices that fit the specific information need are available (Hose, Schenkel, Theobald, & Weikum, 2011). For graph data, we can distinguish between instance-level indices and schema-level indices. Instance-level indices focus on the fast retrieval of nodes or answering queries regarding reachability, distance, and shortest path (Sakr & Al-Naymat, 2010). They support the efficient execution of keyword queries, e. g., searching for metadata about the book "Towards a clean air policy" (Tran, Haase, & Studer, 2009). Schema-level indices (SLIs) focus on summarising nodes sharing common characteristics, i. e., the combination of the types attached and the property-labels used. Thus, SLIs support the efficient execution of structural queries, e. g., searching for bibliographic metadata using the type *bibo:book* (Christodoulou, Paton, & Fernandes, 2013). For example, instead of indexing the instance-level information illustrated in Fig. 7, a simple schema-level index would only memorise the combined use of the types, e. g., *bibo:Book* and *dct:BibliographicResource*. With such SLIs, search systems like LODeX (Benedetti, Bergamaschi, & Po, 2014, 2015) and LODatio (Gottron, Scherp, Krayer, & Peters, 2013) support their users in finding and exploring data sources based on combinations of RDF[8] types and/or properties.

In the past, various SLIs have been developed (Goldman & Widom, 1997; McHugh, Abiteboul, Goldman, Quass, & Widom, 1997; Neumann & Moerkotte, 2011; Ciglan, Nørvåg, & Hluchý, 2012; Konrath, Gottron, Staab, & Scherp, 2012; Benedetti et al., 2015; Spahiu, Porrini, Palmonari, Rula, & Maurino, 2016; Schaible, Gottron, & Scherp, 2016). The problem is that existing SLIs provide their own, proprietary data structure that is designed for solving a specific application scenario or answering only a specific information need. To the best of our knowledge, there is no parameterised index model to flexibly define different SLIs, so far. This is unfortunate since there is no single index model for the Web of Data that serves all the different information needs and application scenarios. We abstract from the existing indices and provide with Formal schema-Level Index model for the web of Data (FLuID) a generic, formal model to flexibly define different SLIs.

The major updates compared to the adaptive index models for Linked Data retrieval defined in deliverable D3.1 (Blume et al., 2017) are (1) a universal representation of SLIs using generic schema elements as basic building blocks (FLuID model) unifying existing approaches and (2) an extended evaluation and discussion of additional SLIs modeled with FLuID not only regarding approximation quality but also regarding storage requirements. Below, we motivate our parameterised index model for the Web of Data describing the MOVING scenario. Subsequently, we summarise our contributions and review related work.

[7] https://doc.scrapy.org/en/latest/topics/spiders.html
[8] http://www.w3.org/standards/techs/rdf

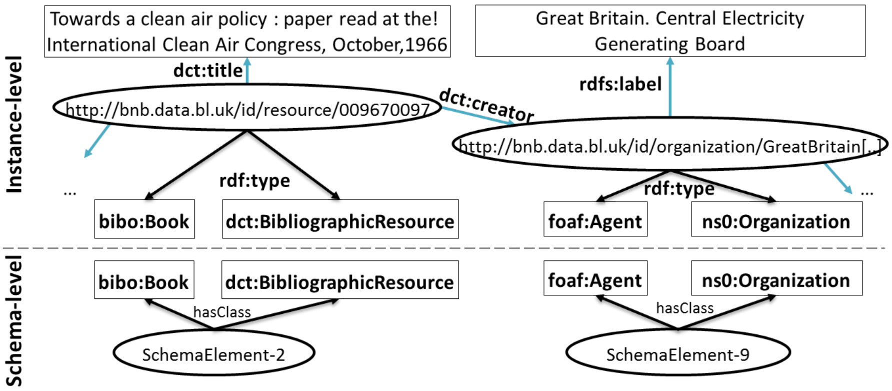

Instance-level
Schema-level
Towards a clean air policy : paper read at the! International Clean Air Congress, October,1966
Great Britain. Central Electricity Generating Board
dct:title
rdfs:label
dct:creator
http://bnb.data.bl.uk/id/resource/009670097
http://bnb.data.bl.uk/id/organization/GreatBritain[..]
rdf:type
rdf:type
...
...
bibo:Book
dct:BibliographicResource
foaf:Agent
ns0:Organization
bibo:Book
dct:BibliographicResource
foaf:Agent
ns0:Organization
hasClass
hasClass
SchemaElement-2
SchemaElement-9

Figure 7: A Bibliographic metadata record provided by the British National Library (instance-level information) and a type-only schema structure (schema-level information).

Motivating Scenario To enhance the content in the MOVING platform, additional bibliographic metadata is automatically harvested from the Web of Data. In order to automatically find such additional bibliographic metadata, we exploit common structural components of bibliographic metadata (Jett, Nurmikko-Fuller, Cole, Page, & Downie, 2016) and create a schema-level index. This index allows discovering data sources containing information about bibliographic resources that we can add to our index. For example, we can search for data sources containing instances of type *bibo:Book*, connected over the property *dct:creator* to another instance of type *foaf:Agent* (see Fig. 7). Knowing where such data can be found, the actual instance information can be (regularly) harvested in a subsequent step. However, there are various possibilities to model bibliographic metadata (Jett et al., 2016) resulting in different graph data structures varying in the number of relevant nodes and completeness. In addition to structural variations of the graph, also various combinations of vocabularies can be used to model similar or even the same information (Jett et al., 2016). Thus, incorporating semantic features like exploiting *owl:sameAs* links and inferencing over ontologies can ease finding relevant data sources. A parameterised model helps dealing with the different variations how bibliographic metadata is modeled on the Web of Data. For more details on the harvesting component, please see Section 3.2.4.

So far, there exists no solution of how to flexibly define schema-level indices to quickly cover all requirements. Thus, a parameterised schema-level index model is needed to develop and compare SLIs targeting different information needs and application scenarios. Furthermore, the Web of Data is growing quickly in size. Thus, efficient solutions are needed to compute such SLIs.

Contributions Based on an extensive analysis of existing SLIs and the requirements from the scenarios above, we have determined the principal functionalities and building blocks to a formal, parameterised model. FLuID distinguishes between schema information and payload information (Gottron et al., 2013). We model the schema information using equivalence relations over RDF instance data (Konrath et al., 2012). These equivalence relations form the so-called schema elements. FLuID consists of three simple schema elements and one complex schema element. Simple schema elements are the basic building blocks and summarise RDF instances, i. e., sets of triples sharing a common subject URI. Complex schema elements are combinations of simple schema elements and can summarise arbitrary graph structures on the Web of Data. Both, the simple schema elements and the complex schema elements, can be further specified using four parameters defined in the following sections.

The chaining parameterisation determines the size of the considered path length to *k*-hops (Tran, Ladwig, & Rudolph, 2013), which basically determines how many simple schema elements are combined to a complex schema element. The label parameterisation allows restricting the schema-level index to consider only specific properties (Tran et al., 2013). The instance parameterisation allows summarising instances, e. g., connected over *owl:sameAs* (Ding, Shinavier, Shangguan, & McGuinness, 2010). Finally, inference parameterisation enables ontology reasoning, e. g., exploiting *rdfs:domain*, *rdfs:range*, *rdfs:subClassOf*, and *rdfs:subPropertyOf*. An overview of FLuID‘s building blocks is given in Table 5.

The payload comprises information about schema elements which is needed for implementing a specific application scenario. For example, search engines like LODeX (Benedetti et al., 2014) and LODatio (Gottron et al., 2013) store statistical information like the number of summarised instances or store data source URIs to memorise where on the Web of Data instances with a certain combination of RDF types and properties can be found. Like the schema information, the payload is also parameterised and can be adapted to different requirements.

Related work SLIs support to efficiently execute structural queries over distributed graph data. Structural queries focus on how nodes (resources) are described, i. e., which combinations of types and properties are used to model the resources. There are various different possibilities and variants of how to define an SLI and different definitions of SLI allow for capturing different schemas. In the following, we present an overview of SLIs with emphasis on their schema structure, their application scenario, and how they were formalised.

Characteristic Sets (Neumann & Moerkotte, 2011) summarise instances along common incoming properties and outgoing properties. They were defined as sets of instances using a first-order-logic expression over triples. They were evaluated with respect to the accuracy of cardinality estimations for queries in RDF databases. SemSets (Ciglan et al., 2012) are defined as sets of instances that share the same outgoing properties which are connected to a common target resource. They were defined as sets over their own "Property Graph Data Model". They were developed to discover semantically similar sets in knowledge graphs in order to improve keyword-based ad-hoc retrieval. Christodoulou et al. (Christodoulou et al., 2013) applied a hierarchical clustering algorithm on RDF data in order to determine clusters, i. e., sets of instances that are characterised by the same set of properties. They were defined as sets in an textual definition. The clusters are annotated using the RDF type information of the clustered instances, which is then used to derive a schema from the data sources on the Web of Data. Their largest dataset contained only 1.1 Million triples. ABSTAT (Spahiu et al., 2016) and LODeX (Benedetti et al., 2014, 2015) summarise instances based on a common set of RDF types and properties linking to resources with the same set of types. ABSTAT's schema structure was informally defined as triples in a textual description. Additionally, ABSTAT selects a minimal number of types from the set of types such that all remaining types are sub-classes of the selected types. ABSTAT was evaluated on different datasets including DBpedia and Linked Brainz. LODeX's schema structure was defined using a new grammar for their own model. LODeX uses a clustering of RDF types to select a representative type. Thus, they can comprehensively visualise several datasets hosted on DataHub. TermPicker (Schaible et al., 2016) summarises instances based on a common set of types, a common set of properties, and a common set of types of all property-linked resources. The schema structure was informally introduced by examples. The goal was to make data-driven recommendations of vocabulary terms.

One of the first SLIs using bisimulation is DataGuides (Goldman & Widom, 1997; McHugh et al., 1997). Bisimulation operates on state transition systems and defines an equivalence relation over states (Sangiorgi, 2009). Two states are considered equivalent (or bisimilar) if they change into equivalent states with the same type of transition. Interpreting a labeled graph as a representation of a state transition system allows for the application of bisimulation on RDF data in order to discover structurally equivalent parts in the graph. Thus, DataGuides (Goldman & Widom, 1997; McHugh et al., 1997) summarises instances for which all outgoing paths for the whole subgraph are equivalent. DataGuides were evaluated on relational database systems using the Object Exchange Model (OEM). Since then, several SLIs adapted the idea of bisimulation and applied a stratified k-bisimulation on RDF and OEM (Nestorov, Ullman, Wiener, & Chawathe, 1997; Konrath et al., 2012). A stratified k-bisimulation is a bisimulation where the maximum length of the considered path is k edges long (Sangiorgi, 2009). Another example is SchemEX (Konrath et al., 2012), that summarises instances similar to ABSTAT and LODeX, based on a common set of types and properties linking to resources with a common set of types. However, it does not perform any selection of types for the purpose of cluster labeling.

All SLIs presented above define a single, fixed schema structure. These schema structures are often defined informally in a textual description or only explained by examples (Ciglan et al., 2012; Christodoulou et al., 2013; Spahiu et al., 2016; Schaible et al., 2016). One investigated SLI is defined using a mathematical model, which, however, was designed not for the SLI directly but rather the surrounding context (Benedetti et al., 2015). Only few indices are defined using basic first-order-logic (Goldman & Widom, 1997; Neumann & Moerkotte, 2011; Konrath et al., 2012), which could be reused. However, to the best of our knowledge, there is only a single parameterisation of a schema-level index suggested by Tran et al. (Tran et al., 2013). They model a label-parameterised and height-parameterised index for executing structured queries on LOD. With label-parameterisation, only specific properties are considered and height-parameterisation limits the maximum path-length of the subgraphs stored in the index. In summary, there exists no single, fully parameterised model which formally describes SLIs in general and which can be reused in order to develop, compare, and validate schema-level indices.

3.2.2 Method description

3.2.2.1 Foundations: equivalences over Linked Data

A data graph G is defined by $G \subseteq V_{UB} \times P \times (V_{UB} \cup L)$, where V_{UB} denotes the set of URIs and blank nodes, P the set of properties, and L the set of literals. A triple is a statement about a resource $s \in V_{UB} \cup P$ in the form of a subject-predicate-object expression $(s, p, o) \in G$. We define instances $I_s \subseteq G$ to be a set of triples, where each triple shares a common subject URI s. We say the instance I_s with the subject URI s is of type c if there exists a triple $(s, rdf{:}type, c) \in G$. Moreover, the properties P can be divided into disjoint subsets $P = P_{type} \dot{\cup} P_{rel}$, where P_{type} contains the properties denoting type information and P_{rel} contains the properties between instances in the data graph. If not stated otherwise, P_{type} only contains *rdf:type* and P_{rel} all $p \in P \setminus P_{type}$.

SLIs summarise instances based on their schema, i. e., common use of types and properties. For SLIs, we can distinguish between *abstract schema level* and the *entity mapping level* (R. Q. Dividino, Scherp, Gröner, & Grotton, 2013). In our context, the abstract schema level defines the schema given by the index definition, e. g., taking only properties into account. We call these the *Schema Elements*. The entity mapping level is a concrete assignment of an instance to such an Schema Element. We call Schema Elements with instances mapped to them *Instantiated Schema Elements*.

Each instance uses a defined set of types and properties and thus exactly one schema. Therefore, the mapping of instances to Instantiated Schema Elements is unique. SLIs partition the data graph into disjoint subsets of instances, where each subset is described by an Instantiated Schema Element. Equivalence relations can describe this graph partitioning in a formal way (European Mathematical Society, 2014).

Definition 1 (Equivalence Relation)**.** *For a given set X, an equivalence relation on X is a subset $EQR \subseteq X \times X$, that is reflexive, symmetric, and transitive. When $(x, y) \in EQR$, we say that x is equivalent to y or $x \sim y$. For any $y \in X$, the subset of X of all x that are equivalent to y is called the equivalence class of y, denoted $[y]_{EQR}$.*

Any two equivalence classes are either disjoint or coincide. This means that any equivalence relation on X defines a partition (decomposition) of X, and vice versa (European Mathematical Society, 2014). Furthermore, it can be shown that the intersection of two equivalence relations over X is also an equivalence relation. In order to ensure the correctness of the approach, we formally define instances as equivalence relation over the data graph G. With instances being defined as equivalence relation any equivalence relation on top of instances consequently will be an equivalence relation over the data graph.

Definition 2 (Instance)**.** *Instances are sets of triples in the data graph G sharing a common subject URI. The equivalence relation $I \subseteq G \times G$ is defined as $((i_1, p_1, o_1), (i_2, p_2, o_2)) \in I \Leftrightarrow\ i_1 = i_2$. We write $[i]_I$ or I_i to denote the equivalence class of the instance with subject URI i.*

This definition of an instance maps each triple in G to exactly one instance determined by its subject URI. Thus, Def. 2 defines a partition over the data graph G and consequently qualifies as an equivalence relation (European Mathematical Society, 2014). In the context of SLI, we call equivalence classes of instances the schema elements. We connect the schema elements to generated instance information by using the notion of payload (Gottron et al., 2013). The payload comprises information about the actual data, e. g., all instances or only references to their data source. In summary a SLI can be defined over the data graph G, an equivalence relation EQR, and an n-tuple of payload function PAY.

Definition 3 (Schema-level Index SLI)**.** *Formally, a schema-level index is a 3-tuple (G, EQR, PAY), where G is the data graph which is indexed, EQR is an equivalence relation over instances in G, and PAY is an n-tuple of payload functions, which map instance information to equivalence classes in EQR.*

In the following sections, we introduce the three parameterised simple schema elements and one parameterised complex schema element. These are sufficient to express the functionalities of existing SLIs and beyond.

3.2.2.2 FLuID's building blocks

The FLuID model consists of basic building blocks, which can be combined to define any given SLI. We have simple and complex schema elements, which can be further specialised with our four parameterisations. An overview is given in Table 5. This section is organised into two parts: first we define the simple and complex schema elements and then we continue with the parameterisation.

Table 5: Overview of the FLuID model building blocks divided into base configuration as well as into possible parameterisations.

Building block	Description	Details	
Simple Schema Element (SSE)	Triples based summarisation of instances.	Def. 4, Figs. 8a and 8c	Base configuration
Complex Schema Element (CSE)	Summarisation of instances using combinations of SSEs	Def. 5, Figs. 8b and 8d	
Chaining parameter cp	Recursively repeat base-configuration for each connected instance	Def. 6	Parameterisation
Label parameter P_l	Ignore existence of properties not in P_l	Def. 7	
Instance parameter *owl:sameAs*	Transitively include instances in the *owl:sameAs*-network.	Def. 8 and 9	
Ontology inference parameter Schema Graph (SG)	Include ontology reasoning by inferring additional triples, e. g., using RDFS	Def. 10	

Schema elements Our first simple schema element (SSE) is the Property-Object Cluster, which summarises instances based on common property-object tuples.

Definition 4 (Property-Object Cluster POC)**.** *Property-Object Clusters partition the data graph by summarising instances $I_1 \in G$ and $I_2 \in G$ based on a common set of objects and a common set of properties. The equivalence relation POC holds true, iff for each triple in both instances there exists a matching triple in the other instance, such that the property and the object are the same.*

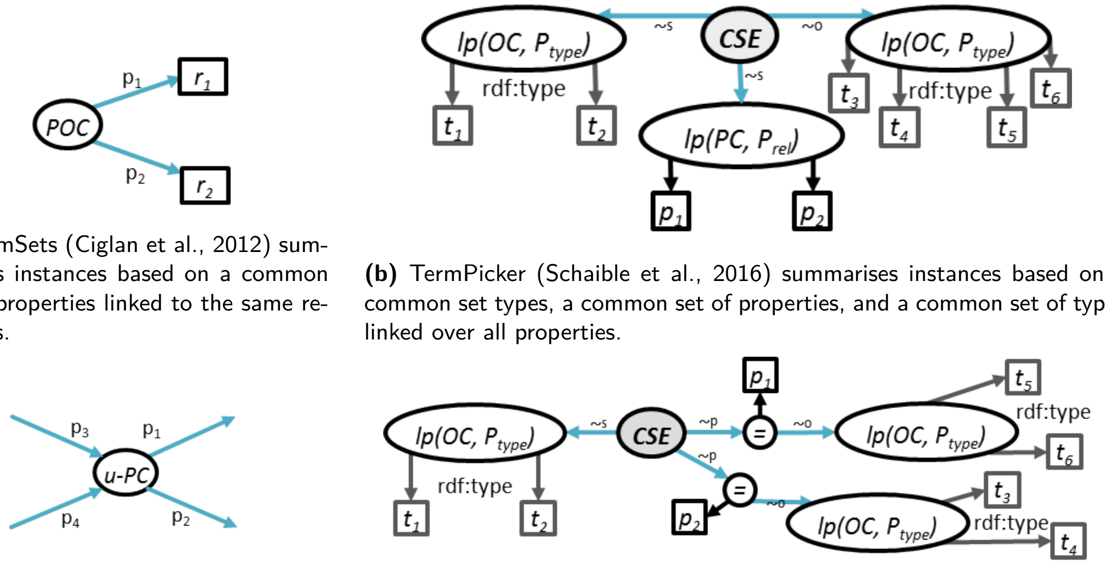
The Property-Object Cluster is sufficient for a schema structure defined by SemSets (see Fig. 8a) since it requires the objects to be connected over the same property. However, schema structures like TermPicker (see Fig. 8b) and Characteristic Sets (see Fig. 8c) define schema structures considering properties independent of the object. To support this, we define analogously to the Property-Object Cluster two further simple schema elements called Property Cluster (PC) and Object Cluster (OC). The PC summarises instances based on the same properties ($p_1 = p_2$) and the OC based on the same objects ($o_1 = o_2$).

Our three simple schema elements OC, PC, and POC only take outgoing properties into account. However, schema structures like Characteristic Sets (Neumann & Moerkotte, 2011) consider also incoming properties (see Fig. 8c). To address incoming properties, an undirected version of the three simple schema elements can be defined by additionally considering the incoming triples $(x, p, i) \in G$ with i as the subject of the instance being in object position. The undirected Property Cluster u-PC resembles the schema structure of Characteristic Sets (see Fig. 8c).

In total, FLuID provides six simple schema elements. Our simple schema elements summarise instances by comparing incoming and outgoing triples of the instance. However, some SLIs like SchemEX (Konrath et al., 2012), TermPicker (Schaible et al., 2016), ABSTAT (Spahiu et al., 2016), and LODeX (Benedetti et al., 2014, 2015) define schema structures that go beyond the scope of a single instance. Thus, we define complex schema elements as an extension of simple schema elements. The simple schema elements are already combinations of equivalence relations by using the identity equivalence "=" on properties and objects. We extend on this fact and define a generic complex schema element.

Definition 5 (Complex Schema Element CSE)**.** *A complex schema element partitions the data graph by summarising instances based on the three given equivalence relations $\sim^s$, $\sim^p$, and $\sim^o$ and therefore can be defined as 3-tuple $CSE := (\sim^s, \sim^p, \sim^o)$. Two instances I_1, I_2 are considered equivalent, iff $i_1 \sim^s i_2 \ \wedge \ p_1 \sim^p p_2 \ \wedge \ o_1 \sim^o o_2$ holds true for all triples in both instances with i_1, i_2 as the subjects of I_1, I_2 respectively.*

Example 1. *We demonstrate the benefit of complex schema elements by defining $CSE\text{-}1 := (PC, T, PC)$ and $CSE\text{-}2 := (PC, =, PC)$, with T being an arbitrary tautology. Since T considers all properties equal, the Property Cluster in object position of CSE-1 considers sets properties. In contrast, CSE-2 uses the identity equivalence on predicate position, thus, all 2-hop property paths have to match exactly. The two instances $[i_3]_I$ and $[i_4]_I$ with outgoing properties as illustrated in Fig. 9 are considered equal according to the equivalence of CSE-1 since the 1-hop properties are equal and the 2-hop properties are equal. However, according to CSE-2, they are not considered equal, since the property paths are not identical.*



(a) SemSets (Ciglan et al., 2012) summarises instances based on a common set of properties linked to the same resources.

(b) TermPicker (Schaible et al., 2016) summarises instances based on a common set types, a common set of properties, and a common set of types linked over all properties.

(c) Characteristic Sets (Neumann & Moerkotte, 2011) summarises instances based on a common set of incoming and outgoing properties.

(d) SchemEX (Konrath et al., 2012) summarises instances based on a common set of types and a common set of properties linked to resources sharing the same set of types.

Figure 8: Simplified visualisation of schema structures captured by different approaches from the related work modeled using SSEs and CSEs.

Parameterisations

Chaining parameterisation Tran et al. (Tran et al., 2013) suggest a parameterisation regarding the maximal path length of the subgraph structure. We consider this in our FLuID model by defining the chaining parameterisation cp. As illustrated in Fig. 9, the complex schema element can consider the neighborhood of up to two hops. When chaining k such schema elements, the same pattern is recursively applied up to k hops. This can be formalised as stratified k-bisimulation (Luo, Fletcher, Hidders, Wu, & Bra, 2013).

Definition 6 (Chaining parameterisation cp)**.** *The chaining parameterisation is a function $cp(CSE,k)$, which takes a complex schema element CSE $:= (\sim^s, \sim^p, \sim^o)$ and a chaining parameter $k \in \mathbb{N}$ as input and returns an equivalence relation CSE_k. Formally, this chaining of k complex schema elements up to length k can be recursively defined as bisimulation (Sangiorgi, 2009). Two instances $[i_1]_I$ and $[i_2]_I$ are equivalent according to $cp(CSE,k)$ if three conditions hold true: (1) For $k = 0$ the subject equivalence $i_1 \sim^s i_2$. (2) For $k > 0$ all three equivalences $i_1 \sim^s i_2 \ \wedge \ p_1 \sim^p p_2 \ \wedge \ o_1 \sim^o o_2$. (3) For $k > 0$ the recursion step $([o_1]_I, [o_2]_I) \in cp(CSE, k-1)$.*

Label parameterisation SchemEX and TermPicker (see Figs. 8b and 8d) use common sets of types in their schema structure. We address this by introducing the label parameterisation lp, which allows ignoring a certain set of properties.

Definition 7 (Label parameterisation lp)**.** *The label parameterisation is a function $lp(EQR, P_r)$, which takes as input an equivalence relation EQR and a set of properties $P_r \subseteq P$ and returns an equivalence relation EQR_{P_r}. The returned equivalence relation EQR_{P_r} is a restriction of EQR in terms that all assertions about the triples in EQR only need to be true iff the property of the triple is included in the parameter property set P_r.*

Restricting any schema element with such a property set in fact relaxes the constraints given by the schema element. For example, the label parameterisation lp applied on the Object Cluster OC using the properties P_{type} summarises instances which have the same set of resources connected over the property rdf:type. This means any other object is not relevant to determine the equivalence. Please note, any label parameterised schema element still qualifies as equivalence relation since the same principle as before applies.

Using the properties P_{type}, the label parameterised Object Cluster $lp(OC, P_{type})$ summarises instances which have the same set of resources connected with the property *rdf:type*. This allows us to model schema structures like TermPicker (Schaible et al., 2016) as complex schema element (CSE):

$$(lp(OC, P_{type}) \cap lp(PC, P_{rel}), lp(=, \emptyset), lp(OC, P_{type}))$$

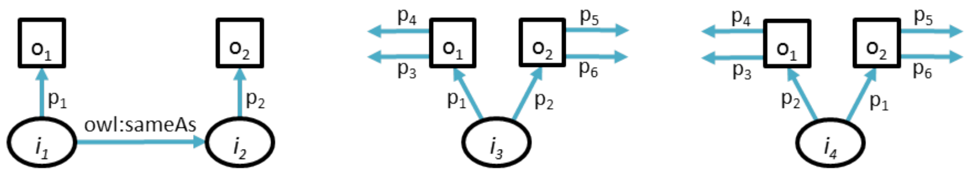


Figure 9: Sample data graph which is summarised to either four Object Cluster or two instance parameterised Object Cluster using SameAs Instances $[I]_\sigma$.

The example of TermPicker is illustrated in Fig. 8b. To model TermPicker, we make use of the intersection of the label parameterised Object Cluster and the label parameterised Property Cluster. This way, instances need to have the same type sets and the same Property Cluster. The independence of the objects' type sets and the connecting properties can be achieved using $lp(=,\emptyset)$ as predicate equivalence $\sim^p$ in the complex schema element. The identity equivalence on the empty set is a tautology. Thus, basically, all type sets of the connected resources are merged, as illustrated in Fig. 8b and explained in Example 1. The schema of ABSTAT (Spahiu et al., 2016), LODeX (Benedetti et al., 2014, 2015), and SchemEX (Konrath et al., 2012) is defined straightforward:

$$(lp(OC,P_{type}),lp(=,P_{rel}),lp(OC,P_{type}))$$

Support for unions of RDF instances FLuID supports a parameterisation of the instance definition which allows considering instances that resemble the same real-world entity by using the *owl:sameAs* property. In order to take this information into account, we formally introduce SameAs instances.

Definition 8 (SameAs Instance). *The equivalence relation σ summarises instances based on the semantics of owl:sameAs in equivalence classes $[I]_\sigma$, called SameAs instance. For all instances $[i_1]_I,[i_2]_I \in [I]_\sigma$, there is a path over all edges (independent of the edges direction) labeled owl:sameAs in G from i_1 to i_2.*

Please note that the *owl:sameAs* property is transitive, symmetric, and reflexive[9]. Furthermore, it can be shown that the assignment of an instance to a SameAs Instance is unique by reducing the problem to finding weakly connected components in an *owl:sameAs*-labeled subgraph of G, as it has been done by Ding et al. (Ding et al., 2010). With the notion of σ, we can now define the instance parameterisation to consider SameAs instances instead of instances.

Definition 9 (Instance parameterisation ip). *The instance parameterisation is a function $ip(EQR,\sigma)$, which extends any simple or complex schema element EQR to additionally consider all connected instances following the instance equivalence relation σ. The returned equivalence relation EQR_σ is an extension of EQR, which restricts the triples to be in $[I]_\sigma$.*

As an example, we apply the instance parameterisation ip on the Object Cluster equivalence relation OC using the SameAs instances: $(I_1,I_2) \in ip(OC,\sigma) \Leftrightarrow \forall(i_1,p_1,o_1) \in [I_1]_\sigma \; \exists(i_2,p_2,o_2) \in [I_2]_\sigma : o_1 = o_2$ (and vice versa). As the example shows, the instance parameterised OC considers the SameAs network (Ding et al., 2010) and thus merges instances. Fig. 9 shows an example graph. According to the Object Cluster definition, the instances $[i_1]_I$, $[i_2]_I$, and $[i_3]_I$ are not equivalent. Summarising $[i_1]_I$ and $[i_2]_I$ to a SameAs instance $[I]_\sigma$ leads to the equivalence of all three instances.

Support for ontology inferencing In the Web of Data, there are assertions about individuals and assertions about RDF types and properties (De Giacomo & Lenzerini, 1996). For example, a dataset can contain the following assertions about the triples of the bibliographic records shown in Fig. 7.

<http://bnb.data.bl.uk/doc/resource/009670097> **<dct:creator>**
<http://bnb.data.bl.uk/id/organization/GreatBritain[..]> .
<dct:creator> **<rdfs:domain>** <bibo:Document> .
<dct:creator> **<rdfs:range>** <foaf:Person> .

The triples using *rdfs:domain* and *rdfs:range* allow inferring additional knowledge about individuals using the property *dct:creator*. The schema summarisation tool ABSTAT (Spahiu et al., 2016) incorporates information derived from an ontology by inferring triples based on a subtype schema graph. ABSTAT's schema graph is

[9] https://www.w3.org/TR/2004/REC-owl-semantics-20040210/

constructed by extracting the contained schema assertions. We extend the idea of the schema graph from ABSTAT but include all RDFS properties in the schema graph. Thus, our RDFS schema graph contains hierarchical dependencies of *rdfs:subClassOf* and *rdfs:subPropertyOf* in a tree structure with further cross connections regarding *rdfs:range* and *rdfs:domain*.

Definition 10 (Schema Graph)**.** *Let* $SG := (V_C \cup P, \mathscr{E})$ *be an edge-labeled directed multigraph and* $\mathscr{E} \subseteq (V_C \cup P) \times (V_C \cup P)$*. The set of nodes is the union of the set of RDF classes and properties. The edge-label function* $\phi : \mathscr{E} \rightarrow P$ *assigns labels from a given set of possible properties* P *to all edges* $e \in \mathscr{E}$*.*

Please note, multigraphs allow parallel edges between nodes, thus multiple relationships between nodes are possible in SG. Such a schema graph enables search for related types and properties.

We construct the RDFS schema graph by extracting all triples containing RDFS vocabulary terms, namely all properties $P_{RDFS} = \{$*rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:range*, *rdfs:domain*$\}$ and label the schema graph using the RDFS edge-label function ϕ_{RDFS}. In the following, we denote the schema graph constructed using the labeling function ϕ_{RDFS} with SG_{RDFS}. Having the hierarchically dependencies of types and properties represented using a Schema Graph, additional triples can be inferred.

Definition 11 (Inferencing parameterisation)**.**
The inferencing parameterisation is a function $\Phi(G, SG)$*, which takes any data graph* G *and schema graph* SG *as input and based on the entailment rules defined in the schema graph* SG *returns a data graph* G_Φ*, which additionally includes all inferred triples.*

Please note, when including the semantics of *owl:sameAs* and RDF Schema, one should explicitly exclude the properties *owl:sameAs* and P_{RDFS} from the instance equivalence relation.

Payload The payload comprises the information connected to a schema element which is needed for the specific application scenario, e. g., the data sources of the summarised instances needed for a search engine (Gottron et al., 2013). The payload PAY is an n-tuple of mapping functions, where each function maps the schema elements to one payload element.

Definition 12 (Payload)**.** *The Payload PAY is an n-tuple of mapping functions, which map schema elements to specific payload elements.*

One such function is the data source mapping function ds, which maps a schema element to the data sources of all summarised instances. Schema elements are defined as equivalence classes using equivalence relations over a set of instances which are sets of triples. Therefore, we can treat a schema element as a set of instances and for each instance, we can extract the data source, e. g., by extracting the graph information from quads[10]. All instances from such an equivalence class can be mapped using the data source function ds (Gottron et al., 2013).

Definition 13 (Datasource ds)**.** *A schema element* EQR *is mapped to the data sources using a function* ds*, which takes a schema element* EQR *as input and returns all sources, with* $ds(EQR) := \bigcup_{I \in EQR} context(I)$*, with the function* $context : \mathscr{P}(G) \rightarrow \mathscr{P}(V_U)$ *returning all data sources of an instance* I*.*

For different applications, e. g., cardinality estimation (Neumann & Moerkotte, 2011), you may define mapping functions, which return payload elements containing information about the number of instances summarised by one schema element.

3.2.3 Experimental evaluation and comparison

To compute a schema-level index, the defined schema elements need to be computed from the instance data. This can, for example, be done, by extracting all properties of an instance to form a Property Cluster. For instances using the same set of properties, the same schema element is computed. Consequently, the Property Cluster summarises the corresponding instances. Instance summarisation can be implemented using hash maps, which ensures constant time access (Konrath et al., 2012). It can be shown that all indices modeled with FLuID can be computed in $\mathscr{O}(n)$ with n triples in the dataset.

FLuID provides parameterised simple and complex schema elements which allow modeling various SLIs. We demonstrate this by presenting the formalisation for five example indices from the related work in Table 6.

We run experiments with four SLIs directly taken from the related work and the SchemEX+U+I index as a semantic version of SchemEX, where we include SameAs instances σ and an on-the-fly generated RDFS[11]

[10] `https://www.w3.org/TR/n-quads/`, last accessed: 28/02/2018
[11] `http://www.w3.org/TR/rdf-schema/`

Table 6: Example schema-level indices from the related work defined using our FLuID model. G denotes the data graph. The payload $PAY_{ds} := (ds)$ contains the data sources using the ds mapping function from Definition 13.

Name	Definition	Source
CharacteristicSets	$(G, u\text{-}PC, PAY_{ds})$	(Neumann & Moerkotte, 2011)
SemSets	(G, POC, PAY_{ds})	(Ciglan et al., 2012)
TermPicker	$(G, (lp(OC, P_{type}) \cap lp(PC, P_{rel}), lp(=, \emptyset), lp(OC, P_{type}), PAY_{ds})$	(Schaible et al., 2016)
SchemEX	$(G, (lp(OC, P_{type}), lp(=, P_{rel}), lp(OC, P_{type})), PAY_{ds})$	(Konrath et al., 2012; Gottron et al., 2013)
SchemEX+U+I	$(\Phi(G, SG_{RDFS}), (lp(ip(OC, \sigma), P_{type}), lp(ip(=, \sigma), P_{rel}), lp(ip(OC, \sigma), P_{type}), PAY_{ds})$	(Konrath et al., 2012; Gottron et al., 2013)

schema graph SG_{RDFS}. The payload of each index comprises the data sources, which is necessary for the evaluation. The indices are formalised in Table 6 and visualised in Fig. 8. Furthermore, we change the original indices and their complexity by setting the chaining parameter value k to 0, 1, and 2. For the empirical evaluation, we focus on a data search application scenario as outlined in Section 3.2.1. A search engine for the Web of Data aims to return data sources containing triples that match a given information need, e. g., query for bibliographic metadata records. The information about the data source is coming from the payload.

3.2.3.1 Procedure

The evaluation is twofold. First, for five indices motivated from the related work, we evaluate the storage requirements of the computed index. Second, we evaluate the quality of a stream-based index computation.

Storage requirements Regarding storage requirements, we evaluate the space requirements for the resulting indices when stored as an RDF graph. To this end, we compare the number of triples in the index and the number of triples in the dataset. The size of the chosen indices is independent of the stream-based approach since the index is exactly computed. Furthermore, we compare the number of different schema elements in the index to the number of instances in the dataset. The computed ratio gives an idea of how well the defined schema structure can summarise instances on the Web of Data.

Approximation quality We implemented a scalable index computation using a stream-based approach, where we observe the data over a stream of triples using a window technique from stream databases (Garofalakis, Gehrke, & Rastogi, 2016; Konrath et al., 2012). Such a stream-based approach allows us to compute indices for graphs of arbitrary size while requiring only limited computational power (e. g., single desktop computer). However, with the scalability of the stream-based approach occur approximation errors, since only a part of the data graph is in the main memory. Thus, we potentially extract incomplete schema information.

To evaluate the five SLIs with respect to the approximation quality, we compute the indices with a window size of 1k, 100k, and 200k schema elements and compare them to a gold standard. Each gold standard is an exactly computed index on our server machine with 1TB main memory. We evaluate if the payload is assigned to the correct schema element during the index computation. To this end, we apply queries on the approximative index as well as on the gold standard and compare the results. The queries are generated from the schema elements taken from the gold standard. We extract all combinations of types, properties, and resources from the gold standard schema elements and formulate queries accordingly. We choose to distinguish simple queries (SQ) and complex queries (CQ). Simple queries target (parameterised) Object Cluster. These simplest queries appear to be the most used SPARQL queries in search scenarios (Arias, Fernández, Martínez-Prieto, & de la Fuente, 2011). The complex queries target the (parameterised) Complex Schema Elements. The approximation quality is measured by comparing the set of data sources returned from an SLI given a query q using the harmonic F1-measure (Manning, Raghavan, & Schütze, 2008). Following Konrath et al. (Konrath et al., 2012), we define precision as $Pr(q) = \frac{|D_{gold} \cap D_{win}|}{|D_{win}|}$ and recall as $Re(q) = \frac{|D_{gold} \cap D_{win}|}{|D_{gold}|}$, where D_{gold} denotes the data sources from the gold standard and D_{win} the data sources from the stream-based index computation.

Table 7: The number of triples #t in Millions (M) the ratio compared to the number of triples in the dataset and the number of schema elements #e in Thousands (T) and the ratio compared to the number of instances in the dataset, for the TimBL-11M and DyLDO-127M and for the chaining parameter $k \in \{0,1,2\}$.

	k	#	Characteristic Sets	SemSets	SchemEX	TermPicker	SchemEX +U+I
TimBL-11M	0	t	na (na)	$0.3M$ (2.9%)	$0.3M$ (2.9%)	$0.3M$ (2.9%)	$0.4M$ (3.8%)
	0	e	na (na)	$2.8T$ (0.4%)	$2.8T$ (0.4%)	$2.8T$ (0.4%)	$3.1T$ (0.5%)
	1	t	$0.7M$ (6.5%)	$7.6M$ (69.2%)	$0.8M$ (6.9%)	$0.7M$ (6.5%)	$0.8M$ (7.1%)
	1	e	$9.6T$ (1.4%)	$13.9T$ (20.6%)	$12.0T$ (1.8%)	$10.8T$ (1.6%)	$11.3T$ (1.7%)
	2	t	$1.6M$ (14.6%)	$7.6M$ (69.2%)	$1.4M$ (12.5%)	$1.8M$ (16.0%)	$1.8M$ (15.9%)
	2	e	$37.2T$ (5.5%)	$13.9T$ (20.6%)	$27.7T$ (4.1%)	$37.3T$ (5.5%)	$31.0T$ (4.6%)
DyLDO-127M	0	t	na (na)	$4.1M$ (3.2%)	$4.1M$ (3.2%)	$4.1M$ (3.2%)	$8.5M$ (6.7%)
	0	e	na (na)	$46.6T$ (0.7%)	$46.6T$ (0.7%)	$46.6T$ (0.7%)	$53.0T$ (0.8%)
	1	t	$0.6M$ (0.5%)	$45.3M$ (35,6%)	$15.7M$ (12.3%)	$11.1M$ (8.7%)	$19.9M$ (15.7%)
	1	e	$23.0T$ (0.3%)	$1733.5T$ (25.0%)	$254.5T$ (3.6%)	$238.4T$ (3.4%)	$249.5T$ (3.5%)
	2	t	$2.1M$ (1.7%)	$45.3M$ (35,6%)	$19.8M$ (15.6%)	$25.4M$ (19.9%)	$22.9M$ (18.0%)
	2	e	$112.8T$ (1.6%)	$1733.5T$ (25.0%)	$431.1T$ (6.1%)	$559.1T$ (7.9%)	$466.9T$ (6.6%)

3.2.3.2 Datasets

We use two datasets of the Web of Data with different characteristics. Although reasonably large, both datasets allow us to compute a gold standard. The TimBL-11M dataset contains about 11 Million triples (673 Thousand instances) (Konrath et al., 2012). The crawling was conducted with a breadth-first search starting from the FOAF profile of Tim Berners-Lee and allows us to compare the approximation results to the previous work of Konrath et al. (Konrath et al., 2012). Regular snapshots from the Web of Data are provided by the Dynamic Linked Data Observatory (DyLDO) (Käfer, Abdelrahman, Umbrich, O'Byrne, & Hogan, 2013). We use their first snapshot containing about 127 Million triples (7 Million instances) crawled from about 95,000 seed URIs. This crawl was done with a breadth-first search but limited to a crawling depth of two (Käfer et al., 2013). Although there are more recent DyLDO snapshots, they are decreasing in size. Thus, we decided to take the first and largest one. Please also note that we did not perform any pre-processing despite removing invalid (not parsable) triples.

3.2.3.3 Results and discussion

Storage requirements We present the number of triples in the exactly computed index and the ratio of triples in the index compared to triples in the original dataset (compression ratio) as well as the number of schema elements partitioning the graph compared to the number of instances in the dataset (summarisation ratio) in Table 7.

SemSets' compression ratio (with $k = 1$) is about 10 times larger than all other indices for the TimBL-11M dataset and up to 75 times larger on the DyLDO-127M dataset. SemSets is the only evaluated index using plain Object Cluster. In contrast, the remaining indices either ignore objects or consider their type information. The size can be explained naturally when taking the summarisation ratio into account. SemSets has a summarisation ratio of $20\% - 25\%$ meaning on average $4-5$ instances share the same schema structure. The smallest index Characteristic Sets has a summarisation ratio of 0.3% meaning about 330 instances share the same schema structure. These two extremes in our experiment show a huge variety in terms of how well indices compress the data.

Including semantics of RDFS and *owl:sameAs* in SchemEX+U+I compared to SchemEX increases the index size by about 3% more triples. This suggests that not many triples using properties in P_{RDFS} or *owl:sameAs* exist in both datasets. Despite being a larger index in terms of the number of triples, for $k = 1$ there exist fewer partitions on the data graph when including semantics of RDFS and *owl:sameAs*. However, for $k = 0$ and $k = 2$ SchemEX+U+I has more different schema elements than SchemEX. In our motivating scenario outlined in Section 1, we expect more relevant search results with such a reduced set of schema elements. Characteristic

Table 8: Harmonic F1-measure or not available (na) for simple queries (SQ) and complex queries (CQ), the chaining parameter $k \in \{0,1,2\}$, and for different window sizes (1k, 100k, 200k) on the two datasets TimBL-11M and DyLDO-127M.

			Character -istic Sets			**SemSets**			**SchemEX**			**TermPicker**			**SchemEX +U+I**		
	k	**Q**	1k	100k	200k	1k	100k	200k	1k	100k	200k	1k	100k	200k	1k	100k	200k
TimBL-11M	0	SQ	*na*	*na*	*na*	.94	.97	.98	.94	.97	.98	.94	.97	.98	.85	.91	.93
	1	SQ	*na*	*na*	*na*	.73	.76	.76	.73	.75	.76	.22	.25	.29	.65	.70	.71
	1	CQ	.60	.77	.78	.39	.44	.44	.39	.44	.46	.14	.23	.29	.33	.42	.45
	2	SQ	*na*	*na*	*na*	.71	.78	.78	.59	.77	.78	.19	.21	.24	.41	.69	.70
	2	CQ	.04	.18	.21	.26	.32	.33	.11	.24	.26	.04	.05	.07	.06	.18	.19
DyLDO-127M	0	SQ	*na*	*na*	*na*	.56	.57	.58	.56	.57	.58	.56	.57	.58	.44	.47	.47
	1	SQ	*na*	*na*	*na*	.49	.50	.50	.49	.49	.50	.36	.37	.37	.41	.43	.43
	1	CQ	.68	.71	.72	.31	.32	.33	.16	.17	.18	.12	.14	.15	.13	.13	.14
	2	SQ	*na*	*na*	*na*	.49	.50	.50	.49	.49	.50	.35	.36	.36	.41	.43	.43
	2	CQ	.03	.13	.15	.23	.25	.25	.08	.09	.09	.04	.04	.04	.06	.07	.08

Sets has the lowest summarisation ratio. Although it ignores type information, the index is worth investigating since properties are highly relevant to identify bibliographic metadata (Jett et al., 2016).

Approximation quality We show in Table 8 the approximation quality in terms of F1-score for the five indices with chaining parameter $k \in \{0,1,2\}$. Please note, for indices with a chaining parameter $k = 0$ the simple queries and the complex queries have the same complexity. Please also note, Characteristic Sets ignore connected resources (see Fig. 8c). Thus, simple queries are not available (na), since they query for parameterised Object Cluster. Furthermore, the complex queries for Characteristic Sets with $k = 1$ are comparable to the simple queries for the remaining queries, since the Characteristic Set is modeled using one simple schema element.

From the results of our experiments, we can state that simple queries consistently show higher F1-scores than complex queries. This is not surprising since less information is required to satisfy the information need. However, one should note that simple queries for three out of four indices computed on the TimBL-11M dataset encounter a decrease in the F1-score for $k = 2$ compared to $k = 1$, while none of the indices has a decreased F1-score on the DyLDO-127M dataset. When computing complex index structures, more schema elements have to be computed for the active window. Thus, it is also more likely to compute incomplete simple schema elements. This effect seems to be depending on the dataset characteristic. The crawling strategy limits the path lengths in the DyLDO-127M dataset. Thus, when comparing $k = 1$ and $k = 2$, all indices computed over the DyLDO-127M dataset also encounter less loss on the F1-score compared to the TimBL-11M.

TermPicker is the only index having a higher F1-score on the DyLDO-127M dataset. As described previously, TermPicker is the only index considering no property paths but sets of properties independent for each hop. This restriction is the only difference in the schema structure compared to SchemEX. Still, TermPicker has a 50% lower F1-score than SchemEX on TimBL-11M dataset. Connecting these results to the results in the instance summarisation Table 7 brings additional insights. TermPicker has larger schema elements (in terms of the number of instances summarised) than SchemEX, due to the index configuration (compare Example 1). Since we evaluate schema elements as a whole, larger schema elements have more impact on the overall F1-score. However, Characteristic Sets have the largest schema elements, but not a generally lower F1-score. As noted above, for $k = 1$ the CQ should be compared to the SQ. SemSets only have a small drop in F1-score from $k = 1$ to $k = 2$. Additionally, there is no increase in index size from $k = 1$ to $k = 2$, but a more than ten-fold increase from $k = 0$ to $k = 1$. The same phenomenon appears for the instance summarisation ratio. These results suggest that the plain Object Cluster does not summarise instances well in both datasets, and in particular, not over 2 hops.

3.2.4 Implementation, APIs and integration

The full implementation and integration of the FLuID is ongoing work. However, the service as described in deliverable D3.1 Section 3.5.4 (Blume et al., 2017) is fully functional and can benefit from the FLuID model. The major changes are capsuled inside the indexing and the querying component. As depicted in Fig. 10, we implemented FLuID in a generic processing pipeline and we are updating our query engine LODatio+[12] to make use of all features provided by FLuID.

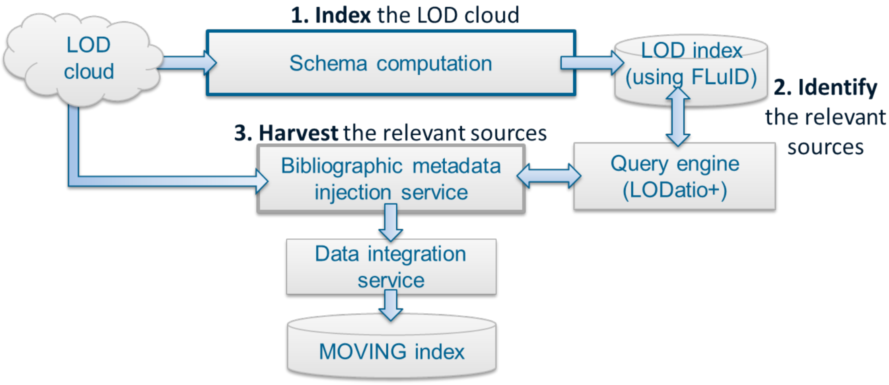


Figure 10: Integration of Linked Open Data in the MOVING platform following the FLuID model.

LODatio+ is a search engine to find relevant data sources given a structural query, and is an extension of LODatio (Gottron et al., 2013). Currently, performing generic queries on any FLuID-index is ongoing work. An exemplifying query is given in Listing 2. This example query identifies bibliographic metadata modeled with a specific set of properties and types from two different vocabularies. LODatio+ provides a user interface to develop and refine such queries. A screenshot of the user interface is provided in Fig. 11. In addition to the user interface, LODatio+ provides an API to automatically search for relevant data sources directly. The service providing this API is currently hosted at the university of Kiel [13]. For a specific search some query parameters need to be provided. Table 9 shows the possible parameters to create queries.

Listing 2: SPARQL schema level query to search for bibliographic metadata on LODatio.

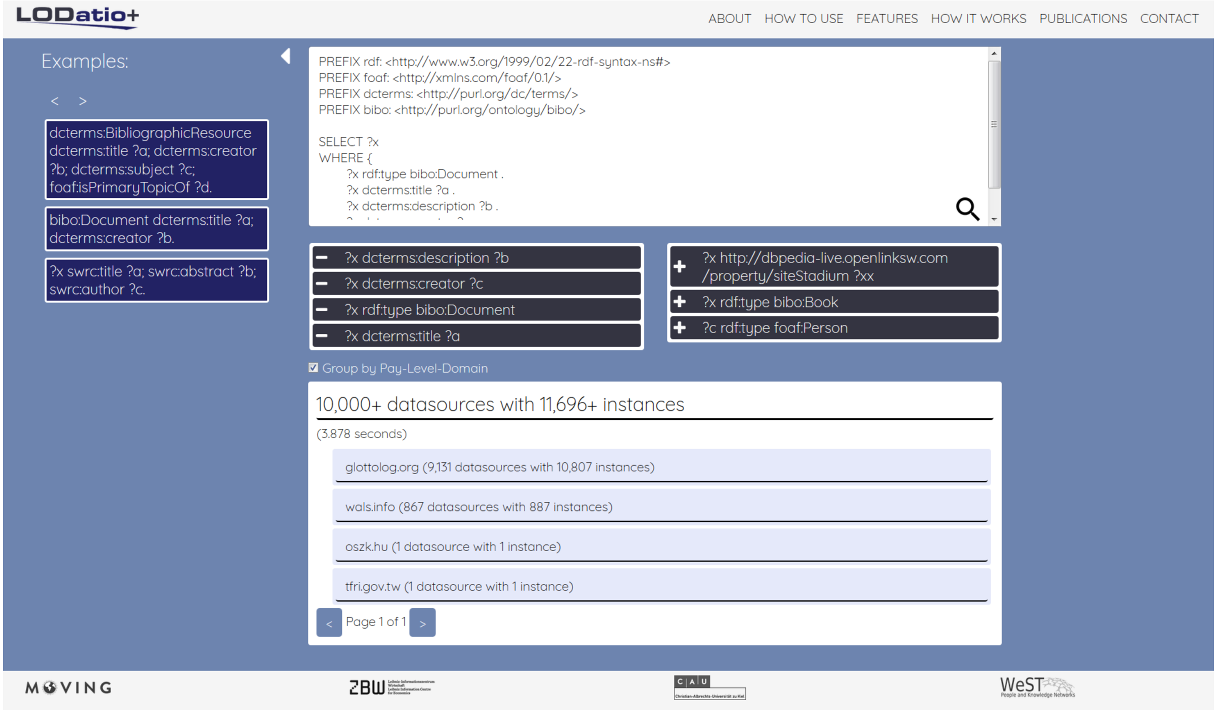
```
SELECT ?x
WHERE {
    ?x rdf:type bibo:Document .
    ?x dcterms:title [] .
    ?x dcterms:description [].
    ?x dcterms:creator [] .
}
```

Table 9: LODatio+ query parameters marked as mandatory (m) or optional (o). LODatio+ is a major component of the bibliographic metadata injection service.

Parameter	Description	Mandatory?
format	only csv is currently supported, i. e., *format=csv*	m
ordered	only un-ordered is currently supported, i. e., *ordered=false*	m
repository	Each index is stored in an individual repository, e. g., *repository=jena_virtuoso_on_kdsrv*	m
query	a SPARQL query expressing the information need	m
rangeStart	query offset	o
rangeEnd	query limit	o

With the new API of LODatio+ integrated, the bibliographic metadata injection service can retrieve csv files containing relevant data sources. Each datasource is then harvested individually and all relevant instance

[12] http://lodatio.informatik.uni-kiel.de/
[13] http://lodatio.informatik.uni-kiel.de/LodatioLogic/MultiMapper



dcterms:BibliographicResource dcterms:title ?a; dcterms:creator ?b; dcterms:subject ?c; foaf:isPrimaryTopicOf ?d.

bibo:Document dcterms:title ?a; dcterms:creator ?b.

?x swrc:title ?a; swrc:abstract ?b; swrc:author ?c.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX bibo: <http://purl.org/ontology/bibo/>

SELECT ?x
WHERE {
      ?x rdf:type bibo:Document .
      ?x dcterms:title ?a .
      ?x dcterms:description ?b .
```

- ?x dcterms:description ?b
- ?x dcterms:creator ?c
- ?x rdf:type bibo:Document
- ?x dcterms:title ?a

+ ?x http://dbpedia-live.openlinksw.com/property/siteStadium ?xx
+ ?x rdf:type bibo:Book
+ ?c rdf:type foaf:Person

☑ Group by Pay-Level-Domain

10,000+ datasources with 11,696+ instances

(3.878 seconds)

glottolog.org (9,131 datasources with 10,807 instances)

wals.info (867 datasources with 887 instances)

oszk.hu (1 datasource with 1 instance)

tfri.gov.tw (1 datasource with 1 instance)

< Page 1 of 1 >

Figure 11: A screenshot of the LODatio user interface.

information is collected. The user still needs to provide a mapping configuration. Such a mapping defines which instance information is used for which attribute within our common data model. Changing to FLuID does not prevent this.

In summary, the harvesting component is able to retrieve bibliographic metadata modeled as Linked Open Data and transform it into our common data model. Subsequently, the data is ingested into our common database (Elasticsearch[14]), which is then to be de-duplicated and disambiguated for further use.

3.3 Evolution of vocabulary terms in knowledge graphs

As explained in Section 3.2, to enhance the content in the MOVING platform, we continuously harvest bibliographic metadata from the Web, notably from the Linked Open Data cloud[15]. As metadata representation may change over time it is important to study how these changes are adopted and which impact have on the data in order to adapt the harvesting technology to deal with those changes, e.g. to avoid to re-index the data from scratch every time they are crawled, but incrementally update the index.

3.3.1 Problem statement

Vocabulary terms define the schema of Knowledge Graphs (KGs), such as the Linked Open Data (LOD) cloud or Wikidata. After ontology engineers built and published the first version of a vocabulary, the terms are subject to changes to reflect new requirements or shifts in the domains the vocabularies model. So far it is unknown how such vocabulary changes are reflected by the KGs that are using their terms. Data publishers may not be aware that changes on the vocabulary terms happened since it occurs rather rarely (Käfer, Abdelrahman, Umbrich, O'Byrne, & Hogan, 2013). Explicitly triggering data publishers to update their model is also challenging due to the distributed nature of KGs such as the LOD cloud. At the same time, although data publishers may be interested in being notified when certain vocabulary term changes happen, they lack proper tools and services to track whether and what kind of changes on vocabulary terms happened. Likewise, ontology engineers lack tools that reflects the adoption status of their vocabularies and changes on the defined terms.

To address this issues, we study the evolution of vocabulary terms in KGs. Specifically, we address three research question: *(1) When are the newly created terms of vocabularies adopted in KGs? (2) What is the use rate of classes and properties for a set of vocabularies in each dataset? (3) Are the deprecated terms*

[14] https://www.elastic.co/products/elasticsearch, see deliverables D4.1 (Gottfried, Grunewald, et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017) for additional information

[15] http://lod-cloud.net/

still used in KGs? Considering these questions, we analysed various vocabularies to better understand their changes and how they are adopted in evolving KGs.

We study the changes on these vocabularies to clarify which terms changed, the type of change, and if those changes affect terms defined in the vocabularies or terms imported from other vocabularies. We considered the two the basic types of changes: addition and deletion. Any other change, e. g. a modification, can be expressed by these two basic changes. Formally, we understand a vocabulary V as a set of terms T. A term T is either a class C or a property P. A set of terms relates to a vocabulary as $T(V) = C(V) \bigcup P(V)$. Changes in a vocabulary V are changes on its terms, i. e., the classes and properties. Data that uses classes and properties of a changed vocabulary should be updated accordingly. We rely on three dataset: Dynamic Linked Data Observatory (DyLDO) (Käfer, Umbrich, Hogan, & Polleres, 2012), Billion Triples Challenge (BTC)[16], and Wikidata[17]. The first includes weekly snapshots for a set of Linked Data resources from the LOD cloud. The second is a collection of data crawled from the LOD cloud in 2009, 2010, 2011, 2012, and 2014. The third is a collaboratively edited knowledge base.

Our analysis shows that changes have a large impact on the real data although they are not too frequent. Furthermore, we found that most of the newly coined terms are adopted in less than one week after their publishing date. However, there are terms that are only adopted after several months or up to a few years after the date of creation. Moreover, there are also some terms adopted before their official publishing date. Often, deprecated terms are still in use in KGs. Therefore those terms are not really deprecated in practice. Finally, we found that the percentages of the actually unused terms in KGs are more than 50% for most vocabularies, especially in the BTC dataset.

We think that this study can help ontology engineers and data publishers in updating their ontologies and datasets. Providing a service to notify changes on ontologies can simplify the update of vocabulary and datasets, as well as foster the adoption of new terms.

Related work In terms of analysing the use of structured data on the web, some works focused on *schema.org*. Meusel et al. (Meusel, Bizer, & Paulheim, 2015) analysed its evolution and adoption. They made a comparison of the use of *schema.org* terms over four years by extracting the structured data from the web pages that use this vocabulary from *WebDataCommons* Microdata datasets[18]. They discovered that not all terms in *schema.org* are used and deprecated terms are still used, as it is also illustrated in this work. Furthermore, they found that publishing new types and properties is preferred over using *schema.org*'s extension mechanism. Guha et al. (Guha, Brickley, & Macbeth, 2016) investigated the use of the *schema.org* in the structured data of a set of web pages. They analysed a sample of 10 billion web pages crawled from Google index and *WebDataCommons* and found that about 31 % of those pages had some *schema.org* elements and estimated that around 12 million websites are using *schema.org* terms. They did not consider the changes in vocabulary terms. Additionally, they were limited to one vocabulary only, while we consider more than one. Mihindukulasooriya et al. (Mihindukulasooriya, Poveda-Villalón, García-Castro, & Gómez-Pérez, 2016) conducted a quantitative analysis for studying the evolution of DBpedia[19], *Schema.org*[20], the Provenance Ontology (PROV-O)[21], and the Friend of a Friend (FOAF) vocabulary[22]. They proposed some recommendations such as dividing large ontologies into modules to avoid duplicates when adding new terms and adding provenance information beside the generic metadata when the change occurred.

Other works exploited Dynamic Linked Data Observatory (DyLDO) to study the use of vocabularies. Dividino et al. (R. Dividino, Scherp, Gröner, & Grotton, 2013) analysed how the use of RDF classes[23] and properties on the LOD cloud changed over time. They studied the combination of classes and properties that describe a resource but did not investigate whether a vocabulary and its terms have changed. The authors applied their analysis on a dataset of 53 weekly snapshots from the DyLDO dataset, as it is also done in this work. Over six months, Käfer et al.(Käfer et al., 2013) observed the resources retrieved from the DyLDO dataset they crawled. They analysed those resources using different factors, their lifespan, their availability and their change rate. They also examined the RDF content that has frequently changed (triple added or removed) and observed how links between resources evolve over time. While their study is important for various areas such as smart caching, link maintenance, and versioning, it does not include information about adopting new and deprecated terms.

[16] http://challenge.semanticweb.org/, last accessed: November 29, 2017

[17] https://www.wikidata.org, last accessed: November 29, 2017

[18] http://webdatacommons.org/, last accessed: October 10, 2017

[19] http://dbpedia.org/

[20] http://schema.org/

[21] http://www.w3.org/TR/prov-o/

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

[23] http://www.w3.org/standards/techs/rdf

Gottron et al. (Gottron, Knauf, & Scherp, 2015) provided an in-depth analysis of the LOD schema information in three different levels by analysing the Billion Triples Challenge (BTC) 2012 dataset. The first level focus on the unique-subject Uniform Resource Identifiers (URIs) and studied the dependency relations between the classes and their properties. They found a redundancy between classes and the attached properties. The second level considered the Pay-Level domains (PLDs) by dividing the BTC 2012 dataset into individual PLDs. They found that 20 % of the PLDs can be ignored without losing the graph explanation. The third level analysed the vocabularies based on the PLD level. They stated that data publishers either made a strong schematic design, or they apply a combination between a set of vocabularies to model their data.

Finally, some works studied the use of vocabularies with other sources. Vandenbussche et al. (Vandenbussche, Atemezing, Poveda-Villalón, & Vatant, 2017) published a report that describes Linked Open Vocabularies (LOV). It provides statistics about LOV and its capabilities such as the total number of classes and properties and the top-10 searched terms but does not include information about adopting new terms and from which Pay Level Domains (PLDs). Rathachari et al. (Chawuthai, Takeda, Wuwongse, & Jinbo, 2016) proposed a model that facilitates the understanding of organisms. Their model presents the changes in taxonomic knowledge in RDF form. The proposed model acts as a history tracking system for changing terms but gives no information about how and when the terms are used, and which PLDs adopted changed terms. Schaible et al. (Schaible, Gottron, & Scherp, 2014) published a survey of the most preferred strategies for reusing vocabulary terms. The participants were 79 Linked Data experts and practitioners and were asked to rank several LOD modeling strategies. The survey concluded that terms that are widely used are preferred. Furthermore, the frequency of use of vocabularies is a more important factor for reuse than the frequency of a single term (ignoring the frequency of the vocabulary where the term belongs to). Their survey can help to understand why there are some terms frequently used and why some of them are not used at all.

3.3.2 Method description

Our analysis method consists of two steps. First, we determined vocabularies that have more than one published version on the web. Second, we investigated how the changed terms of vocabularies are adopted and used in the evolving KGs. For the first step, we relied on Schmachtenberg et al. (Schmachtenberg, Bizer, & Paulheim, 2014) who published a report with detailed statistics about a large-scale snapshot the LOD cloud. The snapshot comprises seed URIs from the datahub.io dataset[24], the BTC 2012 dataset[25], and the public-lod@w3.org mailing list[26]. We selected a set of vocabularies that satisfy the following set of conditions and characteristics:

1. The vocabulary have at least two versions published on the web to make a comparison between them.

2. These two versions are covered by the dataset that we investigate. For example, for the DyLDO dataset, one version of the vocabularies that have been published after May 6th, 2012 is required. This is needed since at this date the first snapshot of the DyLDO dataset has been crawled.

3. The vocabulary terms are directly used for modeling some data, i. e., the vocabulary terms occur in at least one triple in the published dataset. In contrast, vocabularies could also be just linked from a data publisher, where changes of external vocabularies may not have any impact on the published data.

Based on these criteria, we examined 134 of the most used vocabularies listed in the state of the LOD cloud 2014 report by Schmachtenberg et al. (Schmachtenberg et al., 2014). We found 18 vocabularies that have more than one version. From them, 13 vocabularies have changes (additions or deprecations) on terms created by the ontology engineers of those vocabularies in the timeframe of the considered datasets. We downloaded the different versions of the extracted vocabularies using the Linked Open Vocabularies (LOV) observatory[27]. We extracted the changes between every two successive versions of a vocabulary by using Protégé 4.3.0[28]. The vocabularies selected are listed in Table 10, which also provides the number of versions considered for each vocabulary and the total number of changes (additions and deletions) occurred. Considering all the vocabularies and all their versions the total number of terms studied is 936.

[24] http://datahub.io/group/lodcloud, last accessed: October 10, 2017
[25] http://km.aifb.kit.edu/projects/btc-2012/, last accessed: October 10, 2017
[26] http://lists.w3.org/Archives/Public/public-lod/, last accessed: October 10, 2017
[27] http://lov.okfn.org/dataset/lov, last accessed: October 10, 2017
[28] http://protege.stanford.edu, last accessed: October 10, 2017

Table 10: Overview of the vocabularies and their changes.

Vocabulary	Versions	Changes
Asset Description Metadata Schema (ADMS)[29]	2	18
Citation Typing Ontology (CiTO)[30]	3	218
The data cube vocabulary (Cube)[31]	2	6
Data Catalog Vocabulary (DCAT)[32]	2	13
A vocabulary for jobs (emp)[33]	2	1
Ontology for geometry (geom)[34]	2	2
The Geonames ontology (GN)[35]	7	31
The music ontology (mo)[36]	2	46
Open Annotation Data Model (oa)[37]	2	31
Core organization ontology (org)[38]	2	8
W3C PROVenance Interchange (Prov)[39]	5	168
Vocabulary of a Friend (voaf)[40]	4	8
An extension of SKOS for representation of nomenclatures (xkos)[41]	2	1

Subsequently, we investigated how changed vocabulary terms are used in the evolving KGs. We extracted all PLDs from the crawled triples that use the terms from the 13 vocabularies above relying on the Guava[42] library version 16.0.1. Besides the date of the first appearance of a vocabulary term, we also recorded the number of triples that contain the term. This information is then used to compute the adoption time of vocabulary term changes over the dataset snapshots.

We applied our analysis approach on three large-scale KGs. The first two are DyLDO and BTC and are obtained from the Linked Open Data cloud, and the third is Wikidata. Below, we briefly characterise the datasets. DyLDO is a repository to store weekly snapshots from a subset of web data documents (Käfer et al., 2012). For our study, we parse 242 snapshots (from May 2012 until March 2017). BTC is yearly crawled from the LOD cloud from 2009 to 2012, as well as in 2014. We used all available versions to analyse the adoption of the extracted vocabularies in our study. Wikidata[43] is a knowledge base to collaboratively store and edit structured data. To analyse the Wikidata vocabulary, we first extracted the terms introduced by this vocabulary. Specifically, through the RDF Exports from Wikidata page[44], we parsed the terms and properties from the RDF dump files that were generated using the Wikidata toolkit[45]. We assumed that the first snapshot of those files is the first version of the Wikidata vocabulary, and based on this assumption we parsed the next dump files to extract the changes to the first version, and so on. Relying on the 25 RDF dump files (from April 2014 until August 2016), we extracted the terms that are added or deprecated. Then, we parsed those files to extract the adoption of terms to analyse the adoption behaviour for the Wikidata vocabulary's terms.

3.3.3 Analysis

3.3.3.1 Results

In this section, we summarise our findings based on the conducted experiments. We first present the results of vocabulary changes, their use and adoption in the LOD Cloud, then we outline the same findings for Wikidata.

[29] https://www.w3.org/TR/vocab-adms/, last accessed: November 10, 2017
[30] http://www.sparontologies.net/ontologies/cito/source.html, last accessed: November 10, 2017
[31] http://www.w3.org/TR/vocab-data-cube/, last accessed: November 10, 2017
[32] https://www.w3.org/TR/vocab-dcat/, last accessed: November 10, 2017
[33] http://lov.okfn.org/dataset/lov/vocabs/emp, last accessed: November 10, 2017
[34] http://data.ign.fr/def/geometrie/20160628.htm, last accessed: November 10, 2017
[35] http://www.geonames.org/ontology/documentation.html, last accessed: November 10, 2017
[36] http://www.geonames.org/ontology/documentation.html, last accessed: November 10, 2017
[37] http://www.openannotation.org/spec/core/, last accessed: November 10, 2017
[38] https://www.w3.org/TR/vocab-org/, last accessed: November 10, 2017
[39] https://www.w3.org/TR/prov-o/, last accessed: November 10, 2017
[40] http://lov.okfn.org/vocommons/voaf/v2.3/, last accessed: November 10, 2017
[41] http://rdf-vocabulary.ddialliance.org/xkos.html, last accessed: November 10, 2017
[42] https://github.com/google/guava/, last accessed: October 10, 2017
[43] https://www.wikidata.org/, last accessed: November 29, 2017
[44] http://tools.wmflabs.org/wikidata-exports/rdf/exports.html, last accessed: November 29, 2017
[45] https://github.com/Wikidata/Wikidata-Toolkit, last accessed: November 29, 2017

The LOD cloud

Changes in LOD vocabularies We studied the changes of terms in the vocabularies, focusing on creation and deprecation. Overall we observed 35 % of newly created terms and 11 % of deprecated ones. Figure 12 shows the 13 vocabularies in our study and the total number of classes and properties for each version of those vocabularies. Please note that most of them have an increased number of terms, but there are two exceptions: the number of classes and properties is decreased for the *ADMS* vocabulary, while *CiTO* had a huge decline in the number of classes.

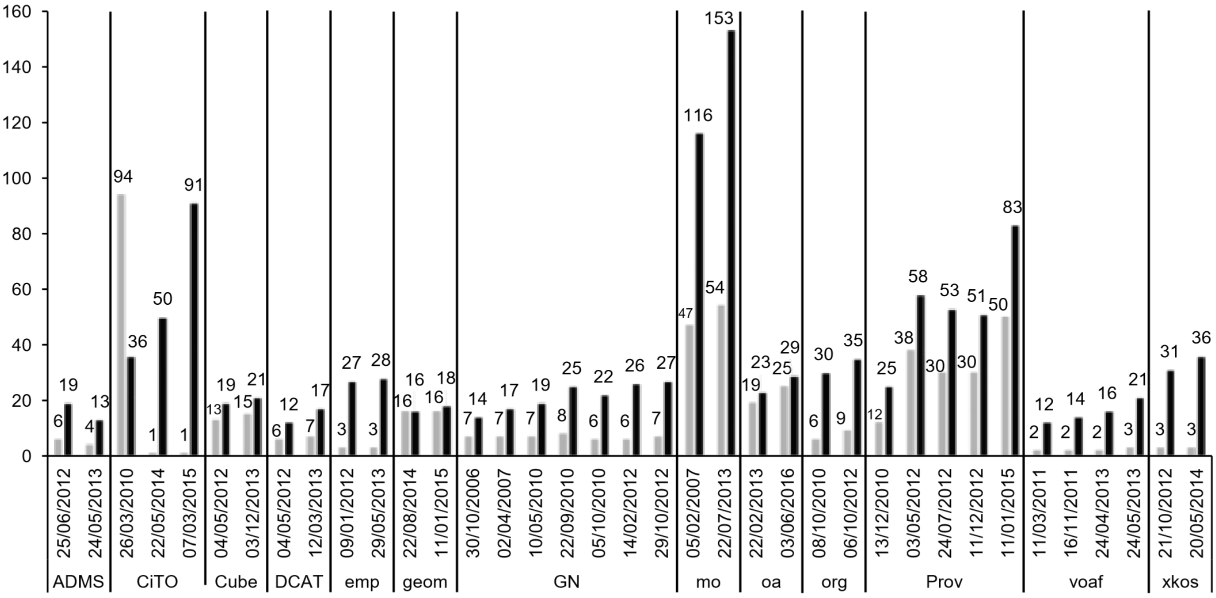


Figure 12: Total number of classes (gray bar) and properties (black bar) for the selected vocabulary versions.

During our analysis, we noticed that some of the deprecated properties were reintroduced later. These recreated terms belongs to the *CiTO* and *GN* vocabularies. The former deprecated 18 properties in May 2014 (introduced in March 2010), which reappeared in the version that was published in March 2015, i. e. after around ten months. The latter also recreated three deprecated properties: `alternateName` (creation: October 2006, deprecation: September 2010, recreation: February 2012), `name` (creation: October 2006, deprecation: September 2010, recreation: October 2010), and `shortName` (creation: September 2010, deprecation: May 2010, recreation: February 2012). *GN* reintroduced two out of three deprecated terms after about 17 months and one shortly after (13 days).

Use of LOD vocabularies We analysed the use of the selected vocabularies by considering triples which contains one of their terms in the predicate and/or the object position and a PLD in the subject. *Geonames.org* is the PLD that uses most terms of the selected vocabularies in the BTC 2009 and 2010 datasets (Table 11). In BTC 2011 and 2012, *zitgist.com* and *rdfize.com* are the most frequent PLDs, while in BTC 2014 and DyLDO, *dbtune.org* accounts for the highest use. However, the number of triples in BTC 2009, 2011, and 2012 is significantly lower than for the other datasets.

Table 11: PLDs with the highest use of terms from the selected vocabularies for each of the datasets.

Dataset	PLD	Triples
BTC 2009	geonames.org	81.0M
BTC 2010	geonames.org	7.0M
BTC 2011	zitgist.com	2.6M
BTC 2012	rdfize.com	3.8M
BTC 2014	dbtune.org	81.5M
DyLDO	dbtune.org	160.0M

Table 12 shows the PLDs that used most of the terms of each vocabulary in the BTC and DyLDO datasets. For most vocabularies, we notice that there are no variety in the two dataset, i. e., the same PLD usually accounts for the highest use of terms in both BTC and DyLDO. Table 13 outlines the number of triples which use terms from the 13 vocabularies considered. We can observe that *geom* does not appear in all the BTC datasets, while *emp* and *oa* vocabularies almost did not appear in the BTC and DyLDO datasets.

Table 12: PLDs with the highest use of terms for each vocabulary considered. The amount represent the total numbers of triples that the corresponding PLD appears in.

Vocabulary	**BTC**		**DyLDO**	
	PLD	Amount	PLD	Amount
ADMS	w3.org	5.0K	w3.org	253.0K
CiTO	ontologycentral.com	3.3M	ontologycentral.com	4.7M
Cube	ontologycentral.com	1.2M	ontologycentral.com	1.8M
DCAT	ontologycentral.com	5.0M	ontologycentral.com	4.8M
emp	purl.org	11.0K	purl.org	14.0K
geom	-	-	ign.fr	5.5K
GN	geonames.org	90.0M	geonames.org	40.0M
mo	dbtune.org	84.0M	dbtune.org	76.0M
oa	w3.org	7.0K	w3.org	13.0K
org	data.gov.uk	21.0K	w3.org	111.0K
Prov	w3.org	62.0K	dbpedia.org	1.0M
voaf	purl.org	381.0K	purl.org	9.0K
xkos	270a.info	609.0K	270a.info	4.0K

Table 13: Number of triples in the DyLDO and BTC datasets that use terms from the 13 vocabularies considered.

Vocabulary	**BTC09**	**BTC10**	**BTC11**	**BTC12**	**BTC14**	**DyLDO**
ADMS	12	2	4	26	7.0K	337.0K
CiTO	0	0	4	4.5K	305.0K	1.0M
Cube	0	0	40.0K	8.4K	56.0M	12.0M
DCAT	0	0	12	4.5K	317.0K	9.8M
emp	0	0	0	0	238	26.0K
geom	0	0	0	0	0	5.7K
GN	81.0M	7.4M	477.0K	441.0K	1.0M	55.0M
mo	2.0M	1.7M	12.0M	4.0M	102.0M	83.0M
oa	0	0	0	0	192	23.0K
org	0	9	129	11.0K	20.0K	700.0K
Prov	0	9	43	902	3.0M	4.0M
voaf	0	0	2	0	4.0K	1.0M
xkos	0	0	0	0	610.0K	194.0K

In DyLDO dataset, the use of most vocabularies is steady. Figure 13 depicts the vocabularies with a varying use. Notably, *mo* shows increasing and declining intervals, *Prov* is increasing in popularity despite some slight negative picks, while *ADMS* had a significant drop in 2015 after an initial increasing use, although it seems slightly increasing. Furthermore *Cube* had a peak towards the end of 2015 to then come back to its initial use rate, while *emp* seems no more used from 2015.

The great majority of the deprecated terms (87 %) are still used after deprecation. We found that *geonames.org* is the PLD that most frequently uses deprecated terms in the BTC and DyLDO datasets. For instance, Figure 14 illustrates the use of the term `gn:Country` in the DyLDO dataset, which was deprecated in September 2010. Despite various fluctuations, its use increased until August 2015, then declined and increased again to reach a peak in August 2016.

Adoption of LOD vocabulary changes The majority of the newly created terms was adopted in less than 10 days, as showed in Table 14. The triples column represents the total number of triples in DyLDO dataset which contains the adopted terms, while μ and σ are the average number of days before adoption and the standard deviation, respectively. Additionally, adopting *geom* and *GN* terms took long time.

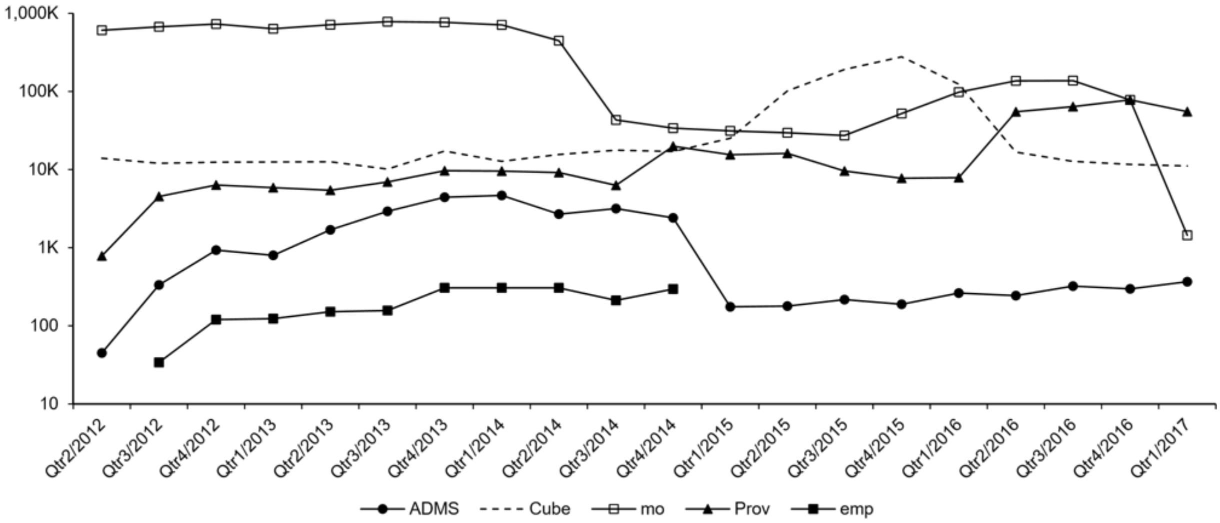


Figure 13: The mean number of triples that use terms for a subset of the vocabularies considered by DyLDO snapshots aggregated in quarters.

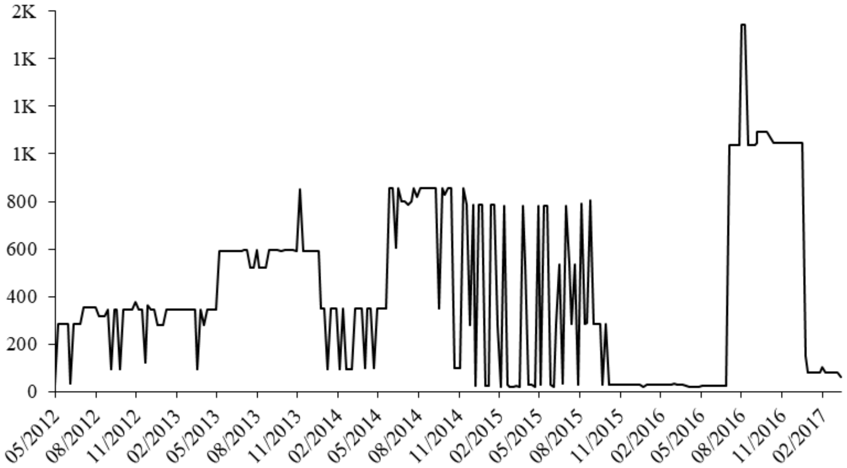


Figure 14: The use of gn:Country class in the DyLDO dataset.

After being adopted, 50 % of the newly created terms decreased in use during the considered period, 47 % showed a steady use, while 3 % increased. For example, during its evolution, the *voaf* vocabulary created 10 new terms. All but one of those have a decline in the use (Figure 15). The figure shows only six terms as the remaining are exploited in much fewer triples. In general, a similar trend holds for all the vocabularies. More details about the adoption time of other vocabularies are available in an extended technical report (Abdel-Qader & Scherp, 2017).

Not all terms are adopted. For example, we found that the percentage of adoption for half of the vocabularies is less than 50 % of terms in the BTC dataset (in total, 50 % of all terms were not used). In DyLDO, the percentage of unused terms of all vocabularies was 23 %, and only one vocabulary (*CiTO*) adopted 60 % of the terms, while the remaining vocabularies less than 40 % (Table 15). Notably, the 21 new terms of the *oa* vocabulary and the only *xkos* term are never adopted.

Wikidata After parsing the terms and properties from the RDF dump files for the period from April 2014 until August 2016, we have extracted the added and deprecated terms of the Wikidata vocabulary. Figure 16 presents the total number of classes and properties in each Wikidata snapshot, which grows to reach 11 classes and 27 properties in August 2017. Notably, there are no terms that are deprecated during the ontology evolution.

For the Wikidata vocabulary, ontology engineers added 3 classes and 9 properties during the analysed period. The new classes are DeprecatedRank, PreferredRank, and NormalRank, while the new properties are propertyTypeMonolingualText, propertyTypeProperty, propertyQualifierLinkage, propertyRefe-

Table 14: The adoption of newly created terms for each of the vocabularies.

Vocabulary	New terms	Adopted terms	Triples	μ	σ
ADMS	6	100 %	31K	7.00	0.00
CiTO	80	100 %	281K	7.00	0.00
Cube	5	100 %	15K	7.00	0.00
DCAT	5	100 %	104K	8.40	3.13
emp	1	100 %	4K	7.00	0.00
geom	2	100 %	16K	420.00	0.00
GN	21	100 %	160M	127.76	255.33
mo	44	100 %	45M	8.75	9.68
oa	21	0 %	-	-	-
org	8	100 %	173K	7.00	0.00
Prov	106	85 %	121M	30.15	37.49
voaf	10	100 %	75K	43.33	68.58
xkos	1	0 %	-	-	-

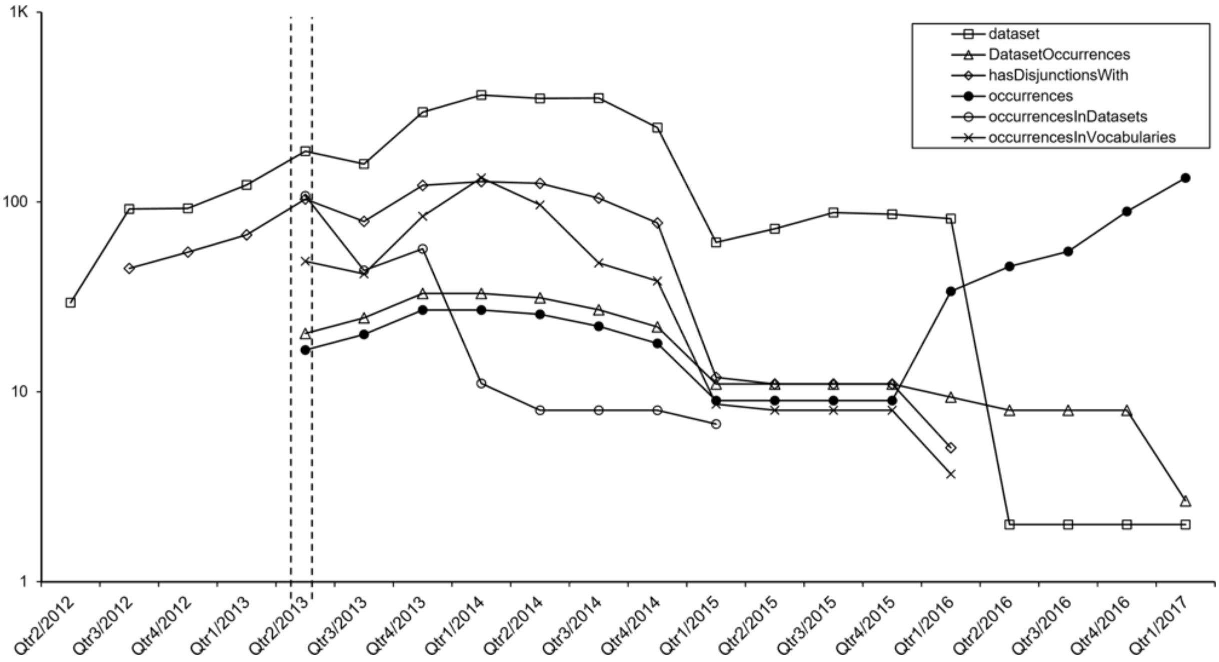


Figure 15: The use (amount of triples in which a term occurs) of the *voaf's* newly created terms by quarters of DyLDO snapshots. The vertical dashed lines represent the publishing time of new versions of the vocabulary. Please note that two versions of *voaf* have been published before the first snapshot of DyLDO (i. e. `dataset` and `hasDisjunctionsWith` are newly created in versions released before the second quarter of 2012).

`renceLinkage`, `propertySimpleClaim`, `propertyStatementLinkage`, `rank`, `propertyValueLinkage`, and `quantityUnit`.

Figure 17 illustrates the use of newly created classes and properties. Only 5 out of 12 terms are adopted. `NormalRank` and `rank` are much more used than the other new terms. Furthermore, the actually adopted terms among all the newly created ones are adopted directly after their creation date.

3.3.3.2 Discussion

We found that not all vocabulary changes are reflected in the data in knowledge graphs, and there is a need for a service or tools to track vocabulary changes. Such service helps ontology engineers and data publishers in updating their ontologies and models. We firstly discuss the results related to the LOD Cloud, then the results of changes and adoption of the Wikidata terms.

The LOD cloud

Table 15: The percentage of unused terms in the BTC and DyLDO datasets.

Vocabulary	Total terms	BTC	DyLDO
ADMS	31	68 %	3 %
CiTO	220	72 %	60 %
Cube	37	35 %	0 %
DCAT	23	48 %	9 %
emp	31	87 %	6 %
geom	34	100 %	3 %
GN	43	26 %	9 %
mo	208	36 %	2 %
oa	63	83 %	35 %
org	44	20 %	11 %
Prov	143	22 %	24 %
voaf	24	33 %	8 %
xkos	35	63 %	14 %

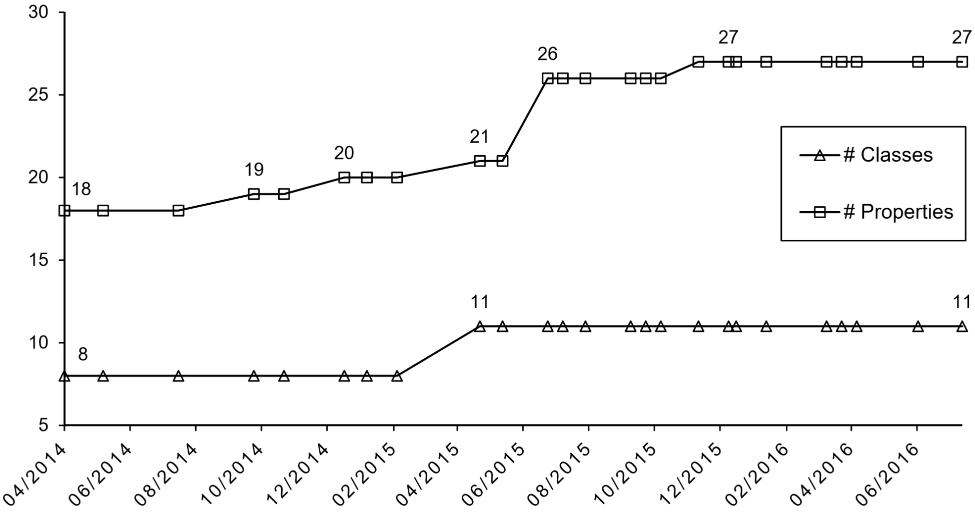


Figure 16: Total number of classes and properties of the Wikidata vocabulary per RDF dump file.

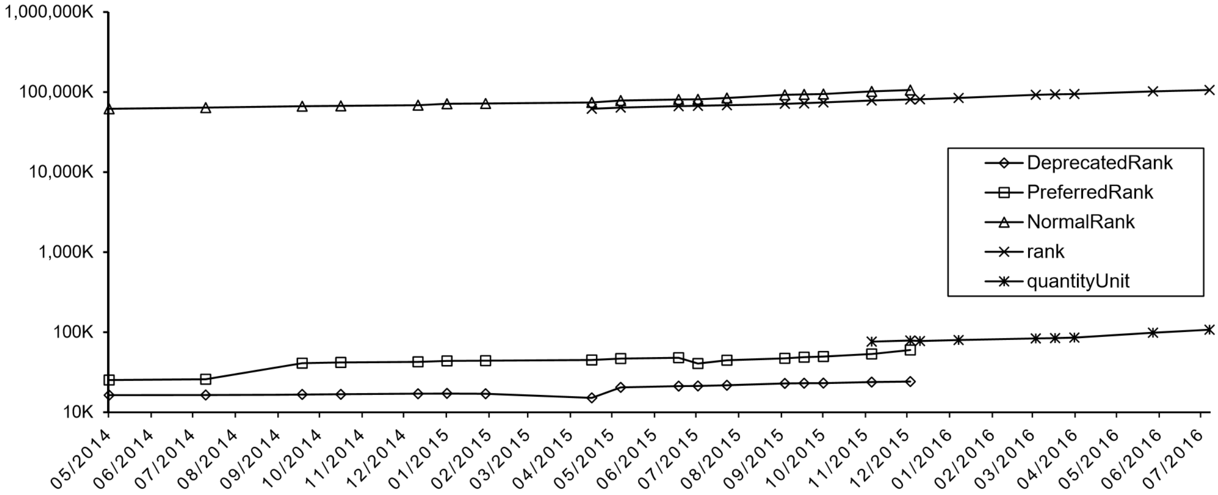


Figure 17: The amount of triples that the adopted newly created classes and properties of the Wikidata vocabulary after parsing Wikidata RDF dump files.

Changes in LOD vocabularies The number of additions and deprecations of terms is small. This is in line with existing studies (Abdel-Qader & Scherp, 2016; Guha et al., 2016; Meusel et al., 2015). However those changes may have a large impact on the data of KGs. For example, the new version of the *oa* vocabulary caused a significant increased of its use: the triples containing its terms almost triplicates (from roughly 400 hundred to over 1100). In general, the changes impact on the use either in an increasing or decreasing way (6 and 5 out of 13 vocabularies, respectively), although with varying time. In the case of *DCAT* there is an increase so delayed in time (3 years) which is probably unrelated to the new version. More details are available in our extended technical report (Abdel-Qader & Scherp, 2017).

Most of the vocabularies increased in the total number of classes and properties. This suggests that more knowledge is represented in the LOD cloud, requiring new terms. One exception is *CiTO*, which consisted of 94 classes and 36 properties when initially published. The second version counted only one class and 50 properties. Specifically, all the 94 classes were replaced with the new class *CitationAct* and most of the 36 properties of the first version were substituted. The third version provided 91 properties, although 18 of the new properties were reintroduced from the first version (deprecated in the second and recreated in the third). In practice, almost a new ontology was built. This is particularly important since *CiTO* has grown much in popularity (BTC 2014 contained over 300 thousand triples compared to 40 thousand in BTC 2011).

New versions of vocabularies, together with the great variety of vocabularies already existing, and the new ones may overwhelm ontology engineers, which need to choice among a vast amount of alternatives when building or updating their ontologies. Similar issues may occur to data publishers when deciding which vocabularies to exploit in their datasets. Missing some changes and consequently not update an ontology or a dataset is likely (see the following discussion on the use of terms), notably in a distributed environment as the LOD cloud. This holds particularly for deprecation. Tools to notify ontology engineers and data publishers are lacking of these changes as well as new vocabularies are lacking. While these systems can ease the maintenance of vocabularies and datasets, more advanced one could also recommend terms and vocabularies according to the specific needs of their users. The insights provided in this study can be beneficial to build such tools.

Use of LOD vocabularies Cross-domain (*Prov*, *voaf*, *DCAT*, and *ADMS*) and Geographic (*Cube* and *GN*) vocabularies were the most popular among data publishers. Some of them are exploited by few PLDs. For instance, *w3.org* widely used the *ADMS* terms at the beginning of the investigated time-frame, while later *deri.de* accounted for the highest use of this vocabulary. On the other hand, some vocabularies have been used by various PLDs. For example, *Cube* has been employed by *ontologyCenter.com, esd.org.uk, linkedstatistics.org*, and *linkedu.en*. This may suggest that some vocabularies are applicable in multiple domains, while others are more application-specific, but should be further investigated.

Overall, *geonames.org* and *dbtune.org* are the most frequent PLDs. In the BTC 2009 and BTC 2010 datasets, *geonames.org* was the PLD that uses most of the terms. This is caused by wide use of the *GN* vocabulary in those years. Later, *dbtune.org* accounted for the highest number of triples in the BTC 2014 and DyLDO snapshots from 2012 to 2014.

Although some terms are deprecated, 87 % of them were still exploited. This is in line with (Meusel et al., 2015). *Geonames.org* is the PLD that accounts for highest number of deprecated terms. For example, in the BTC 2011 dataset, *geonames.org* used six deprecated terms in about 522 thousand triples. That number declined to three terms and roughly 181 thousand triples in BTC 2012, but increased again to 49 terms in BTC 2014 (5.5 thousand triples). It appears that data publishers have not updated their data models. A possible reason for this could be that they are not aware of changes in the vocabularies exploited. Thus, as previously discussed, they could benefit from tools to notify these changes.

Adoption of LOD vocabulary changes Most of the newly coined terms are adopted directly (in less than one week). Surprisingly, we even found some terms adopted before their official publishing date. We believe that some of the new versions of vocabularies are already online and can be used before their official announcement. In some cases, it may take time to finish the procedures to publish the new version of the vocabulary. Thus, data publishers can access the new terms before their formal release, simply because they are available online.

Although most of the terms have fast adoption time, some vocabularies, such as *GN*, took more than 120 days, in average, to adopt new terms. However, this average does not reflect the actual adoption behaviour: the new version of *GN* provides 21 new terms, 17 terms are adopted within 7 days, while the remaining 4 terms are adopted in over 600 days. Therefore, the average result was affected by those few terms that have a vast adoption time.

Another interesting point is that some newly created terms are never adopted. For example, ontology engineers published a new version of the *oa* vocabulary in June 2016, with 21 new classes and properties. None of those terms have been adopted (at least until April 2017, the last DyLDO snapshot considered), while the first version of *oa* was published in February 2013 with 42 terms and all but one were adopted in less than

3 months. As the reasons why those terms are unused likely depend on the specific application scenario, we suggest that ontology engineers investigate these issues and possibly revise them.

Wikidata We found that the Wikidata vocabulary showed no deprecated terms, although some were never adopted during the time-frame investigated in this paper (e. g., the `Article` class). Likewise most of the LOD vocabularies, the Wikidata vocabulary counts a small number of additions (3 classes and 9 properties) and no deprecation.

Three classes (`DeprecatedRank`, `NormalRank`, and `PreferredRank`) suddenly disappeared from Wikidata statements after the snapshot in December 2015, i. e., after about 8 months of their creation in May 2015. There is a huge difference in the number of triples in which the terms occur. For instance, `NormalRank` and `Statement` classes have been used in about 106 and 81 million triples, respectively. The other classes (except `Item`) are used in less than 2.4 million triples. The same observation can be made for properties: all but `rank` appeared in less than 2.7 million triples, while `rank` accounted for approximately 62 million triples when introduced in May 2015, then reached about 106 million triples in August 2016. Evidently, those new terms were highly needed, given their wide exploitation.

Only 5 out of 12 of the newly created terms are adopted and their adoption occurs directly after their creation date. This was expected in Wikidata which is a more controlled and centralised environment than a distributed KG as the LOD cloud. Surprisingly, the majority of new terms (2 classes and 9 properties) seems not adopted in any statements of Wikidata. However a deeper analysis showed that these are used to define properties and their types, except the `Article` class, which needs further investigation.

3.4 Author alignment

Author alignment includes the task of author name disambiguation and a preprocessing step called *blocking* in which one tries to find a good partitioning of all author names (based on the names only) such that the partitions are as small as possible without separating authors that are mentioned with varying name specification. Each separate partition is then fed to the disambiguation method that uses a variety of features in order to try and discriminate different authors with a similar name. In D3.1, we have already presented a new method of author name disambiguation, which is now integrated into the MOVING platform, as described in deliverable D4.2 (Gottfried, Pournaras, et al., 2017). However, the task of author name blocking had only marginally been addressed by using a simple normalisation scheme and separating all author names by last name and first initial. This neglects valuable information like an author's full first name, if given. Thus, we focus on author name blocking in this section making the following contributions: (1) we analyze the problem setting as such and present a graph-theoretical framework that allows the formulation and evaluation of standard and more elaborate blocking schemes; (2) we propose and evaluate a new blocking scheme called *entropy-isolate* that generalises the idea behind existing schemes and allows tuning a threshold parameter; (3) our evaluations allow to estimate to which extend previous work evaluated within name blocks has neglected a blocking-related loss of recall.

3.4.1 Problem statement

In author name disambiguation (AD), a set of author names mentioned on (scientific) documents is clustered into separate real-world persons. This resolves *author name homonymy*, which occurs when different authors can share the same name. A related problem is *author name synonymy*, where the same author has different names. Usually, this problem is solved as a preprocessing step to disambiguation, by partitioning the set of all author names into subsets (blocks) while minimising the size of those blocks and the number of synonymous author mentions that are not in the same block. While there are many factors like spelling mistakes, short-forms, language-specificity or name changes that can lead to synonymous author names, we only deal with the problem of varying completeness in the specification of a name. Consider *John Doe* and *J. H. Doe* which might both refer to the same person. Both names give four pieces of information. The former shows the initial of the surname, the full surname, the initial of the first name and the full first name. The latter does not provide the full first name, but the initial of the second name. Some pairs, like *John Doe* and *J. Doe,* stand in a direct *generalise*, or parent, relation to each other. As this relation is transitive, it allows us to view all names in a directed acyclic graph where every node is a specification of its ancestors and a generalisation of its descendents, as depicted in Figure 18.

Often, synonymy is resolved by selecting a specific type of node (i. e. surname and first initial) and collecting all names of this type in a single block together with their descendents. In this context, we note that a node usually has more than one child of the same type, e. g. *John Doe* and *Jack Doe*. Sometimes, more elaborate blocking schemes are applied. Most papers on AD do not evaluate the effect of the applied

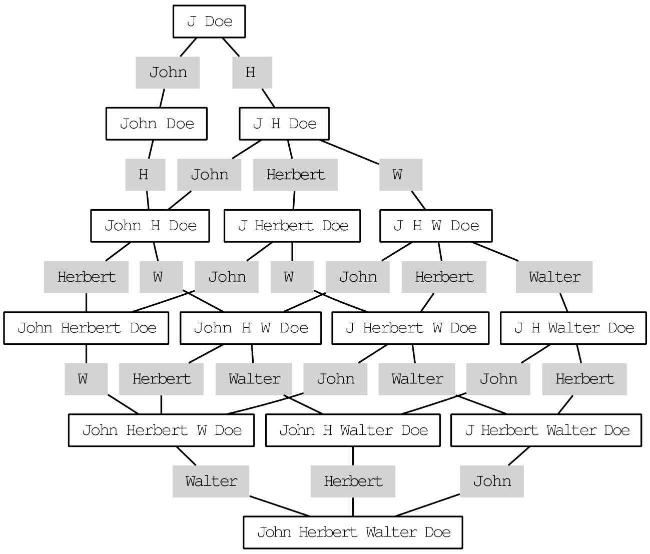


Figure 18: A subset of the graph that structures all possible name variations.

blocking scheme, or compute recall only within a block, such that mistakes introduced by the blocking scheme remain hidden (Levin, Krawczyk, Bethard, & Jurafsky, 2012). Consider the extreme case where the blocking scheme separates all names into size-one blocks. This leads to perfect precision and recall if the AD method is only evaluated within a block.

The concept of blocking in the sense of splitting a large collection into smaller portitions in order to allow for further in-depth analysis is known not only in AD: for example, Christen (Christen, 2012) compares different approaches in the context of document deduplication, Papadakis et al. (Papadakis, Svirsky, Gal, & Palpanas, 2016) in entity recognition. While we limit ourselves to simple decision rules that create *disjoint* blocks of author mentions solely based upon their names, many of the compared approaches go much further, creating non-disjoint blocks, using various features or applying machine learning techniques in order to achieve optimal results. This shows that blocking can solve a considerable part of the actual task (i.e. disambiguation), as in this case blocking already uses the tools and information that would otherwise be restricted to the actual task. Considerable effort is involved in deploying such an advanced blocking methodology. All blocking methods have in common that they avoid the quadratic complexity of pairwise comparisons, which are postponed to the actual task. In a recent unpublished contribution, Kim, Sefid and Giles (Kim, Sefid, & Giles, 2017) point out that standard blocking means conjunctive combination of boolean predicates (i.e. $hasFullLast(N,Doe) \wedge hastFirstInitial(N,J)$). Applying such standard blocking implies that a block contains exactly the set of names that generalise to a common ancestor name. This is not the case in disjoint blocking, where blocks may overlap. According to Kim, Sefid and Giles, disjoint blocks are rather inconvenient. They are usually the result of non-standard, disjunctive blocking schemes, in which names only need to satisfy one out of many conditions to be assigned to a certain block (i.e. $hasFullFirst(N,John) \vee (hasFirstInit(J) \wedge hasSecondInit(H))$). Then, one name may be assigned to multiple blocks. Standard blocking is perhaps the most popular blocking technique in AD, usually in its simplest form of separating all author mentions by *surname* and *first initial*. For example, Levin et al. (Levin et al., 2012) use it when disambiguating the Web of Science. Kim, Sefid and Giles seem to be the only group that has focused on applying advanced methods like machine learning or disjoint blocking to AD. However, sometimes author names are compared using non-boolean similarity measures, potentially identifying misspellings or short forms. Milojevic (Milojević, 2013) presents a standard blocking scheme that allows variable specificity in the block-defining name. We build mostly on his approach and elaborate further in the next section.

3.4.2 Method description

Author names can be represented as instantiations of a number of types. Using the first three first names of an author, we identify 30 different sensible combinations (see Table 16) which could all refer to the same person, but in different completeness. The most general name is the empty name (the type of name where nothing is specified) and the most specific is i.e. *John Herbert Walter Doe* (the type of name where everything is specified). A more frequent case is for example *John H. Doe* (type 11111000: full surname, full first name, initial of second name). These types can be ordered in a directed acyclic graph (see Figure 18) where each node is a name, his parents are maximally specific generalisations and his children are maximally general specifications. We note that we reach a child by either completing or adding an initial (if possible) and that the level of specificity of a name is the sum of 1's in its type. This means that we assume that it is not possible to specify a first name (by name or initial) without having indicated all of its previous first names (by name or initial). For example, we have to assume that *John W. Doe* is a different person than *John H. W. Doe*.

D	Doe	J	John	H	Herbert	W	Walter
1	1	1	1	1	1	1	1

John Herbert Walter Doe

D	Doe	J	John	H			
1	1	1	1	1	0	0	0

John H. Doe

Table 16: Two examples of author name type representations

In order to fully represent the dataset in our name graph, we store the frequency of a name in the respective node. We say a node *covers* as many names as all of his children cover. A node covers at least as many nodes as his frequency count. When we add a name as a node to the graph, we add all ancestors of the name as nodes to the graph. Then, we increase the cover count of each ancestor by the frequency of the name in a way that ensures the cover count of the empty name node always gives the number of author mentions in the data. As long as ancestor names have not been observed before, they constitute hypothetical nodes with a frequency of zero. We abstain from adding hypothetical descendents of an observed name to the graph.

Theoretically, blocking could be omitted. Nothing prevents a hypothetical, super-effective AD method from discriminating perfectly among a single giant block of author mentions. However, the computational complexity of AD – which is roughly quadratic – would make this approach infeasible, and we would also have to find a way to combine the boolean logics of generalisation and specification with the similarities of our disambiguation method. The latter could be solved in a constraint clustering setup, but constraint clustering is NP-hard. Therefore, we opted to have blocking as a preprocessing step of AD. Using the notion of the acyclic graph, we can make the following logical deductions:

1. Name specifications can be in a parent relation, e. g. *John Doe* is a direct generalisation of *John H. Doe*. The parent-relation holds for all pairs of names where the first name is a parent of the second. Remember that we introduce hypothetical unobserved parents for every node.

2. The transitive closure of the parent-relation is the ancestor-relation. The descendent-relation is the set of all pairs where the first name is an ancestor of the second.

3. The symmetric closure of the parent-relation is the parent-child relation. This relation is the union of the parent-relation and the child-relation, which is the inverse of the parent-relation.

4. The transitive closure of the parent-child-relation is the reachable-relation which contains all pairs of author names that are together in the same block.

Accounting for blocks of size one, we also apply the reflexive closure to say that every name is together with itself. Then, the reachable relation is the equivalence relation and partitions the set of author names into equivalence classes (blocks). So any pair of names for which there exists a connecting path in the graph has to be in the same block. For example if there is a *J. Doe*, a *Jack Doe* and a *John H. Doe*, then all three names are in the same block, because *J. Doe* could refer to the same person as *Jack Doe* and could also refer to the same person as *John H. Doe*. This means that based on the names, we would have to put all three of them in the same block. Given only the names themselves as indicators, this is the only way to definitely rule out any loss in recall. For example, if we assign *J. Doe* only to *Jack* we would end up with suboptimal recall if *J.* actually refers to *John* and vice-versa. To some degree, this is not practical: one unnamed author suffices to connect all names in the entire collection. For frequent names, even using surname and first initial might join too many different mentions and introduce unnecessarily low precision: why discard the information that *John Doe's* cannot be *Jack Doe's* for a few doubtful connections to *J. Doe*? In Figures 19 and 20, we show an empirical example from a subset of the Web of Science (WoS).

Blocking scheme type *field-isolate* We can observe that given the above logical assumptions, in large collections like the Web of Science (WoS) all names are somehow connected. For example *Jack Doe*, *J. H. Doe* and *John Doe* are connected by (hypothetical) nodes *Jack H. Doe* and *John H. Doe*. To limit the computational complexity, we isolate certain highly connected nodes in the graph to achieve a well-behaved partition at the cost of possibly lower recall. This means that a blocking scheme consists of the logical assumptions made above and a set S of names or nodes to be isolated. In theory, we could also remove elements of the parent-relation instead of completely isolating names, but for now we restrict ourselves to the simpler case. We identify the following obvious blocking schemes:

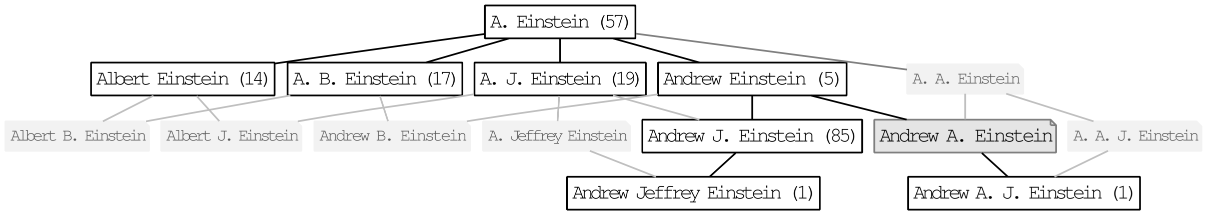


Figure 19: From the WoS: reachability is fully expressed without the shaded parts, but those can be relevant when using blocking schemes (i.e. isolating *A. Einstein*). The unshaded subgraph is a reduction by minimal spanning tree and removing useless hypothetical nodes.

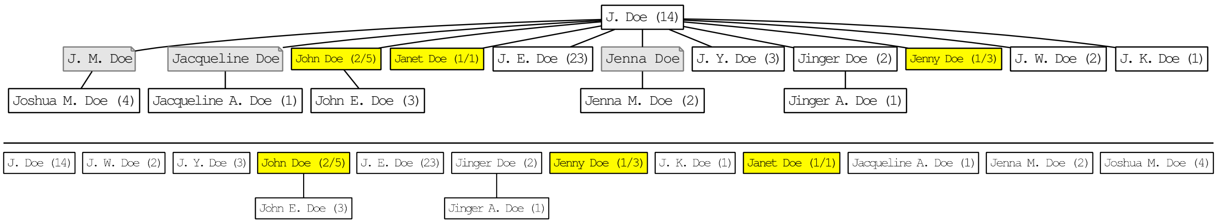


Figure 20: An example from the WoS: the initial graph consists of a single, highly connected block (here reduced). After isolating all nodes without full first name, the picture becomes clearer and we are left with twelve blocks – but we have separated names that could actually refer to the same author. Counts are in brackets; hypothetical nodes have no counts; (2/5) means that two out of five mentions are annotated with researcher-ID (rID).

1. S is the set of all names without full first name;
2. S is the set of all names without second initial.

These schemes are commonly found in the literature, although it is usually not mentioned what happens with names that are in S. We suspect that these names are often dropped, affecting the results of AD in unforeseen ways. While there is no optimal solution, in our analysis, these names form their own blocks. We call this type of scheme, *field-isolate*.

Blocking scheme type *entropy-isolate* There are more advanced blocking schemes than what we refer to as *field-isolate*. For example, Milojevic (Milojević, 2013) introduces a quite complex scheme:

1. *J. Doe* generalises to *J. Doe*;
2. *John Doe* generalises to *J. Doe*;
3. *John H. Doe* generalises to *J. H. Doe* unless there is no other *J. X. Doe*, in which case it generalises to *J. Doe*
4. Anything more general than *J. Doe* is not considered.

Obviously, this approach is trying to utilise the second name's initial if it helps discrimination among siblings, and ignore it if it only separates child and parent (which by definition could both refer to the same person). Neither the frequency of observation, nor full first names are considered. Also, the constant of one sibling being enough is a rather arbitrary choice. Absorbing the basic idea, we introduce a straightforward scheme called *entropy-isolate*. Here, we simply calculate the normalised entropy H of the frequency distribution of a node's children and determine that it should be isolated if H is above a certain threshold t:

$$H(C(x)) = -\sum_{x' \in C(x)} p(x'|C(x)) \cdot \log_2 \left(p(x'|C(x)) \right)$$

$$p(x'|C(x)) = \frac{\#(x')}{\sum_{x'' \in C(x)} \#(x'')}$$

where $C(x)$ is the set of children names of name x. The frequency distribution can be over the childrens' own frequency or the aggregated frequency of mentions that they cover.

3.4.3 Experimental evaluation and comparison

Our experiments are conducted on the Web of Science corpus with more than 150 million author mentions. A subset of these mentions have been annotated with researcher's identifier (*rIDs*) that allow us to evaluate the performance of different blocking schemes. See Table 17 for some insights regarding authors in the Web of Science, the impact of author homonymy and the coverage of annotated identifiers. To evaluate the effectiveness of a blocking scheme, we compare the following three matrices / relations:

1. Blocking matrix B, where $B_{i,j} = 1$ iff names i, j are in the same block;
2. Name-matching matrix C, where $C_{i,j} = 1$ iff names i, j could refer to the same person;
3. Gold-standard matrix G, where $G_{i,j} = 1$ iff mentions i, j have the same rID.

Table 17: Statistics on the Web of Science. Surnames is the number of different surnames, mentions is the number of name mentions and size x refers to the number of surnames which are mentioned x times.

Settings	Surnames	Mentions	Size 1	Size 2	Size 3
With rID	$125k$	$7116k$	$19k$	$9k$	$6k$
With or without rID	$3144k$	$171249k$	$278k$	$87k$	$45k$

We can compare all these matrices in terms of precision and recall with one-another: (1) B against C, (2) B against G and (3) C against G (see Figure 21). We note that B has a row and column for each *observed name* with the current surname, whether one of the name's mentions has been assigned an rID or not. The same holds for C. The gold-standard matrix G has a row and column for *mentions*, not names, and only for those that have been annotated with an rID. For (1) in Figure 21, we can directly compare B and C. For (2), we first have to expand B by using every row as many times as the respective name has been annotated with an rID. This is often zero times, which means removing the row. For (3), we do the same on C. The comparison between B and C can be done without any annotation and may be used to estimate expected loss of recall for a given blocking scheme. The comparison between B and G is done to obtain the exact performance if annotation is present. The comparison between C and G also requires annotation but is independent of the blocking scheme. It allows to get an idea of the relationship between name-matching (as in C) and actual co-reference (as in G). This is particularly interesting to observe for varying problem sizes. Precision is the number of pairs that are given in $B \cap C$, $B \cap G$ and $C \cap G$ devided by the number of pairs in B, B and C, respectively. Recall is the number of pairs in $B \cap C$, $B \cap G$ and $C \cap G$ devided by the number of pairs in C, G and G, respectively. This amounts to the *pairwise-F1* metric (Menestrina, Whang, & Garcia-Molina, 2010). Table 18 gives an intuitive explanation of the meaning of Precision and Recall with the different comparisons.

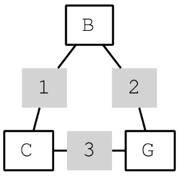


Figure 21: Comparing (1) blocks B against name-matches C, (2) against gold-standard G and (3) C against G.

Table 18: Intuitive explanations of the comparisons done in the evaluation.

Comparison			Suboptimal precision	Suboptimal recall
(1)	B	C	*blocks contain names that do not match*	*blocking separates matching names*
(2)	B	G	*blocks contain mentions with different rIDs*	*blocking separates mentions with the same rID*
(3)	C	G	*names match but refer to different authors*	*names do not match but have the same rID*

For our experiments, we normalise all names in the Web of Science to our name representation. For every blocking scheme $s \in S$, we then iterate over all distinct surnames $l \in L$ for which there is at least one mention with an annotated rID and calculate the precision and recall for the comparisons (1) to (3). This corresponds to one row of results for every pair in $S \times L$. As stated above, the comparison (3) is indifferent to s. We also store the number of rIDs annotated with the current surname as well as the number of observed names under it to keep track of the problem size. Finally, for every scheme s and every comparison, we plot the respective precision and recall value (one point per surname) against the problem size (either *number of annotated rIDs* or *number of observed names*). A moving average over these points shows how the performance of a blocking scheme relates to different problem sizes. Comparing the figures for different schemes gives an idea of a scheme's usefulness.

In Figure 22 and 23, we can see that evaluation results for the currently tested blocking schemes are subject to strong variance. Still, using a rolling average over growing problem sizes, we can identify that for field-isolate blocking schemes, the name-matching heuristic becomes more and more inappropriate in terms of recall while precision remains stable. Real recall values calculated against annotated authors are not particularly promising for field-isolate blocking schemes in general. The meaning of precision and recall in our setting is explained in Table 18. As a clear result, it is better to isolate names without two initials than to isolate those without a full first name. The precision is almost the same, while relying on the existence of first names separates too many mentions that belong together. Regarding entropy-isolate (Figure 23), we see that the threshold t has great impact of the effect of the scheme. This could be seen as an advantage, as this value allows to control blocking behaviour. For a good recall value, we would have to use a high entropy threshold around 0.8. These results are still preliminary. In particular, the aspect of computational cost (block sizes) requires further investigation.

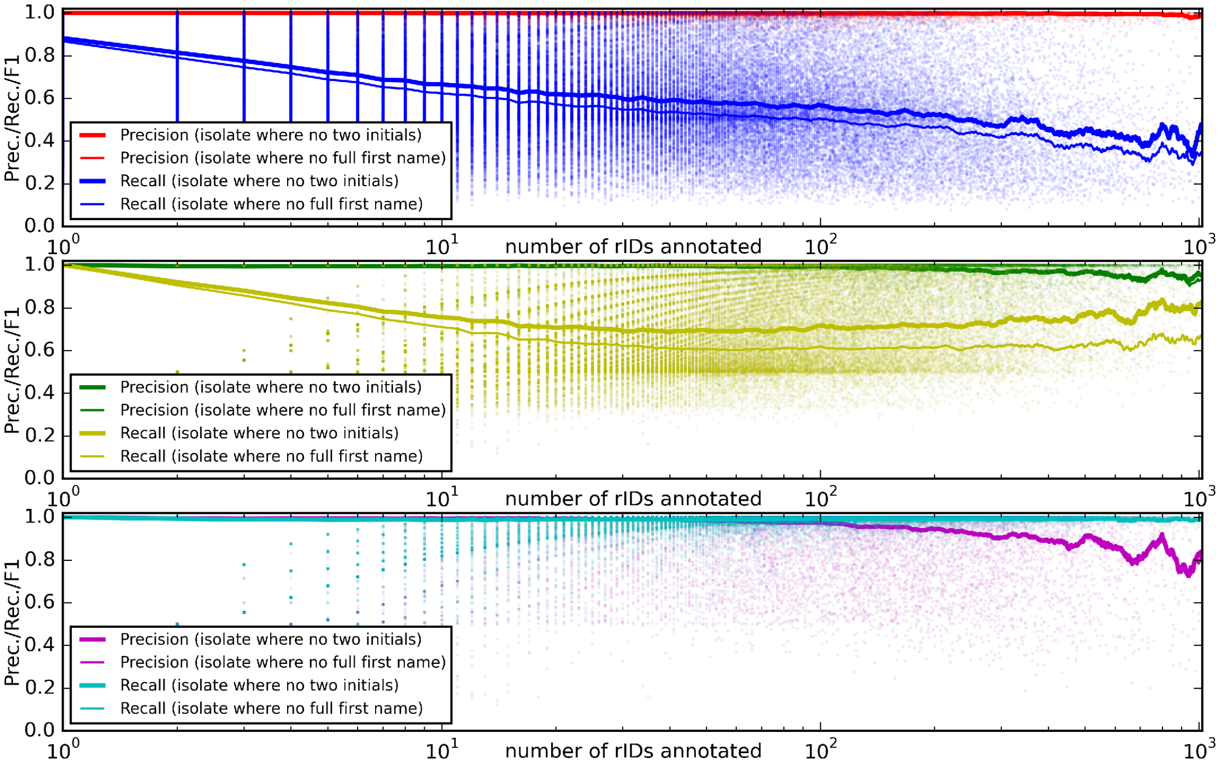


Figure 22: Preliminary evaluation by problem size: two field-isolate blocking schemes compared: (a) isolate all names without two initials, (b) isolate all names without a full first name. We show the comparisons (1), (2) and (3) from top to bottom.

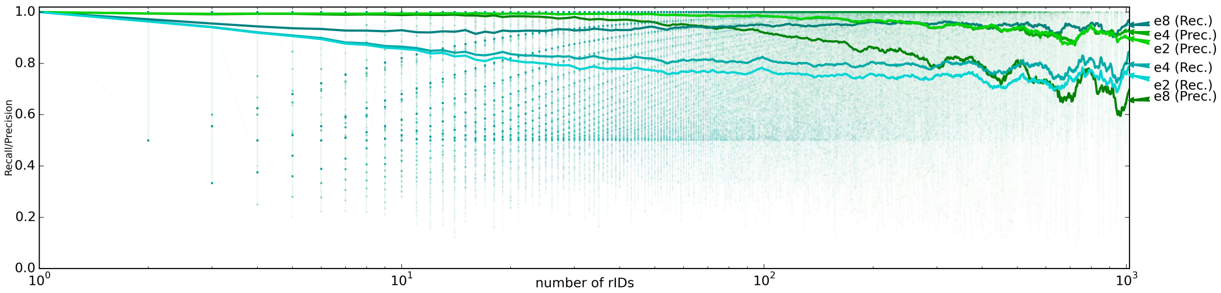


Figure 23: Preliminary evaluation by problem size: three entropy-isolate blocking schemes compared: (e2) $t = 0.2$, (e4) $t = 0.4$, (e8) $t = 0.8$.

3.4.4 Implementation, APIs and integration

It cannot be denied that our analysis comes with considerable computational complexity, which is why we briefly sketch the implementation that makes it feasible in the following. First of all, we note that due to a minimum amount of information required to be present in a reasonable name, we iterate over the data surname by surname. For every surname, we create a (sparse) quadratic matrix N that represents the reflexive closure of the parent-relation and its symmetric closure M as well as the frequency of observation: $N_{i,j} = \#(i) \cdot \#(j)$, where $\#(i)$ is how many times name i has been observed in the data. This includes all unobserved names k (below the current surname; $\#(k) = 0$) that are parents of observed names. In addition to that, for every potentially coreferring pair of names under the surname, we add their most general specification (i.e. *J. H. Doe*+*John Doe*=*John H. Doe*). In the next step, we isolate all nodes as indicated by the current scheme (setting all other entries in the respective row and column to zero). After this, we have to 'clean up' N, by deselecting the row and column of nodes that now have either (1) no observed ancestors, or (2) no observed descendents and at most one observed parent.

For a directed acyclic graph (such as represented by N) there is a known transitive closure algorithm that comes at acceptable computational cost and can be used to compute the number of descendents and ancestors. After cleaning, we are left with a much smaller number of relevant rows and columns. When we apply this selector on M, computing the transitive closure of the symmetric and reflexive relation becomes feasible, even though we cannot use the same efficient algorithm as on N. Here, we can use a standard algorithm for detecting connected components. The transitive closure on the reduced M specifies the reachability relation on the modified graph (with certain names isolated) and thereby the blocking matrix B as such, which can be evaluated against other (gold-standard) matrices as described in the experiments section.

3.5 Duplicate detection

3.5.1 Problem statement

In a large database that is continuously expanding due to new documents added from different sources, duplicates is a common problem: two metadata entries refer to the same document, but might describe it in a slightly different way. For example, one field could be empty in one of the items and filled in the other. It is obvious that one wishes to merge such duplicates so as to avoid confusion of the user, keep the index as small and clean as possible and aggregate as much information as possible. The task can be divided into two steps: identifying the duplicates and merging them. So far, we have only worked on the first problem. Although these are traditionally different research fields, both author name disambiguation and duplicate detection can be solved in the same framework. Nevertheless, the empirical distribution of duplicates and same authors are very different and so are the features and parameters that optimise any approach trying to model them.

3.5.2 Method description

In order to identify duplicates in a large collection of metadata entries, one has two solve two problems: (1) determine a similarity measure that can distinguish a duplicate pair from a non-duplicate pair; (2) find a way to keep the computational complexity low. Regarding the first problem, we reuse the effective similarity measure deployed in our work on author name disambiguation, as described in deliverable D3.1 (Blume et al., 2017). The set of features used has to be adapted to the nature of duplicates. When using a standard set of features (in particular terms from title and abstract), we found similar documents. This can be useful for presenting the user with results similar to the one he is currently viewing. However, this similarity does not necessarily model the probability of a pair of documents being duplicates. The definition of the optimal feature set for duplicate detection is ongoing work but we have already implemented the framework and adopted it for integration into the MOVING platform. The features are extracted directly from the Elasticsearch[46] index[47]. In order to keep computational complexity low, we build upon work done at GESIS in the DFG Project POLLUX[48]. Certain high-precision features are used to create blocks of documents and only within those are all pairs compared. Otherwise, the quadratic complexity of the problem would make it impossible to apply any similarity measure.

In order to reduce the number of pairs to be compared, we have to partition the collection into blocks and then only compare pairs inside each block. Folowing the work done in the *POLLUX* project, we distinguish between similarity features f mapped to documents x in a $doc \times feat$ matrix $\#(\vec{f},x)$ and duplicate indicators

[46] https://www.elastic.co/products/elasticsearch

[47] See deliverables D4.1 (Gottfried, Grunewald, et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017) for additional information on the MOVING platform's architecture

[48] http://www.pollux-fid.de/

$\#(\vec{f},x)_d$. The method was then reimplemented in MOVING with linear algebra, to make it more efficient. For every feature-type (i.e. DOI or title terms), there is a separate matrix. A requirement on duplicate indicators is that for each document and feature-type there is exactly one feature (i.e. one DOI). Therefore, we can sort the documents for each feature-type by the one feature they are assigned and determine groups of documents that have the same feature for a given feature-type. We then know for every feature-type, which pairs of documents to consider for further comparison. Details are given in Table 19. The input matrix $\#(\vec{f},x)_d$ assigns each document its feature in the current feature-type (marked by ⊗ in the respective cell). In step (1), the matrix is sorted by features, which leads to the second matrix. Here, we can find offsets where the features change, going through the sorted matrix document by document. From the offsets, we can derive the sizes of the blocks. In step (2), we create a new matrix using the sizes of the blocks (i.e. first a 2×2 block, then a 1×1 block, then another 2×2 block, and so on). Since we neither need to compare pairs of identical documents nor a pair (b,a) if we already have (a,b), only the upper or lower triangle of the matrix is relevant (irrelevant pairs are marked by ⊙). Finally, in step (3), we undo the sorting of the matrix to determine the position of the selected pairs in the input matrix. The result is the matrix M_ftype.

feat → *doc* ↓	a	b	c	d	e	f	g
1	⊗						
2			⊗				
3						⊗	
4					⊗		
5			⊗				
6	⊗						
7		⊗					
8				⊗			
9							⊗

$\#(\vec{f},x)_d$

(1) ⇒

	a	b	c	d	e	f	g	*size*
1	⊗							2
6	⊗							
7		⊗						1
2			⊗					2
5			⊗					
8				⊗				1
4					⊗			1
3						⊗		1
9							⊗	1

(2) ⇒

	1	6	7	2	5	8	4	3	9
1	⊙	⊗							
6	⊙	⊙							
7			⊙						
2				⊙	⊗				
5				⊙	⊙				
8						⊙			
4							⊙		
3								⊙	
9									⊙

(3) ⇒

	1	2	3	4	5	6	7	8	9
1	⊙					⊗			
2		⊙			⊗				
3			⊙						
4				⊙					
5		⊙			⊙				
6	⊙					⊙			
7							⊙		
8								⊙	
9									⊙

M_{ftype}

Table 19: Blocking with duplicate indicators: (1) sort the $doc \times feat$ matrix by $feat$ and get offsets and sizes of blocks, (2) get indices to compare in the sorted matrix, (3) unsort the matrix to get M_{ftype}.

We can combine multiple duplicate indicators in boolean logic. For example, we might want to say that we want to have a closer look at cases where year AND first author are identical OR where the title is identical OR where the volume AND issue AND venue are identical OR where the DOI is identical:

$$DOI \vee title \vee (year \wedge author_1) \vee (volume \wedge issue \wedge venue)$$

The set of pairs to be computed for this blocking scheme can be computed directly by boolean addition and multiplication of the respective feature-type matrices M_{ftype}:

$$M_{DOI} + M_{title} + (M_{year} \times M_{author_1}) + (M_{volume} \times M_{issue} \times M_{venue})$$

Once we have the set of pairs that we want to compare, we apply the same similarity measure as used for author name disambiguation, where the features are now document-specific similarity features $\#(\vec{f},x)$, for example title terms or author names (one probability p for every feature-type):

$$p(x|x') = \frac{\sum_f (\#(f,x) \cdot \#(f,x')) + \frac{\varepsilon}{|X|}}{\#(x') + \varepsilon}$$

$$\#(x') = \sum_f \#(f,x')$$

Based on the methodology proposed in the *POLLUX* project, we then apply a logistic regression classifier on the similarities given as $p_{ftype}(x|x')$:

$$p(Y = dup|X = (x,x')) = \frac{1}{1 + \exp(-(b + \sum_{ftype} \lambda_{ftype} \cdot p_{ftype}(x|x')))}$$

where b is a bias for the decision boundary between the classes $Y = dup$ or $Y = nodup$ (normally $b = 0.5$) and λ_{ftype} is the feature-type weight, which is trained on a portion of the data annotated in the *POLLUX* project.

3.5.3 Experimental evaluation and comparison

In the POLLUX project, a gold standard of annotated duplicates has been created. The approach has been tested against it with a selection of features available in the MOVING platform. As Precision and Recall were lower as for the setup that was tested on the POLLUX data using the respective features, we consider it ongoing work to determine a set of features that can be applied reliably in the MOVING context as well. Detailed feedback was provided on why duplicates found by us were sometimes not actual duplicates. We will further investigate how we can exploit the knowledge about the nature of duplicates in digital libraries that is present in this feedback.

3.5.4 Implementation, APIs and integration

Both the similarity measure and the blocking procedure have been formalised in the MOVING project as efficient linear algebra operations on sparse matrices. Therefore, we are optimistic that upcoming tests on the entire MOVING index will be successful in terms of runtime and memory consumption. The framework has been integrated into the MOVING context by providing scripts that run on the Elasticsearch index used in the MOVING platform. Table 20 explains the blocking step of the duplication method in a way very close to the actual implementation. The implementation of the similarity step uses the $doc \times feat$ matrix $\#(\vec{f},x)$ and is as follows:

$$p(\vec{x}|x') = \frac{\#(\vec{f},x)\cdot\#(\vec{f},x)^T + \frac{\varepsilon}{|X|}}{\#(\vec{x})+\varepsilon}$$

$$\#(\vec{x}) = \left(\#(\vec{f},x)\cdot\left\langle \begin{matrix}1\\ \vdots\\ 1\end{matrix}\right\rangle\right)^T$$

where only for those (x,x') that are selected for comparison.

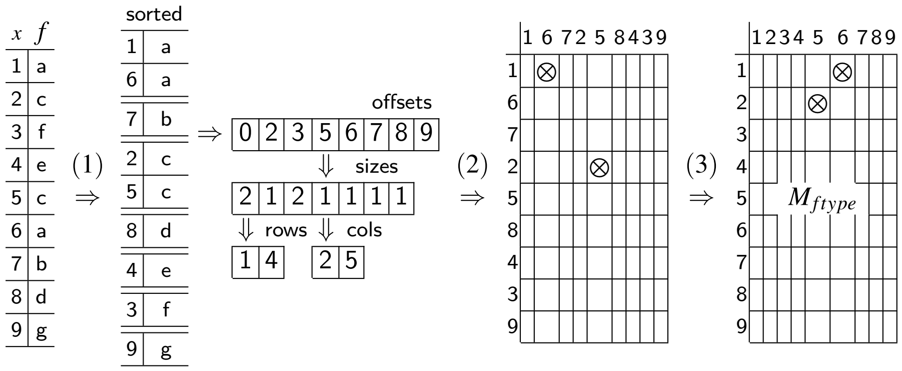


Table 20: Blocking with duplicate indicators: the same method as shown in Table 19 but closer to the actual implementation. The result is M_{ftype}.

3.6 Comparison of the Performance of Ad-hoc Retrieval Models over Titles vs. Fulltexts

While there are many studies on information retrieval (IR) models using full-text, there are presently no comparison studies of full-text retrieval vs. retrieval only over the titles of documents. This is an interesting question since the full-text of documents like scientific papers, news, etc. is not always available due to, e. g., the copyright policies of academic publishers. However, conducting a search based on titles alone has strong limitations. Titles are short and therefore may not contain enough information to yield satisfactory search results. We compare classical, modern, and recent IR models with respect to their search performance on full-text and titles of documents using different datasets. The results show that it is possible to build effective title-based retrieval models that provide competitive results comparable to full-text retrieval.

3.6.1 Problem statement

Using only titles has shown to be effective for document classification (Galke, Mai, Schelten, Brunsch, & Scherp, 2017) and top-k recommendations (Nishioka, Große-Bölting, & Scherp, 2015). This motivates us to investigate the possibility of building retrieval models based only on documents' titles. According to recent literature (Croft, Metzler, & Strohman, 2010), there are four main categories of ranking models: (1) set theoretic models or Boolean models, (2) vector space models (e. g., TF-IDF), (3) probabilistic models (e. g., BM25), and (4) feature-based retrieval (e. g., learning to rank). Furthermore, there are recent advances in Deep Learning that provide semantic neural network models capable of capturing the semantics of words. These models have been used for building information retrieval models based on short text or full-text. To the best of our knowledge, only the work by Shoval et al. (Shoval & Kuflik, 2004) compares ad-hoc retrieval over titles with ad-hoc retrieval over full-text by assessing both precision and recall of the retrieved documents. The authors conducted a case study with the help of 34 participants. Each participant defined two search queries and used the Google search engine to run each of his queries twice: One with the title field search and one with the simple full-text search. Afterwards, the users evaluated the top 10 results. In their study, they found no significant differences in the web retrieval performances when using titles or full-text. In fact, the results for title-based ad-hoc retrieval were slightly better. However, the authors acknowledge that their study is limited due to the small number of queries and the fact that only a single search engine, Google, was used. Nishioka and Scherp (Nishioka et al., 2015) showed that a title-based recommender system can provide very competitive results compared to a full-text based recommender system. The authors presented a novel profiling method called HCF-IDF. Other studies compare full-text retrieval with descriptive metadata in general, often including the abstract.

In our study, we used datasets from different domains and retrieval models from different Information Retrieval (IR) categories in order to compare the full-text vs. title searches. For this purpose, we utilised five datasets, out of which three datasets are obtained from well-known digital libraries: Econbiz[49], PubMed[50] and IREON[51], and two standard test collections for information retrieval (Christopher, Prabhakar, & Hinrich, 2008): NTCIR-2[52] and TREC Disks 4&5 [53]. We assume an ad-hoc retrieval task, i. e., users search for a specific information and usually choose from the first *twenty* search results (Shoval & Kuflik, 2004).

Our results show that it is possible, given certain constraints, to build an effective title-based information retrieval model that provides competitive results compared to a retrieval model operating on full-text. From the different categories of ranking models, learning-to-rank (L2R, see Section 3.6.2.2) outperforms other title-based statistical ranking models. L2R only requires a small set of features, which is automatically determined by a correlation-based feature selection method applied on a large set of established IR retrieval features. The average evaluation results over all datasets show that the best full-text-based retrieval models outperform the best title-based retrieval models by only 6.6%.

3.6.2 Compared models

We discuss the selected models for comparing title vs. full-text retrieval. We cover the dominant categories of retrieval models (Croft et al., 2010). We start by discussing the vector space and probabilistic models. Subsequently, we present a set of learning to rank and deep neural networks models. Finally, we describe the previous comparative works of titles vs. full-text retrieval.

3.6.2.1 Traditional models

As a baseline, we employ the vector space model Term Frequency – Inverse Document Frequency (TF-IDF) (Salton, Wong, & Yang, 1975). TF represents the frequency of occurrence of a term, while the IDF factor of a term is inversely proportional to the number of documents in which the term appears. This means the fewer the term appears in the corpus, the higher the IDF factor and vice versa.

Goosen et. al (Goossen, Ijntema, Frasincar, Hogenboom, & Kaymak, 2011) presented their novel concept-based model, Concept Frequency - Inverse Document Frequency (CF-IDF). It is an extension of TF-IDF that counts concepts C instead of terms T which outperforms the popular TF-IDF model. Nishioka and Scherp presented Hierarchical Concept Frequency - Inverse Document Frequency (HCF-IDF) (Nishioka et al., 2015), an extension of CF-IDF that considers the hierarchical structure of concepts C. Their algorithm gives less weight to the more general concepts in the hierarchy and outperforms CF-IDF. The concepts are defined in

[49] https://www.econbiz.de
[50] https://www.ncbi.nlm.nih.gov/pubmed
[51] https://www.ireon-portal.de
[52] http://research.nii.ac.jp/ntcir/permission/perm-en.html#ntcir-2
[53] http://trec.nist.gov/data/qa/T8_QAdata/disks4_5.html

controlled vocabulary thesauruses and are extracted from the documents by text matching. As a concept-based model, we employ the TF-IDF extensions, CF-IDF and HCF-IDF.

Another retrieval model which utilises the IDF weighting for ranking the documents is Best Matching 25 (BM25) (Robertson & Walker, 1994). It is one of the most popular information retrieval methods. It has been used as a baseline in TREC Web track[54].

3.6.2.2 Learning to rank models

Learning to Rank (L2R) is a family of machine learning techniques that aim at optimising a loss function regarding the ranking of items. It has been successfully applied in the past for different IR tasks. Chen et al. (Chen, Spina, Croft, Sanderson, & Scholer, 2015) proposed a L2R approach for finding non-factoid answers in an answer sentence retrieval task. They used a combination of Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007; Bai et al., 2010), Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) as semantic text representations, and Metzler and Kanungo's features (MK) (Metzler & Kanungo, 2008). Chen et al. showed that the combination of the semantic features and the MK feature set provides better ranking results than ranking based on MK feature set.

L2R consists of a set of supervised ranking models that are trained with a set of numerical feature vectors in order to retrieve the top-k relevant documents in response to a user's query. The feature vectors are calculated using the content of the documents and/or the queries. L2R models are generally categorised in point-wise, pairwise, and list-wise approaches depending on the way the model performs the optimisation task (Liu, 2009).

Point-wise models generate a relevancy degree for each single document regardless of its position in the result list for the query. In contrast, the loss function of pairwise approaches only considers one pair of documents at a time. For list-wise techniques, the input consists of the entire list of documents associated with a query and the output consists of a ranked list of documents for each query. As a pairwise approach, we use RankNet (Burges et al., 2005), LambdaMART (Wu, Burges, Svore, & Gao, 2010), and RankBoost (Freund, Iyer, Schapire, & Singer, 2003). Finally, for list-wise L2R we use AdaRank(J. Xu & Li, 2007), Coordinate Ascent (Metzler & Croft, 2007), and ListNet (Cao, Qin, Liu, Tsai, & Li, 2007). In the following, we briefly introduce these methods.

RankNet is a feed-forward neural network that uses stochastic gradient descent to adjust the ranking of documents. For a given query and each pair of documents, the model computes two scores based on the feature vectors. The scores are utilised to rank the documents.

LambdaMART combines LambdaRank, a neural network pairwise L2R approach, and Multiple Additive Regression Trees (MART), which uses gradient boosted decision trees for prediction. When comparing a pair of documents, the gradient of the cost function indicates in which direction a document should move in a ranked list. In LambdaMART, this gradient is incorporated in the gradient descent to decrease the loss of the overall model score.

RankBoost is a pairwise approach which uses boosting to improve model performance. It is based on the AdaBoost algorithm introduced by Freund and Schapire (Freund, Schapire, & Abe, 1999). The aim is to train a number of so-called *weak learners* which are then distributed over the initial data. For each subsequent learner, the distribution is updated based on the ranking performance of the previous learner. For each update step, the weights of the document pairs that are correctly ordered are increased and the weights of incorrect pairs are decreased. The final ranking is then a weighted sum of all *weak* rankings.

AdaRank trains ranking models by directly optimising performance measures like normalised discounted cumulative gain. AdaRank is closely related to RankBoost as it uses boosting to construct *weak learners* and uses the combination of those to predict rankings. However, the update step for AdaRank uses the performance.

Coordinate ascent is a list-wise approach introduced as an optimisation method for constrained optimisation problems. Coordinate ascent's scoring function is comprised of a linear combination of the features. It optimises the objective function by iteratively choosing one dimension (or feature) to search for and fix all remaining dimensions. The objective function aims to directly optimise an evaluation function, which can be any standard information retrieval evaluation metric like mean average precision or normalised discounted cumulative gain.

[54] http://trec.nist.gov/data/webmain.html

ListNet is a list-wise ranker that optimises a list-wise loss function using a neural network as a model and gradient descent as an optimisation algorithm. It is similar to RankNet, but uses document lists instead of document pairs.

3.6.2.3 Deep learning models

As with almost all fields, the recent resurgence of neural networks has also affected the Information Retrieval community. Zhang et al. (Y. Zhang et al., 2016) provided a detailed survey to illustrate the rough evolution of Neural IR research and word embedding approaches to IR. For web search, Huang et al. (P.-S. Huang et al., 2013) propose a series of deep structured semantic models (DSSM). The most successful instance of the model uses a multilayer feed-forward neural network to map both the query and the title of a webpage to a common low-dimensional vector space. The similarity between the query-document pairs is computed using cosine similarity. The main novelty is the usage of word-hashing, which dramatically reduces the vocabulary size without neglecting too much information. The reduction in vocabulary size allows the neural network to learn effectively from a large amount of available labeled data. DSSM is composed of four different layers. The first layer is the input layer. It contains the word sequences of the document and the user query. The second layer transforms the word sequences into sub-word units to reduce the large amount of vocabulary size. Subsequently, the sub-word units are used as input for a feed-forward neural network. In order to determine the relevancy of a document, cosine similarity between the query and the documents is computed on the output layer. The documents are ranked with respect to their similarity scores. As an extension to the DSSM model, Shen et al. (Shen, He, Gao, Deng, & Mesnil, 2014b) enhance on that by replacing the feed-forward neural network with a convolutional neural network. Afterwards they introduced convolutional neural networks with max-pooling in the DSSM architecture (C-DSSM) (Shen, He, Gao, Deng, & Mesnil, 2014a). The convolutional layer and max-pooling layer are utilised to identify key words and concepts, in both the query and the document, and project them into a lower-dimensional semantic layer.

However, recently, the DSSM's success has been put into perspective. Cohen et al. (Cohen, Ai, & Croft, 2016) investigated the suitability of different neural network approaches for different IR tasks. They showed that DSSM performs poorly on a traditional dataset for ad-hoc retrieval (Robust04) and argue that the word-hashing method discards too much information when documents are long. Moreover, when the document is long, supervised methods require an increasing amount of training data to correctly identify which of the words from the document are relevant to the query. However, we are interested in the best possible solution for IR on titles. Thus, DSSM is well-suited to our experiments.

Another large branch of research that is related to neural networks attempts to leverage word embedding and document embedding for IR. In their examination of neural network approaches for IR tasks, Cohen et al. argue that embedding-based methods suffer from the inability to distinguish between semantically similar words and require additional mechanisms to account for exact matching signals (Cohen et al., 2016). Therefore, a lot of the related work that leverages word embeddings for IR incorporates the usage of word embeddings in a language model (Ganguly, Roy, Mitra, & Jones, 2015; Zuccon, Koopman, Bruza, & Azzopardi, 2015; Zamani & Croft, 2016) or document embeddings in a language model (Ai, Yang, Guo, & Croft, 2016a, 2016b). Chen et al. (Chen et al., 2015) proposed a learning to rank approach for finding non-factoid answers in an answer sentence retrieval task. In their learning to rank retrieval approach they used a combination of Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007), Word2Vec (Mikolov, Sutskever, et al., 2013) as semantic text representations and Metzler and Kanungo's features (MK) (Metzler & Kanungo, 2008). This motivates us to use Word2Vec as a feature in our learning to rank approach.

3.6.3 Experimental evaluation and comparison

In order to evaluate the effectiveness of title-based retrieval vs. a full-text retrieval, we use five datasets, which are described in Section 3.6.3.1. In Section 3.6.3.2, we present our evaluation procedures. In Section 3.6.3.3, we show how we have applied the correlation-based feature selection algorithm to derive a subset of features for learning to rank models. In Section 3.6.3.4, we describe the metric used for evaluating the retrieval results. Finally we outline the results in Section 3.6.3.5.

3.6.3.1 Datasets

We use labeled and unlabeled datasets which have a full-text and a title. Thus, a comparison is possible. The two labeled datasets, namely NTCIR-2 and TREC 4&5, provide a set of topics and human relevance judgments. The three unlabeled datasets are Econbiz, IREON, and PubMed. The unlabeled datasets come with a hierarchical, domain-specific thesaurus that provides topics on economic, political, and medical subjects.

Labeled datasets

NTCIR-2 The dataset consists of 49 search topics and around 322,059 documents. We use the search topics as queries. The documents were extracted from the NACSIS Academic Conference Paper Database, collected between 1997-1999, and NACSIS Grant-in-Aid scientific research database, collected between 1988-1997. The documents includes electronics, chemistry, physical sciences, and clinical reports. Furthermore, the dataset includes relevance judgments of 66,729 query-document pairs.

TREC 4&5 Also known as TREC Robust, it consists of 507,011 English documents from various newspaper or newswire sources (Financial Times, Foreign Broadcast Information Service, Los Angeles Times) and government proceedings (Congressional Record, Federal Register) collected between 1998-1994. For our investigation all data items needed to have a full-text and a title. When examining the files, around 50 thousand documents were missing one of these elements. These documents (mainly from Federal Register and Los Angeles Times) were ignored for our experiments.

TREC provides human annotated relevance judgments for some query-document pairs. We used TREC-6 ad-hoc qrels[55] in our experiments. TREC-6 ad-hoc qrels consists of 50 topics and relevance judgments of 72,270 query/document pairs.

Unlabeled datasets

EconBiz EconBiz[56] is a search portal for economics' scientific publications. From EconBiz, we obtained around 288,344 English publications. As user queries, we used two sources: the economics thesaurus (STW)[57] and user logs of EconBiz. The economics thesaurus provides more than 6,000 economics subjects, i. e., concepts in economics. The thesaurus is developed and maintained by an editorial board of domain experts. The user logs contain 28,353 unique user queries in English which were collected between 16th of July 2015 and 22nd of Oct 2015.

IREON The German information network *International Relations and Area Studies*[58] provided us with a dataset of 27,575 full-text politics' publications in English. The dataset also contains a politics thesaurus (FIV) with more than 7,000 political subjects. The thesaurus subjects were used as queries in our experiments.

PubMed The dataset contains 646,655 English full-text articles provided by the US national library of medicine. As queries, we used the medical terms from the Medical Subject Headings (MeSH) thesaurus[59]. MeSH consists of around 28,000 subjects.

3.6.3.2 Experimental procedure

In order to compare the retrieval performance over titles versus full-text, we have implemented the ranking models described in Section 3.6.2. In the case of the labeled datasets where the human relevance judgments are provided, the lists of the *top-20* documents generated by the full-text and titles retrieval models, respectively. The lists are compared to the relevance judgments provided as gold standard. Whereas with the unlabeled data set, the list of *top-20* relevant documents generated using the title-based retrieval models are compared to the relevance judgments of the full-text TF-IDF model. The relevance judgments of the *top-20* relevant documents of the baseline are grouped into four relevance groups and the remainder is considered as non-relevant.

In order to generate the evaluation results for vector space models and probabilistic models, tokenisation, stop words removal, and porter stemming were applied. The concept-based approaches HCF-IDF and CF-IDF utilise the concepts from STW, FIV and MeSH, while BM25CT means exploiting BM25 using a vector union of the terms and concept features.

For the L2R models, we use a set of 29 features (see Table 21) to train our models. These features are the Metzler and Kanungo (MK) (Metzler & Kanungo, 2008) set, modified LETOR (Qin & Liu, 2013), semantic features, and statistical features. The original MK feature set used six features for their query-based summarisation task. Due to the difference in our task (comparing query and title), we ignore the sentence location feature because the titles usually consist only of one sentence. In addition, we do not consider full-text TF-IDF for training the L2R models as it is utilised as baseline. Regarding the features Term Overlap

[55] http://trec.nist.gov/data/qrels_eng/
[56] https://www.econbiz.de
[57] http://zbw.eu/stw/version/latest/about
[58] http://www.fiv-iblk.de/eindex.htm
[59] https://www.nlm.nih.gov/mesh/

Table 21: Overview of the L2R features used in our comparison of full-text vs. title-based retrieval.

Type	Features
MK set (Metzler & Kanungo, 2008)	Sentence length, Exact match, Term Overlap, Synonym Overlap, Language Model with Dirichlet smoothing
Modified LETOR (Qin & Liu, 2013)	Covered Query Term Number, IDF, sum/min/max/mean/variance of TF, sum/min/max/mean/variance of length normalised TF, sum/min/max/mean/variance of TF-IDF, Language model absolute discounting smoothing, Language model with Bayesian smoothing using Dirichlet priors, Language model with Jelinek-Mercer smoothing
Semantic space set (Chen et al., 2015)	Word2Vec
Ranking model features	TF-IDF, BM25, CF-IDF, HCF-IDF

and Synonym Overlap, we removed stop-words and performed porter stemming on the queries and titles. The Term Overlap is the fraction of query terms that occur in the document (titles or full-text), while the Synonyms Overlap is the fraction of query terms that either occur or have a synonym in the document. We utilised NLTK[60] to generate synonyms. For the LETOR feature set, we ignore all web-related features (e.g. Sitemap term propagation). The language model parameters were taken from the original work. Additionally, based on previous work (Galke, Saleh, & Scherp, 2017), we use the vector representation of words (Word2Vec) to compute the similarity between a query-document pair and use it as an L2R feature. For this purpose, we use Google News, the pre-trained distributional model (Mikolov, Chen, Corrado, & Dean, 2013) and gensim(Rehurek & Sojka, 2010). Regarding the language model features, we used an Elasticsearch[61] full-text index to generate them. Moreover, we use the scores computed from BM25, CF-IDF and HCF-IDF on titles or full-text according to the experiment content configuration, as features.

We have trained in total seven L2R models using 29 features using default parameters of RankLib (Dang, 2010). In the following, we report the training parameters of the three top scoring models. We applied 5-fold cross validation. The LambdaMART model has been trained using 1,000 trees with 10 leaves per tree. The learning rate has been set to 0.1 and the threshold for tree splitting was 256 candidates. The minimum number of samples for a leaf was set to 1. Early stopping was applied, if there was no improvement for 100 consecutive rounds. The RankNet model was trained using 100 training epochs, one hidden layer and 10 hidden nodes. The learning rate was set to 0.00005. For the RankBoost model, we used 300 training epochs and 10 threshold candidates. AdaRank was trained in 500 rounds with a learning tolerance of 0.002. The number of epochs for the ListNet model was 1500 and the learning rate was set to 0.00001. In the case of Coordinate Ascent, we applied 5 random restarts and 25 search iterations per dimension. The performance tolerance was set to 0.001. No regularisation was used.

Finally, to generate the evaluation results for the semantic models DSSM and C-DSSM, we have used the Sent2vec framework [62]. A trained model, with a click through data of 30 million query/clicked-title pairs from Microsoft was used to determine the semantic cosine similarity between each query-document pair.

3.6.3.3 Feature selection for L2R models

A good IR system can retrieve most important documents in a fast and scalable way using only a limited amount of information about the query and documents. The information is contained in the features of both document and query and therefore a good set of features has to be found. The aim of the feature selection is to find a small and meaningful subset of features which can still produce accurate results. Given a large number of different IR features, we want to find those features which cover diverse information and still contribute the most to the retrieval of the most important documents.

Feature selection can be accomplished on the basis of the following hypothesis: "*a good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other*" (Gennari, Langley, & Fisher, 1989). In order to test our set of 29 features (shown in Table 21), we applied a Correlation-based Feature Selection (CFS) algorithm (Hall, 2000) on the labelled datasets. The CFS algorithm computes a score for a subset S of the 29 features containing k features using the following equation (denoted as $score_{CFS}(S)$

[60] http://www.nltk.org

[61] https://www.elastic.co/

[62] https://www.microsoft.com/en-us/download/details.aspx?id=52365

in (Hall, 2000)):

$$score_{CFS}(S) = \frac{k \cdot \overline{r_{gf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{1}$$

where $\overline{r_{gf}}$ is the average gold standard(g)-feature correlation and $\overline{r_{ff}}$ represents the average inter-correlation between the features. The formula scores those better subsets whose 'feature-feature' correlations are low and 'gold standard-feature' correlations are high.

We generated two files, one file contains titles and one contains the full-text. Afterwards, we calculated $score_{CRF}(S)$ for both feature files. Overall, $score_{CRF}(S)$ was positive value between 0.1 and 0.6, which indicates the high 'feature-gold standard' correlation and the low 'feature-feature' correlation.

Feature derivation

We have further investigated the possibility of deriving a meaningful subset of features that decreases the error rate of the ranking models. For this purpose, we calculated $score_{CRF}(S)$ for all feature subsets of sizes $|S| = \{1,\ldots,29\}$, which equals $2^{29} - 1 = 536,870,911$ possible subsets, for each dataset and configuration (full-texts or titles).

Table 22: Best feature subsets based on the CFS approach.

Dataset	Content	Best Feature Subset
NTCIR-2	Full-text	BM25, Exact Match
	Titles	BM25, Exact Match
TREC	Full-text	BM25, Exact match, Sum of length normalised TF
	Titles	BM25, Language model, Minimum of TF-IDF, Term overlap, Word2vec
Econbiz (STW)	Full-text	BM25, IDF, Language model absolute discounting smoothing, Language model with Bayesian smoothing using Dirichlet priors, Language model with Jelinek-Mercer smoothing, Maximum TF, Mean TF, Sum TF-IDF
	Titles	Minimum of TF-IDF, Sentence length
Econbiz (Logs)	Full-text	Language model with Dirichlet smoothing, Language model absolute discounting smoothing, Language model with Jelinek-Mercer smoothing, Max TF-IDF, Sum TF-IDF, Variance TF-IDF
	Titles	BM25, Exact match, Sentence length, Min length norm TF, Min TF, Min TF-IDF
Politics	Full-text	Language model with Dirichlet smoothing, Language model absolute discounting smoothing, Language model with Jelinek-Mercer smoothing, Mean TF-IDF, Minimum TF-IDF, Sum TF-IDF, Variance TF, Variance TF-IDF
	Titles	HCF-IDF, Exact match, Min length norm TF, Min TF, Min TF-IDF
PubMed	Full-text	BM25, Language model with Dirichlet smoothing, IDF, Maximum TF-IDF, Mean TF-IDF, Minimum TF, Variance of TF-IDF, Sum TF-IDF
	Titles	Minimum TF-IDF

In order to evaluate the effectiveness of the feature selection method, we chose the Best Feature Subset (BFS) for each dataset and configuration. The best feature sets, in terms of their $score_{CRF}(S)$, are reported in Table 22. We then repeat our learning-to-rank experiments using this feature subset. The results are presented in Table 23 and 24.

The CFS results showed that some features, such as BM25, contribute the most to the results. Those results are consistent with that of Qin et al. (Qin & Liu, 2013), who found that using BM25 as a feature in L2R models improves the overall performance of the L2R models.

3.6.3.4 Evaluation metric

We evaluate the retrieval results using normalised discounted cumulative gain ($nDCG$) (Järvelin & Kekäläinen, 2002). We assume that users do not look beyond two pages of 10 results (Shoval & Kuflik, 2004). Thus, we limited our evaluation to the top 20 results. The metric *nDCG* compares the top-20 ranked list of documents (*DCG*), with the gold standard and is computed as follows:

$$nDCG@k = \frac{DCG@k}{IDCG@k}, \quad where \quad DCG@k = rel(1) + \sum_{i=2}^{k} \frac{rel(i)}{\log(i)} \tag{2}$$

D is a set of documents, $rel(d)$ is a function that returns one if the document is rated relevant, otherwise zero, and $IDCG_k$ is the optimal ranking.

3.6.3.5 Results

In this section, we present the results of the titles versus full-text retrieval comparison. Considering the evaluation results, we observe that the retrieval over titles yield a close $nDCG@20$ values, or even higher metric values in case of the NTCIR-2.

Table 23 presents the performance of the title- and full-text-based ranking models on the NTCIR-2 and TREC Robust dataset. In this case the gold standard is a manually annotated binary relevance judgments. For the NTCIR-2 dataset, we observe that the learning to rank model Coordinate Ascent, with the full set of 29 features, attained the $nDCG@20$ value 0.33, which is 0.01 higher than C-DSSM and BM25 on full-text retrieval. This metric value of 0.33 has also been attained from the titles retrieval using the DSSM model. Having reduced the features to the best feature set, BM25 and exact matching, coordinate ascent performance slightly improved on full-text retrieval (0.37). However the titles retrieval of the same model remained at the same value (0.29). In case of TREC Robust, we observe that BM25 achieved the best results on full-text and titles retrieval. BM25 attains a *nDCG* of 0.41 on full-text compared to 0.23 on titles. The L2R methods, LambdaMart and Coordinate Ascent, using the full feature set and the best feature set attained a very close results to BM25 on both full-text and titles.

Table 23: Average $nDCG@20$ of the full-text based and title-based retrieval models for NTCIR-2 and TREC Robust collection.

Method		**NTCIR-2**		**TREC Robust**	
Family	IR Method	Titles	Full-text	Titles	Full-text
VSM	TF-IDF	0.19	0.18	0.21	0.39
	CF-IDF	0.05	0.05	0.12	0.13
	HCF-IDF	0.23	0.24	0.10	0.12
PM	BM25	0.24	0.32	**0.23**	**0.41**
	BM25CT	0.24	0.31	0.20	0.405
L2R - FFS	LambdaMART	0.25	0.30	0.22	0.39
	RankNet	0.28	0.29	0.13	0.10
	RankBoost	0.26	0.32	0.21	0.34
	AdaRank	0.21	0.31	0.19	0.22
	ListNet	0.21	0.24	0.15	0.07
	Coord. Ascent	0.29	0.33	0.22	0.39
SM	DSSM	**0.33**	0.26	0.18	0.23
	C-DSSM	0.32	0.32	0.18	0.20
L2R - BFS	LambdaMART	0.20	0.15	0.16	0.33
	RankNet	0.28	0.25	0.05	0.046
	RankBoost	0.26	0.37	0.13	0.38
	AdaRank	0.29	**0.37**	0.18	0.37
	ListNet	0.29	**0.37**	0.19	0.37
	Coord. Ascent	0.29	**0.37**	0.18	0.38

Table 24 shows the performance of the title-based and full-text-based retrieval models on the Econbiz, IREON, and PubMed datasets compared to a baseline of a full-text TF-IDF model. STW concepts, Econbiz query logs, FIV and MeSH concepts were used as user queries, respectively. Considering the evaluation results for the EconBiz dataset, we inspect four configurations of either the full-text or the titles field and either STW or EconBiz queries. We observe once more that the retrieval over titles yields higher nDCG metric values. In case of the STW queries, the best title-based retrieval model, LambdaMart, attained a nDCG of 0.83 compared to 0.76 of Coordinate Ascent on full-text (7% relative improvement). In case of the EconBiz query logs, we observe that Coordinate Ascent, using the best feature set, on titles attained a high nDCG value of 0.74 and outperformed all other retrieval model. For IREON and PubMed, the titles retrieval was not competitive with the full-text retrieval. The best title-based model, LambdaMart, attained nDCG values of 0.51 and 0.60, 18% and 24% lower than the best full-text retrieval model (Coordinate Ascent), respectively.

Table 24: Average *nDCG@20* of the titles and full-text-based ranking models using the EconBiz, IREON and PubMed datasets with STW, user logs, FIV and MeSH queries respectively. Full-text TF-IDF gold standard was utilised, thus the 1.0 score (in italics)

Method		**Econbiz**				**IREON**		**PubMed**	
		STW		user logs		FIV		MeSH	
Family	IR Method	Titles	Full-text	Titles	Full-text	Titles	Full-text	Titles	Full-text
VSM	TF-IDF	0.43	*1.0*	0.38	*1.0*	0.43	*1.0*	0.58	*1.0*
	CF-IDF	0.39	0.39	0.16	0.17	0.22	0.22	0.03	0.001
	HCF-IDF	0.23	0.27	0.16	0.10	0.22	0.22	0.43	0.45
PM	BM25	0.44	0.59	0.24	0.45	0.43	0.61	0.58	0.62
	BM25CT	0.41	0.41	0.25	0.43	0.36	0.57	0.57	0.62
L2R - FFS	LambdaMART	**0.83**	0.73	**0.74**	0.57	**0.51**	**0.69**	**0.60**	**0.84**
	RankNet	0.60	0.10	0.64	0.10	0.19	0.10	0.22	0.72
	RankBoost	NaN	0.67	0.41	0.48	0.48	0.49	0.22	0.82
	AdaRank	0.20	0.67	0.43	0.50	0.23	0.25	0.32	0.79
	ListNet	0.50	0.11	0.69	0.12	0.43	0.12	0.24	0.72
	Coord. Ascent	0.54	**0.76**	0.73	0.62	0.49	0.50	0.45	0.83
SM	DSSM	0.29	0.22	0.29	0.22	0.32	0.25	0.36	0.33
	C-DSSM	0.29	0.25	0.29	0.22	0.31	0.28	0.35	0.36
L2R - BFS	LambdaMART	0.41	0.72	0.29	0.52	0.22	0.52	0.40	0.83
	RankNet	0.18	0.46	0.30	0.49	0.23	0.64	0.36	0.83
	RankBoost	0.16	0.59	0.30	0.43	0.22	0.40	0.28	0.82
	AdaRank	0.17	0.11	0.33	0.49	0.22	0.20	0.33	0.75
	ListNet	0.16	0.42	0.30	0.49	0.22	0.67	0.34	0.83
	Coord. Ascent	0.27	**0.76**	0.30	**0.64**	0.22	0.60	0.33	**0.84**

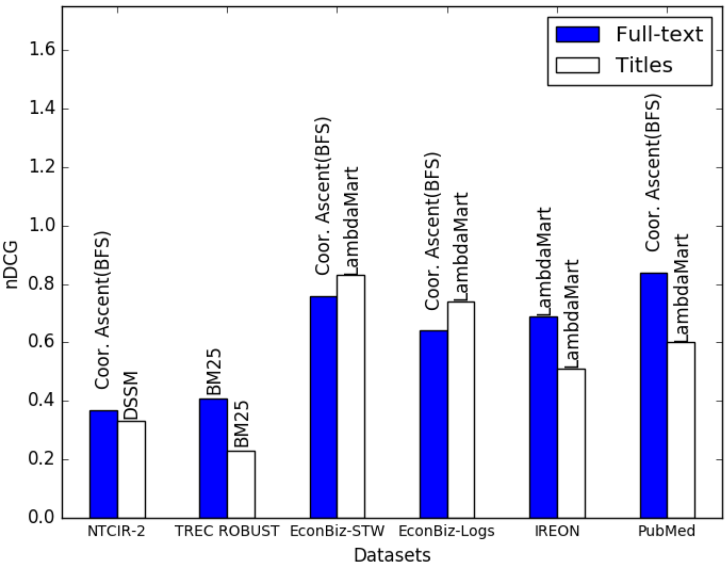


Figure 24: Average *nDCG@20* of the best performing retrieval methods for each dataset.

Overall, the results show that title-based retrieval models provided, in most cases, competitive results comparable to the full-text retrieval models. Figure 24 visualises the nDCG scores of the best performing retrieval methods for each dataset.

3.7 Titles vs. full-text for automated semantic document annotation

In this section, we investigate to what extent automated semantic document annotation can be conducted with only using the titles of documents instead of the full-text. We evaluate multiple traditional as well as modern multi-label classification approaches operating either only on the title or the full-text of the documents. The experiments reveal that, 80%-90% of the F1 score can be retained, when the classifier operates only on titles.

3.7.1 Problem statement

A significant amount of today's largest knowledge graph on the web, the so-called Linked Open Data cloud[63], consists of metadata about documents such as scientific papers and news articles. Domain-specific vocabularies are used to describe the semantics of these documents. Simple Knowledge Organization System (SKOS)[64] is an established W3C standard for modeling thesauri in domains such as economics, politics, social sciences, news, etc. Those thesauri are often of high quality since they are manually crafted as well as maintained by domain experts, and made freely available on the web[65].

The challenge is to use those SKOS thesauri to successfully annotate the documents with semantic concepts. However, the full-text PDF of the documents may not be available (linked from the documents' metadata) or may not be legally accessible due to licensing or copyright issues (even though there is a link to the PDF). Thus, it is highly desirable to conduct a semantic annotation of the documents with the SKOS thesauri by just using the already published documents' metadata like the title, year, authors. In contrast to the full-text of documents, the metadata is directly available on the Linked Open Data cloud, accessible in the machine readable format RDF[66], and can be processed with no legal barriers for semantic annotation. Conducting semantic annotations by using only the title (or further metadata of the documents) is challenging, since the title is short and thus carries only little information compared to the full-text. The process of semantic annotation is a multi-label classification task where not only one label has to be chosen as annotation but a set of labels since many concepts of the SKOS thesauri are needed to appropriately describe the semantics of the documents.

We tackle the challenge of conducting a semantic multi-label classification into SKOS thesauri by using only the title metadata of the documents. To this end, we run an extensive series of experiments to compare established methods and recent methods from machine learning for document classification. The goal is to decide whether it is possible to reach a comparable classification performance when using only the title of the documents. It is noteworthy that all the compared approaches operate on the underlying machine learning level which makes a comparison with prevalent end-to-end ontology tagging systems such as SOLR ontology tagger[67] and MAUI[68] difficult. Instead, we show that without using the hierarchical properties of the thesauri, the presented methods outperform the best-performing methods that do make use of the hierarchy such as the ones of our own prior work (Große-Bölting, Nishioka, & Scherp, 2015). Apart from the well-known multi-label classification baseline k-Nearest Neighbors (kNN) (Keller, Gray, & Givens, 1985) and Support Vector Machines (SVM) (Suykens & Vandewalle, 1999), we revisit traditional text classification methods such as Naive Bayes (McCallum, Nigam, et al., 1998), Rocchio (Joachims, 1997), and logistic regression (LR) (Genkin, Lewis, & Madigan, 2007). We also include the prominent Learning to Rank (L2R) approach to multi-labelling problems (M. Huang, Névéol, & Lu, 2011), as well as a modern variant of neural networks motivated by the success of the Deep Learning field. Please note, the present work focuses solely on using the titles of documents, since the title is the richest metadata attribute and contains keywords relevant in the domain. In the future, we may also incorporate other metadata like authors' names and publication year.

Our experiments show that it is possible to reach a competitive performance for semantic annotation using solely the title of documents, compared to exploiting the full-text of the documents. Using a sample-averaged F_1 measure (Manning, Raghavan, & Schütze, 2008) as evaluation metric, we compare the automated predictions of semantic annotations from different methods with those annotations provided by domain experts. We run our experiments over four large-scale documents corpora of different origin and domain with a total

[63] See latest version from 02/2017: http://lod-cloud.net/versions/2017-02-20/lod.svg
[64] https://www.w3.org/2004/02/skos/
[65] An overview of current SKOS vocabularies is maintained by the W3C: https://www.w3.org/2001/sw/wiki/SKOS/Datasets
[66] http://www.w3.org/standards/techs/rdf
[67] https://www.opensemanticsearch.org/solr-ontology-tagger
[68] https://github.com/zelandiya/maui-standalone

of over $300,000$ documents. All datasets offer professional labels, i. e., manual annotations from domain experts. Two datasets are from professional scientific libraries in economics and politics while the other two datasets are the well-known news corpora from New York Times and Reuters. In the past, algorithms of the lazy learner family such as kNN used to dominate multi-label classification tasks on such datasets with a high amount of classes (Spyromitros, Tsoumakas, & Vlahavas, 2008; Große-Bölting et al., 2015). However, we show that eager learners such as logistic regression and feed-forward neural networks outperform lazy learners. Most eager learners have the benefit of $\mathcal{O}(N_{\text{parameters}})$ time complexity to predict a label set for an unseen document, which is important when applying an automated semantic annotation process for on-the-fly enrichment of metadata on the Linked Open Data cloud. In contrast, lazy learners as well as Learning to Rank need to store and traverse $\mathcal{O}(N_{\text{training examples}} \cdot N_{\text{features}})$ space to predict the labels for a single new document at test time. Finally, focusing on the metadata also allows direct processing of data in published RDF format (e. g., the `rdfs:literal` and `rdfs:label` information) without accessing the full-text of the documents at all. Overall, we conclude that eager learning algorithms are well-suited for automated semantic annotation of RDF resources in Linked Data. Summarised, the contributions of this work are:

1. To the best of our knowledge, we conduct the first large-scale systematic comparison of multi-label classifiers applied to either the full-text or only the titles of documents.
2. Our results show that eager learners such as neural networks and linear models outperform lazy learners even when a high amount of possible labels is considered.
3. We offer evidence that using only the title for high-dimensional multi-label classification is a reasonable choice for semantic annotation of resources where only metadata is available, such as documents modeled in RDF on the Linked Open Data cloud.

3.7.2 Method description

We present an end-to-end apparatus for semantic annotation of unstructured text. Figure 25 shows our generic text processing pipeline that we used for the experiments. Each path through the graph resembles a possible configuration. We first describe the conversion from unstructured text to a vector representation, then we elaborate in detail on the classification methods that we have compared.

Counting terms and extracting concepts In the first step of our text processing pipeline, the raw text needs to be converted into a vector representation that can be supplied as input to the classifiers. As features, we use the counts of term occurrences in the text (TF) as well as the number of times a concept provided by a domain specific thesaurus can be extracted from the text (CF). A concept is a set of concept-specific phrases. In case of SKOS format, each concept has one preferred phrase (`skos:prefLabel`) and optionally a set of alternative phrases (`skos:altLabel`). We extract these concept-specific phrases from the text using a finite state machine. When there is more than one possible match in a sequence of words, we favor the longest phrase. We assume that longer phrases carry more specificity. Hence, the occurrences of a concept (set of concept-specific phrases) are counted in the same way as term occurrences. The effect of concept extraction is to ensure that domain-specific synonyms encoded in the thesauri are mapped to the same concept. The concepts are also directly associated to the respective class labels. Hence, it is left to the learning algorithm, to decide about the concrete label assignment, given the extracted terms or concepts.

Discounting frequent terms and concepts Inverse Document Frequency (IDF) is a re-weighting scheme introduced in the 1980s by Salton and Buckley (Salton & Buckley, 1988) which has proven to work well for information retrieval (Manning, Raghavan, Schütze, et al., 2008). IDF discounts features that occur in many documents of the corpus, and thus do not hold discriminative information. This can be both term counts and counts of extracted concepts. Let D be the set of documents, then the IDF re-weighted score for some term or concept w in a document $d \in D$ is defined as: $\text{TF-IDF}(w,d) = \text{TF}(w,d) \cdot \text{IDF}(w,D)$, where $\text{IDF}(t,D) = 1 + \log \frac{|D|+1}{|\{d \in D : w \in d\}|+1}$. To avoid division by zero, both the nominator and the denominator are incremented by one, as if there was one artificial document containing all possible terms and concepts. This can happen because the thesaurus covers all concepts but not the data itself. The fraction as a whole is as well incremented by one, to ensure that words that appear in all documents are not completely discarded. Okapi BM25 is an extension of IDF by Robertson et al. (Robertson, Walker, Beaulieu, & Willett, 1999) that slightly modifies the IDF term to include the average length of a document. It offers two hyper-parameters for interpolating the difference between the current document length and the corpus-wide mean document length. The literature suggests to use BM25 especially for fields with short texts using hyper-parameters $k = 1.6$ and

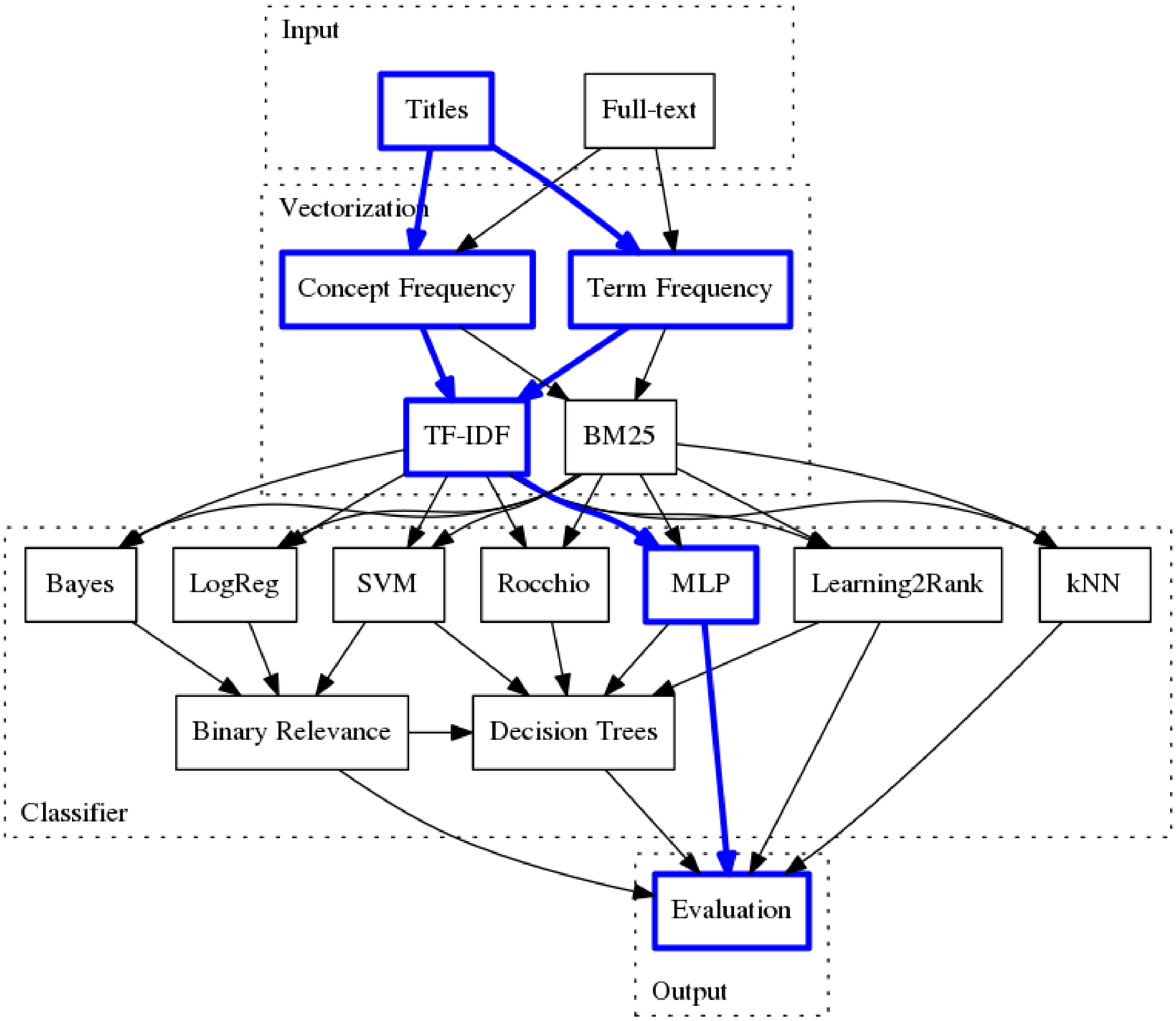


Figure 25: Illustration of the configurable text-processing pipeline used for our experiments. The pipeline starts with the vectorisation of the input text, followed by feature re-weighting, classification and evaluation. The emphasised edges and nodes show the most successful strategy applied to title data.

$b = 0.75$ (Manning, Raghavan, Schütze, et al., 2008). Hence, variants of our text vectorisation methods using BM25 instead of TF-IDF re-weighting are included in our comparison.

Combining terms and concepts After re-weighting by either inverse document frequency or BM25, the resulting vectors are normalised to unit length (with respect to the L2-norm). This leads to desirable invariance to document length. Besides using either the term frequency (TF) or the concept frequency (CF), we concatenate the respective feature vectors (CTF). Table 25 shows a complete list of considered text vectorisation methods.

Table 25: Overview of text vectorisation methods.

Name	Explanation
TF-IDF	Term Frequency (TF) re-weighted by inverse document frequency (IDF)
BM25	TF with BM25 re-weighting
CF-IDF	Concept Frequency (CF) with IDF
BM25C	CF with BM25 re-weighting
CTF-IDF	CF combined with TF (CTF) with IDF
BM25-CT	CTF with Okapi BM25

In the second step of the pipeline, a classifier is consulted to predict the desired set of labels based on the vector representation of the input text (compare Figure 25). Given training data, the classifiers have the opportunity to learn how to associate the features with the respective class labels. Lazy learners such as k-nearest neighbors or the Rocchio classifier merely copy their input at training time, shifting the main computational effort to test time. On the other hand, eager learners such as Naive Bayes, generalised linear models, and multi-layer perceptron, use the training data for adapting their parameters according to the correct classification result. Learning-to-rank algorithms for classification are typically a hybrid of the lazy and eager learning paradigms. Those ranking algorithms operate on the label set in the neighborhood of the current documents. In the following, we describe all of these classifiers in more detail.

Nearest neighbor classifier The most typical lazy-learning algorithm is kNN (Keller et al., 1985). All training examples are stored along with their class annotations. At test time, the k nearest neighbors with respect to some distance metric (we chose cosine) vote on class membership. For multi-label problems, variants are proposed that assign the union of label annotations in the neighborhood as well as conducting a separate vote for each label (Spyromitros et al., 2008). By auto-optimising the k hyperparameter for these methods, we found $k = 1$ to be the optimal value in our setting, as in our prior work (Große-Bölting et al., 2015). In this case all multi-label variants coincide to copy the label set from the nearest neighbor of the training set.

Rocchio classifier The Rocchio classifier (Joachims, 1997), or nearest-centroid classifier, resembles a lightweight modification of the nearest neighbor classifier. During training, only the centroid of each class is stored. The classification result is then determined by the nearest of these centroids at test time. In multi-label classification however, the classifier is only capable to return a ranked list of labels based on the distance to the respective centroids. As in the nearest neighbor classifier above, we use cosine distance as criterion.

Naive Bayes The Naive Bayes classifier (McCallum et al., 1998) is one of the most traditional classifiers for text classification tasks. We consider two Naive Bayes variants, multinomial and Bernoulli. In the multinomial variant, the features of term or concept frequencies are assumed to be generated by a multinomial distribution. The Bernoulli variant only takes the occurrences of (binary) features into account, which leads to penalising the non-occurrences of features. The Bernoulli variant is an intuitive approach for short text such as titles since duplicate words are rather infrequent, while the multinomial variant is more intuitive for full-texts. For both variants, we apply Lidstone-Smoothing with $\alpha = 10^{-5}$. The main drawback of Naive Bayes is the assumption of statistical independence among the input features.

Linear models Generalised linear models (J. A. Nelder, 1972) use the training examples to learn a decision boundary. This decision boundary is a separating hyperplane specified by a linear combination of the input features $\mathbf{w} \cdot \mathbf{x} - b = 0$. The parameters $\mathbf{w}$ and b are optimised to minimise the regularised training error: $\frac{1}{n}\sum_{i=1}^{n} J(y_i, y(\mathbf{x_i})) + \alpha R(\mathbf{w})$ where $y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$ is the model's output and $\alpha R(\mathbf{w})$ is a regularisation term on the model's weights such as the L2-norm. For the loss function J, we consider two variants: logistic loss $J_{\text{logistic}}(y, p) = \ln(1 + \exp(-p \cdot y))$ as in logistic regression (LR) and hinge loss $J_{\text{hinge}}(y, p) = \max(0, 1 - p \cdot y)$ as in linear SVM. At test time, the binary decision is determined by the side of the hyperplane, on which the documents in question fall. We employ stochastic gradient descent as an optimiser for these generalised linear models, which is known to yield good generalisation on large-scale datasets (T. Zhang, 2004; Bousquet & Bottou, 2008; Bottou, 2010). We apply the learning rate schedule $\eta^{(t)} = \frac{1}{\alpha \cdot (t_0 + t)}$, where t_0 is chosen by a heuristic of Léon Bottou (Bottou, 2012). We average the weights $\mathbf{w}$ over time, which allows higher learning rates and leads to faster convergence (Bottou, 2012). In this setting, we empirically determined $\alpha = 10^{-7}$ to be a good hyper-parameter value for all datasets (in the range $10^{-1}, 10^{-2}, \ldots, 10^{-9}$). This leads to comparatively high initial learning rates and low regularisation.

Learning to rank Learning to Rank (L2R) refers to a set of techniques that can be used to learn the ranking of a list from training data. As suggested by Huang et al. (M. Huang et al., 2011), we restrict the supplied list to those labels that occur in the k neighboring documents (we empirically determined $k = 45$). Those labels, that are also assigned to the current document in question should be ranked higher than the others. To learn the ranking, we use the neighborhood, overlap, and translation-probability features proposed by Huang et al. (M. Huang et al., 2011). Hence, at test time the union of labels among the k nearest neighbors are ranked via the learned parameters. However, the algorithm itself does not offer the possibility of hard decisions on label assignments. Thus, we chose to cut off the ranked list at the position of the average number of assigned labels in the training data. In our experiments, we made use of the RankLib library[69] and found LambdaMART(Wu et al., 2010) to outperform other list-wise L2R algorithms.

Multi-layer perceptron As representative for the neural network family, we employ a fully connected feed-forward neural network with one hidden layer, a so-called Multi-Layer Perceptron (MLP) (Hecht-Nielsen, 1988; Nam, Kim, Mencía, Gurevych, & Fürnkranz, 2014). Compared to the linear models, the MLP has an additional intermediate hidden layer h with a nonlinear activation function f. Thus, we first compute $h = f\left(W^1 x + b^1\right)$, and then $y = W^2 h + b^2$. The output y is then scaled to the interval $(0,1)$ by the sigmoid function σ as in logistic regression and compared to the gold-standard by cross-entropy. The gradient for updating the parameters is computed by the chain-rule, also known as back-propagation. The optimisation itself is carried out by Adam (Kingma & Ba, 2014) with the default hyper-parameters and $\alpha = 0.01$. We chose a hidden layer size of 1000 and use rectified linear units (Nair & Hinton, 2010) as activation function f (except for the NYT dataset where we use $\tanh$ due to numerical difficulties). For regularisation, we apply dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) with a probability of 0.5. The intermediate hidden layer can be regarded as a fine-tuned task-specific word embedding, which enables the classifier as a whole to learn nonlinear relationships among the features. To convert the odds $\sigma(y)$ into a binary decision, several approaches suggest to use a threshold learning technique (Nam et al., 2014; Tang, Rajan, & Narayanan, 2009). In our initial experiments, however, we experienced that the most recent threshold learning technique yields rather unsatisfactory results in terms of the F_1 measure. Instead, we use a fixed threshold of 0.2.

Some of the learning algorithms are only designed for single-label classification (SVM, logistic regression, Naive Bayes), others do only return a ranked list of possible labels (kNN, Rocchio, Learning to Rank). In the following, we depict multi-label adaption strategies for converting those single-label classifiers and ranking algorithms into multi-label classifiers.

Binary relevance Linear models as well as Naive Bayes are restricted to mutually exclusive class assignments by design. Only one class out of all possible ones is selected. In multi-label classification, however, multiple labels need to be assigned. The most common approach to adapt such classifiers, also known as one-vs-all or one-vs-rest, is to train one classifier per class, which distinguishes its respective class from all others, i. e., decides for binary relevance (Tsoumakas & Katakis, 2007). The training documents are supplied to all label-specific classifiers. Depending on the prevalence of the label that corresponds to the respective classifier, the example is either treated as positive or as negative. At test time, the classification result is composed of the binary decisions for each label.

Classifier stacking Multi-value classification stacking (Heß, Dopichaj, & Maaß, 2008) refers to a technique where the final classification result is composed by two classifiers. The so-called base-classifier returns a ranked list of label predictions with confidence scores. Then for each class, a meta-classifier takes these confidence scores along with the position in the ranked list as input and outputs a binary decision for the specific class. This technique enables transforming any classifier that returns confidence scores into a multi-label classifier. As meta-classifiers, we use decision trees with Gini impurity as splitting criterion (Breiman, Friedman, Olshen, & Stone, 1984). To limit complexity, we generate training data only for those meta-classifiers, whose class is among the top 30 of the base-classifier's ranking (Heß et al., 2008). We use this decision tree module as an alternative to hard cut-offs in Learning to Rank, and the fixed thresholds in multi-layer perceptrons. For comparison with the original work of Heß et al. (Heß et al., 2008), we also consider Rocchio as a base-classifier. We furthermore experiment with applying the decision tree module on top of binary-relevance logistic regression. We abbreviate this decision tree module with the suffix DT.

3.7.3 Experimental evaluation and comparison

In the following, we first depict the characteristics of the four datasets before we describe the experimental procedure. The procedure comprises a two-step approach: vectorisation and classification.

3.7.3.1 Datasets

We have conducted our experiments on four datasets of English documents: two datasets are obtained from scientific digital libraries in the domains of economics and political sciences along with two news datasets from Reuters and New York Times. Table 26 summarises the basic statistics of the datasets. For each document in

[69] `https://people.cs.umass.edu/~vdang/ranklib.html`

the datasets, there are manually created gold-standard annotations provided by respective domain experts, who work as professional subject indexers in the corresponding organisations. In addition, each dataset provides a domain-specific thesaurus that serves as controlled vocabulary of the gold-standard. Its concepts are used as target labels in our multi-label document classification task. The thesaurus also offers sets of concept-specific phrases (i. e., `skos:prefLabel` and `skos:altLabel` in case of SKOS format) that are used for concept extraction from the documents' full-text and titles (Goossen et al., 2011).

The economics dataset EconBiz consists of 62,924 documents and is provided by ZBW — Leibniz Information Centre for Economics. The annotations are taken from the Standard Thesaurus Wirtschaft (STW) version 9[70], which is a controlled domain-specific thesaurus for economics and business studies maintained by ZBW. The thesaurus contains 6,217 concepts with 12,707 concept-specific phrases. From these concepts, 4,682 are used in the corpus and thus considered in the multi-label classification task. Each document is annotated by domain experts with an average of 5.26 labels, and a standard deviation (SD) equal to 1.84.

The political sciences dataset IREON has 28,324 documents. Similar to the economics dataset, we made a legal agreement for the political sciences dataset with the German Information Network for International Relations and Area Studies[71] that is providing the documents. The labels are taken from the thesaurus for International Relations and Area Studies[72], which contains 9,255 concepts (and an equivalent number of concept-specific phrases, i. e., there are no synonymous phrases). From these concepts, 7,234 are used in the corpus. Each document in the dataset has on average 8.07 labels (SD: 3.03).

The Reuters RCV1-v2 dataset contains 805,414 articles. We chose articles where both the titles and the full-text of the documents are available. From this set of documents, we randomly selected 100,000 articles to match the scale of the scientific corpora. In our experiments, we employ the thesaurus re-engineered from the Reuters dataset by Lewis et al. (Lewis, Yang, Rose, & Li, 2004). The thesaurus contains 117 concepts and a total of 173 concept-specific phrases. From these concepts, 101 are used in the corpus. Each document was annotated with on average 3.21 labels (SD: 1.41).

The New York Times Annotated Corpus Dataset (NYT) contains 1,846,656 articles. Each article has two sets of annotations, consisting of annotations created by a professional indexing service and annotations which were added by the authors using a semi-automatic system. We used the annotations provided by the indexing service because it is reasonable to expect that they are more consistent and of higher quality (Heß et al., 2008). As for the Reuters dataset, we chose a random subset of 100,000 documents containing both full-text and titles. The number of concepts in the NYT dataset is 25,226. From these concepts, 6,809 are used in our random sample. Each document is annotated with on average 2.53 labels (SD 1.78). Like the political sciences dataset, each concept consists of only a single specific phrase.

Table 26: Statistics for the datasets: $|D|$ documents, $|C|$ concepts in the thesaurus, $|L|$ labels assigned in the dataset, d/l mean documents per label, l/d mean labels per documents along with median l/d_{50}, V vocabulary size, w/d mean terms per document, and c/d mean concepts per document

	EconBiz	**IREON**	**RCV1**	**NYT**
$\|D\|$	62,924	27,576	100,000	100,000
$\|C\|$	6,217	9,255	117	25,226
$\|L\|$	4,682	7,234	101	6,809
d/l	70.8 (322.9)	32.6 (116.8)	3174.9 (6371.3)	37.1 (213.0)
l/d	5.26 (1.84)	8.57 (3.03)	3.21 (1.41)	2.53 (1.78)
l/d_{50}	4	5	14	2
V_{title}	19,579	15,419	32,859	40,736
w/d_{title}	7.07 (3.03)	8.13 (5.29)	12.21 (2.39)	4.46 (2.25)
c/d_{title}	3.33 (1.83)	3.69 (2.36)	0.57 (1.02)	0.70 (0.83)
V_{full}	1,340,628	1,165,919	155,339	270,710
w/d_{full}	6,750 (6,854)	11,255 (15,565)	136 (114)	310 (294)
c/d_{full}	247 (121)	346 (189)	6.80 (8.60)	37.0 (38.2)

[70] http://zbw.eu/stw/versions/9.0/about.en.html
[71] http://www.fiv-iblk.de/eindex.htm
[72] http://www.fiv-iblk.de/information/information_thesaurus.htm

3.7.3.2 Procedure

Vectorisation methods We compare the different vectorisations of the input text as shown in Figure 25. One vectorisation is based on term frequencies (TF-IDF) and the other is based on concept frequencies (CF-IDF). We experiment with the re-weighting method BM25 using term frequencies and BM25C using concept frequencies. The concatenation of both terms and concepts is denoted by CTF-IDF and BM25CT, respectively. As classifier, we employ kNN with cosine distance. The performance of kNN relies on the assumption that documents are well represented by the features and that similar documents have similar labels. Therefore, its classification performance is a good indicator for the quality of the features.

Classification methods After determining the best-performing vectorisation method, we compare lazy learning as well as eager learning classifiers of combined with multi-label adaption methods, where appropriate. We leverage the linear models (SVMs and logistic regression) to perform multi-label classification with binary relevance, i. e., training one classifier per label. To adapt the Learning to Rank approach and the multi-layer perceptron to multi-labeling, we consider using thresholds as well as stacking with decision trees. We also experiment with stacking the decision tree module on top of binary-relevance logistic regression. Careful tuning of the hyper-parameters is crucial to the success of machine learning algorithms, especially in those multi-label classification tasks, where only few training examples are available per class. Striving to identify well-suited hyper-parameters that are invariant to the concrete dataset, we keep all hyper-parameters fixed across all experiments and datasets.

Preprocessing Prior to counting terms and extracting concepts, both the input text and the concept-specific phrases of the thesauri are subject to preprocessing steps. This includes discarding all characters except for sequences of alphabetic characters with a length of at least two. Words connected with a hyphen are joined (i. e., the hyphen is removed). Detected words were lower-cased and lemmatised based on the morphological processing of WordNet[73].

Evaluation For evaluation, we separate each dataset into 90% training documents and 10% test documents and perform a 10-fold cross-validation, such that each document occurs exactly once in the test set. Hence for each test document, we compare the predicted labels with the label set of the gold standard and evaluate the F_1 measure. The F_1 measure is the harmonic mean between precision, i. e., true positives w.r.t false positives, and recall, i. e., true positives w.r.t false negatives. When no label is predicted, the precision is set to zero. The F-scores are then averaged over the test documents. We chose this sample-based F_1 measure over class-averaged or global variants because it is closest to an assumed application, where each individual document needs to be annotated as good as possible. It has to be emphasised that there is a possibility that all documents annotated with a specific label fall only into one test set. Despite no training data is available for these labels, we do not exclude those from our evaluation metric. Finally, we report the mean sample-based F-score over the 10-fold cross-validation.

We now describe the results of our experiments. Due to the high amount of possible pipeline configurations, we applied a step-by-step approach. For both the text vectorisation step and the classification step, we search for a local optimum solution to find the best overall classification strategy.

3.7.3.3 Results

Vectorisation methods Table 27 shows the results for the text vectorisation experiment. The term-based vectorisation method TF-IDF performed consistently better than the purely concept-based vectorisation CF-IDF methods on both the titles and the full-text. The difference ranges from 0.003 on economics to 0.307 F-score on Reuters. When combining the term vector with the concept vector, the performance is at least as good as the other text vectorisation methods and in many cases yields better results. This is more noticeable on titles than on full-texts. BM25 re-weighting does not improve the results compared to TF-IDF neither in case of the titles nor the full-text. Rather, we observe a decrease in performance by up to 0.13. The experiment using a nearest neighbor classifier indicates that CTF-IDF is the best-suited vectorisation method. Henceforth, we use CTF-IDF for comparing the performance of the classifiers.

Classifiers The results of comparing the different classifiers are documented in Table 28. As shown in the table, Bernoulli Bayes has a slight advantage over multinomial Bayes for titles. On the other hand, the multinomial variant has a slight disadvantage on full-texts. However, both methods consistently fall far behind

[73] http://wordnet.princeton.edu

Table 27: Sample-averaged F-scores of the text vectorisation methods with using kNN as common classifier.

Input	Vectorisation	EconBiz	IREON	RCV1	NYT
Full-text	TF-IDF	0.406	0.269	0.758	0.394
Full-text	BM25	0.370	0.230	0.740	0.370
Full-text	CF-IDF	0.402	0.266	0.451	0.367
Full-text	BM25C	0.296	0.161	0.423	0.236
Full-text	CTF-IDF	**0.411**	**0.272**	**0.761**	**0.406**
Full-text	BM25CT	0.377	0.231	0.742	0.379
Titles	TF-IDF	0.351	0.201	0.709	0.238
Titles	BM25	0.349	0.196	0.687	0.230
Titles	CF-IDF	0.303	0.183	0.275	0.105
Titles	BM25C	0.304	0.172	0.193	0.073
Titles	CTF-IDF	**0.368**	**0.212**	**0.717**	**0.242**
Titles	BM25CT	0.364	0.208	0.693	0.239

Table 28: Sample-averaged F-scores for classification methods with the best vectorisation method CTF-IDF. The suffix -DT means the approach is stacked with decision trees a meta-classifier (see Section 3.7.2).

Input	Classifier	EconBiz	IREON	RCV1	NYT
Full-text	kNN (baseline)	0.411	0.272	0.761	0.406
Full-text	Bayes (Bernoulli)	0.318	0.191	0.657	0.281
Full-text	Bayes (multinomial)	0.235	0.207	0.703	0.349
Full-text	SVM	0.481	0.319	0.852	0.554
Full-text	LR	0.485	0.322	0.851	0.556
Full-text	L2R	0.431	0.328	0.727	0.435
Full-text	MLP	**0.519**	**0.373**	**0.857**	0.569
Full-text	RocchioDT	0.291	0.225	0.645	0.393
Full-text	LRDT	0.498	0.339	0.843	0.562
Full-text	L2RDT	0.415	0.280	0.751	0.421
Full-text	MLPDT	0.492	0.340	**0.857**	**0.578**
Titles	kNN	0.368	0.212	0.717	0.242
Titles	Bayes (Bernoulli)	0.301	0.179	0.708	0.233
Titles	Bayes (multinomial)	0.254	0.178	0.699	0.214
Titles	SVM	0.426	0.272	0.804	0.325
Titles	LR	0.429	0.274	0.803	0.326
Titles	L2R	0.419	0.296	0.699	0.296
Titles	MLP	**0.472**	**0.309**	**0.812**	0.332
Titles	RocchioDT	0.335	0.219	0.584	0.252
Titles	LRDT	0.451	0.279	0.796	**0.353**
Titles	L2RDT	0.428	0.261	0.730	0.25
Titles	MLPDT	0.457	0.277	0.808	0.340

kNN on full-texts. In the case of working with titles, the Bayes classifiers are able to keep up with kNN on two datasets. RocchioDT's scores are depending on the datasets and range from the lowest (Reuters) to a score only slightly different from kNN (NYT, political sciences). The generalised linear models SVM and logistic regression are close to each other. The difference is no more than 0.04 for any dataset. Considering L2R, we observe that the technique yields consistently lower scores than the multi-layer perceptron. Overall, the eager learners SVM, LR, L2R and MLP outperform both Naive Bayes and the lazy learners RocchioDT, and kNN. Among all classifiers, MLP dominates on all datasets apart from NYT on titles, where LRDT achieves a 0.021 higher score. While the stacked decision tree module increases the F-scores of logistic regression on all datasets with fewer than 100 documents per label (all but Reuters), the impact of the stacking method is inconsistent for the Learning to Rank and MLP approaches. It is noteworthy that there are cases where a classifier performs better on the title data than the same classifier applied on the full-text data. These are Bernoulli Bayes on the Reuters dataset and RocchioDT on the economics dataset. As a general rule, however, full-texts generate higher scores than the titles. Comparing different classifiers across titles and full-text, we can make the observation that some classifiers trained on titles outperform others that were trained on the full-text. Apart from the NYT corpus, the eager learners LR, LRDT and MLP on titles are superior to kNN on full-texts. Finally, we compare the F-scores of the best-performing multi-layer perceptron on titles with its scores obtained on full-text. On the NYT dataset, 58% of the F-score is retained when using only titles. On the political sciences and economics datasets, the retained F-score is 83% and 91%, respectively. On the Reuters dataset, the MLP using solely titles retains 95% of the F-score that is obtained with full-text information available.

3.7.3.4 Discussion

The results show that multi-label classification of text documents can be reasonably conducted using only the titles of the documents. Over all datasets, the multi-layer perceptron on titles retains 82% of the F-score obtained on full-text. This gives an empirical justification for the value of automated semantic document annotation using metadata. From the first experiment, we find that combining words with extracted concepts as features is preferable over one of them alone. Concepts hold valuable domain-specific semantic information. The term frequency, on the other hand, holds implicit information which is as well important for correct classification. Eager learners are, by design, capable of learning which terms or concepts need to be associated to the respective class. The results show that also lazy learners benefit from this joint representation. The second experiment shows that eager learners such as logistic regression and MLP consistently outperform lazy learners for multi-label classification. This result extends recent advancements in multi-labeling (Nam et al., 2014; Tanaka, Nozawa, Macedo, & Baranauskas, 2015) towards document classification scenarios with many possible output labels and only few examples per class.

Inspecting the results for titles and full-text, the best-performing classifiers still perform better on the full-text. This is not surprising since the full-text holds considerably more information (including the title). However, for all datasets apart from the NYT dataset, the difference in F-score of the best-performing MLP is small. The difficulties in classifying the documents in the NYT dataset can be explained by a characteristic that the titles consist only of 4 words on average. There may be a lower bound on the title length to perform the classification task, since a short title limits the amount of available information and thus prohibits discrimination. From the other datasets, we can state that an average of 7 words per title leads to at least 80% retained F-score. Thus, it would require further investigation to understand the specific influence of the title length on the classification performance. The complexity of a multi-labeling problem depends on the number of available documents per label, independent of whether the full-text or the titles are used. Especially binary-relevance classifiers suffer from conservative label assignments (high precision, low recall), when many negative examples and only few positive examples are presented during training. While the results of the stacked decision tree module are inconsistent for MLP and L2R, it does alleviate the conservative assignments problem of binary-relevance, when only few documents per label are available.

In our experiments over four large-scale real-world corpora covering a broad range of domains (economics, political sciences and news), we did not limit the complexity by excluding rare labels and kept all independent variables as well as hyperparameters fixed. In our prior work (Große-Bölting et al., 2015), we have used the thesaurus hierarchy to model label dependencies which improves the classifications obtained by kNN. Despite not making use of the hierarchy anymore, we are able to achieve even higher absolute F-scores using eager learning techniques and supplying term features in addition to extracted concepts. We can therefore drop the constraint of a hierarchical organisation among the labels. Due to this minimal amount of requirements and invariant configurations of the text processing pipeline, we can expect our findings to generalise to a wide range of other corpora.

To validate the practical impact of the experimental results, we have conducted a qualitative assessment of the experimental results in an expert workshop with three subject indexing specialists at ZBW, the national library for economics in Germany. The experts state that titles can be sufficient for classification of scientific documents. They further noted that titles contain less information than what an intellectual indexer has available when manually conducting the classification tasks for the documents. They also pointed out that researchers carefully chose their titles for findability. The experts argued that reasonably good automatic indexing based on titles is valuable since it does not raise legal problems compared to processing full-text as discussed in the introduction. We conclude that using the documents' title for automated semantic annotation is not only technically possible with a high quality but also valuable from a practical point of view.

In summary, we have shown that it is reasonable to conduct semantic annotations of documents by just analyzing the titles. Our experiments show that by using titles, a performance of over 90% can be reached w.r.t to the classification performance obtained when using the full-text of the documents. This opens many new possibilities for using document classification even when only little input data is available such as titles obtained from the documents' metadata on the Linked Open Data cloud.

3.8 Video processing

3.8.1 Problem statement

In deliverable D3.1 (Blume et al., 2017), we presented the best set of video processing technologies of MOVING, which include technologies for video annotation and transcript analysis. In this deliverable we build on this initial set of technologies and extend them by introducing (1) a new machine learning method for concept annotation with complex concept labels, that will be applied on non-lecture videos and (2) a first approach to the problem of performing temporal fragmentation on the lecture videos based on their transcripts. Furthermore, we report additional software improvements to the Video Analysis service (VIA).

3.8.2 Method description for video annotation

Video concept detection is about deciding whether a certain video depicts a given concept or not. A typical complex concept (also referred to as an event) is an interaction between humans, or between humans and physical objects (Y.-G. Jiang, Bhattacharya, Chang, & Shah, 2013). Building a concept detector usually starts with the sampling of the given raw videos such that a number of keyframes are extracted at regular time intervals. Then, each keyframe is represented using various static, such as Scale-Invariant Feature Transform (SIFT) (Lowe, 1999) and Speeded Up Robust Features (SURF) (Bay, Tuytelaars, & Van Gool, 2006), or motion visual descriptors, such as Histogram Of Gradient (HOG), Histogram Of Flow(HOF) and dense trajectories (Wang & Schmid, 2013). Furthermore, recently, the use of Deep Convolutional Neural Network (DCNNs) has been shown to be very effective for video understanding problems and, thus, has been also used for the problem of video concept detection (Z. Xu, Yang, & Hauptmann, 2015). The majority of the learning methods employed in video concept detection problem do not address the uncertainty in the training data explicitly. That is, firstly, each training example is assumed to be described by a fixed position in some vector space (feature representation). However, such an approach does not account for the fact that the underlying process of extracting the feature representation may be imperfect or noisy, hence introducing some degree of uncertainty to the generated features. We deal with this by introducing an algorithm that learns from uncertain data (i.e., data that are characterised by uncertainty in their input-space representation).

Uncertainty estimation Let us define a set $\mathscr{X}$ of ℓ annotated random vectors representing the video-level feature vectors. Each random vector is assumed to be distributed normally; i.e., for the random vector representing the i-th video, $\mathbf{X}_i$, we have $\mathbf{X}_i \sim \mathscr{N}(\mathbf{x}_i, \Sigma_i)$. That is, $\mathscr{X} = \{(\mathbf{x}_i, \Sigma_i, y_i) \colon \mathbf{x}_i \in \mathbb{R}^n, \Sigma_i \in \mathbb{S}^n_{++}, y_i \in \{\pm 1\}, i = 1, \ldots, \ell\}$. For each random vector $\mathbf{X}_i$, a number, N_i, of observations, $\{\mathbf{x}_i^t \in \mathbb{R}^n \colon t = 1, \ldots, N_i\}$ are available (these are keyframe-level vectors that have been computed for each video; details on how these have been computed in our experiments are given in Section 3.8.4). Then, the sample mean vector and the sample covariance matrix of $\mathbf{X}_i$ are computed. However, the number of observations per each video that are available for our dataset is in most cases much lower than the dimensionality of the input space. Consequently, the covariance matrices that arise are typically low-rank; i.e. $\operatorname{rank}(\Sigma_i) \leq N_i \leq n$. To overcome this issue, we assumed that the desired covariance matrices are diagonal. That is, we require that the covariance matrix of the i-th training example is given by $\widehat{\Sigma_i} = \operatorname{diag}\left(\hat{\sigma}_i^1, \ldots, \hat{\sigma}_i^n\right)$, such that the squared Frobenious norm(Golub & Van Loan, 1996) of the difference $\Sigma_i - \widehat{\Sigma_i}$ is minimum. That is, the estimator covariance matrix $\widehat{\Sigma_i}$ must be equal to the diagonal part of the sample covariance matrix Σ_i, i.e. $\widehat{\Sigma_i} = \operatorname{diag}\left(\sigma_i^1, \ldots, \sigma_i^n\right)$. We note that, using this approximation approach, the covariance matrices are diagonal but anisotropic and different for each training input example. This is in contrast with other methods, such as (W. Zhang, Stella, & Teng, 2012; Bi & Zhang, 2004; Qi, Tian, & Shi, 2013), which assume more restrictive modeling for the uncertainty (e.g., isotropic noise for each training sample).

Linear SVM with Gaussian Sample Uncertainty In order to exploit input uncertainty in the problem of video concept detection, we use the Linear Support Vector Machine with Gaussian Sample Uncertainty (LSVM-GSU) (Tzelepis, Mezaris, & Patras, 2017) for the problem of learning under input uncertainty. Let us briefly present the LSVM-GSU's (Tzelepis et al., 2017) optimisation problem. LSVM-GSU is a maximum-margin classifier that takes as input training data that are described not solely by a set of feature representations, i.e. a set of vectors $\mathbf{x}_i$ in some n-dimensional space, but rather by a set of multi-variate Gaussian distributions which model the uncertainty of each training example. That is, every training datum is characterised by a mean vector $\mathbf{x}_i \in \mathbb{R}^n$ and a covariance matrix $\Sigma_i \in \mathbb{S}^n_{++}$. LSVM-GSU is obtained by minimising, with respect to $\mathbf{w}$, b, the objective function $\mathscr{J} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ given by

$$\mathscr{J}(\mathbf{w},b) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{1}{\ell}\sum_{i=1}^{\ell}\mathscr{L}(\mathbf{w},b), \tag{3}$$

where ℓ is the number of training data, $\mathbf{w}^\top \mathbf{x} + b = 0$ denotes the separating hyperplane, and the loss $\mathscr{L} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is given by

$$\mathscr{L}(\mathbf{w},b) = \frac{d_{\mathbf{x}_i}}{2}\left[\operatorname{erf}\left(\frac{d_{\mathbf{x}_i}}{d_{\Sigma_i}}\right)+1\right] + \frac{d_{\Sigma_i}}{2\sqrt{\pi}}\exp\left(-\frac{d^2_{\mathbf{x}_i}}{d^2_{\Sigma_i}}\right), \tag{4}$$

where $d_{\mathbf{x}_i} = 1 - y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right)$ and $d_{\Sigma_i} = \sqrt{2\mathbf{w}^\top \Sigma_i \mathbf{w}}$. As shown in (Tzelepis et al., 2017), the above objective function is convex with respect to $\mathbf{w}$ and b; therefore, we can use a Stochastic Gradient Descendant (SGD) algorithm (Tzelepis et al., 2017) for solving the corresponding optimisation problem. Since the objective function is convex, we can obtain the global optimal solution. For further details on the theory behind this machine learning method, the interested reader is referred to (Tzelepis et al., 2017). Details on how this method was used in experiments on the problem of video annotation are available in Section 3.8.4.

3.8.3 Method description for lecture video fragmentation

As multimedia based e-learning systems and online video-lecture databases grow up, accessing and searching in these video content is very tricky and inefficient due to the extremely homogeneous visual content of those videos. Also, the huge amount of multimedia content makes manual indexing prohibitive in practice. Video lecture fragmentation is the problem of segmenting lecture videos in logical, meaningful parts, easy to access, in order to generate annotations that enable searching and indexing of video parts, using knowledge that can be extracted from these parts. Lecture video fragmentation differs from the classic video segmentation approaches since there are no visual changes, e.g. shot or scene alterations, in those videos. Therefore, the visual content of a lecture video is not suitable for solving the problem, and for that reason previous works as well as our approach exploit the audio and textual content of the lecture videos.

Our lecture video fragmentation method is based on the transcripts (SRT) of the lecture video, that can be easily generated by an automatic speech recognition system. We determine meaningful blocks of SRT by calculating the corresponding boundaries. Following state-of-the-art approaches on video lecture segmentation (Lin, Chau, Cao, & Nunamaker Jr, 2005)(Shah, Yu, Shaikh, & Zimmermann, 2015), ad-hoc video search (Markatopoulou, Galanopoulos, Mezaris, & Patras, 2017), and including word embedding features (Mikolov, Chen, et al., 2013), we build a novel video lecture fragmentation system. In order to determine segment boundaries we used standard NLP techniques such as stop-word removal and text cleaning. Then the Stanford Part-Of-Speech (POS) tagger (Toutanova, Klein, Manning, & Singer, 2003) is used for noun phrases and phrase extraction and the Stanford Named Entity Recognizer (NER)(Finkel, Grenager, & Manning, 2005) for named entity extraction (e.g. names, organisations etc). To extract phrases from subtitles we used the NLP rules from (Markatopoulou et al., 2017). All the above procedures resulting a set of topics that comprises words and phrases.

A sliding window (W_i) of 120 words moves across the entire text with a certain step (e.g. 20 words) and we calculate the similarity between two neighboring windows (W_i W_{i+1}). To find the similarity we utilised the standard formula for cosine similarity $(V_a * V_b)/(\|V_a\| + \|V_b\|)$ where V_a and V_b are the feature vectors of the corresponding windows. As a feature vector we adopt two different representations. The first is the TF-IDF weighting of the extracted topics that are induced in the corresponding window, resulting in a vector $f(x) \in \mathbb{R}^{<Total\_number\_of\_topics>}$, while the second approach utilises the word2vec embedding (Mikolov, Chen, et al., 2013), forming a semantic vector $f(x) \in \mathbb{R}^{300}$. The result of the similarity calculation across the entire transcript is a similarity graph in where the X-axis represents time and the Y-axis represents the two neighboring windows similarity value. We determined the valleys and the peaks of this graph and the deepest valleys are assigned as candidates for segment boundary. Finally the valleys with values larger than $C = \mu - \sigma$, where μ is the mean of the values of all valleys and σ is the standard deviation, are selected as the actual segment boundaries.

3.8.4 Experimental evaluation and comparison

Experiments on video annotation In our experiments on video complex concept detection we used datasets from the challenging TRECVID MED task (Over et al., 2015). For training, we used the MED 2015 training dataset consisting of the "pre-specified" video subset (2,000 videos, 80 hours) and the "event background"

(Event-BG) video subset ($5,000$ videos, 200 hours). For testing, we used the large-scale "MED14Test" dataset (Over et al., 2015; L. Jiang, Yu, Meng, Mitamura, & Hauptmann, 2015) (roughly $24,000$ videos, 850 hours). Each video in the above datasets belongs to, either one of 20 target complex concept classes, or to the "rest of the world" (background) class. More specifically, in the training set, 100 positive and $5,000$ negative samples are available for each complex concept class, while the evaluation set includes only a small number of positive (e.g., only 16 positives for concept E021, and 28 for E031) and approximately $24,000$ negative videos.

For video representation, approximately 2 keyframes per second were extracted from each video. Each keyframe was represented using the last hidden layer of a pre-trained deep convolutional neural network (DCNN). More specifically, a 22-layer inception style network, trained according to the GoogLeNet architecture (Szegedy et al., 2015), was used. This network had been trained on various selections of the ImageNet "Fall 2011" dataset and provides scores for $5,055$ concepts (Russakovsky et al., 2015).

We experimented using two different feature configurations. First, we used the mean vectors and covariance matrices as computed with the method discussed above. Furthermore, in order to investigate the role of variances in learning with baseline LSVM, we constructed mean vectors and covariance matrices as shown in Table 29, where σ_0 is typically set to a small positive constant (e.g., 10^{-6}) indicating very low uncertainty for the respective features.

Table 29: Mean vector and covariance matrix of the *i*-th example for feature configurations 1 and 2 of the video concept detection experiments.

Configuration	Mean vector	Covariance matrix
Configuration 1	$\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})^\top \in \mathbb{R}^n$	$\Sigma_i, = \operatorname{diag}\left(\sigma_i^1, \ldots, \sigma_i^n\right) \in \mathbb{S}_{++}^n$
Configuration 2	$\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n}, \sigma_i^1, \ldots, \sigma_i^n)^\top \in \mathbb{R}^{2n}$	$\Sigma_i, = \operatorname{diag}\left(\sigma_i^1, \ldots, \sigma_i^n, \sigma_0, \ldots, \sigma_0\right) \in \mathbb{S}_{++}^{2n}$

For both feature configurations, Table 30 shows the performance of the proposed LSVM-GSU in terms of average precision (AP) (Over et al., 2015; Tzelepis, Gkalelis, Mezaris, & Kompatsiaris, 2013) for each target complex concept in comparison with LSVM, Power Support Vector Machine (PSVM) (W. Zhang et al., 2012), and LSVM-iso approaches. Moreover, for each dataset, the mean average precision (MAP) across all target concepts is reported. The optimisation of the λ parameter for the various SVMs was performed using a line search on a 10-fold cross-validation procedure. The bold-faced numbers indicate the best result achieved for each complex concept class. We also report the results of the McNemar (McNemar, 1947; Gkalelis, Mezaris, Kompatsiaris, & Stathaki, 2013), statistical significance tests. A $*$ denotes statistically significant differences between the proposed LSVM-GSU and baseline LSVM, a $\diamond$ denotes statistically significant differences between LSVM-GSU and PSVM, and a $\sim$ denotes statistically significant differences between LSVM-GSU and the LSVM extension for handling isotropic uncertainty (LSVM-iso).

From the results obtained, we observe that the proposed algorithm achieved better detection performance than LSVM, PSVM, and LSVM-iso, in both feature configurations. For feature configuration 1, the proposed LSVM-GSU achieved a relative boost of 22.2% compared to the baseline standard LSVM and 19.4% compared to Power SVM, while for feature configuration 2 respective relative boosts of 12.7% and 11.7%, respectively, in terms of MAP. We also experimented directly using the samples from which the covariance matrix of each example was estimated and obtained inferior results; that is, a MAP of 10.15%, compared to LSVM's 14.78% and 18.06% of the proposed SVM-GSU.

Experiments on lecture video fragmentation The formal evaluation of our lecture video segmentation approach is an ongoing process and for this reason no detailed evaluation results can be reported in this deliverable. However, in Figure 26 a sample video is presented with the extracted segments and the corresponding key terms. It is clear that the visual content is extremely homogeneous throughout the whole video and therefore could not be used for segmentation. However, by exploiting the textual information of the video transcripts, our method segments the video in meaningful fragments in which different topics are discussed.

3.8.5 Implementation, APIs and integration

The video processing capabilities are harnessed by the MOVING platform through the Video Analysis service (VIA). The VIA is an external REST web service capable of downloading videos from different locations in the web and performing visual analysis such as temporal fragmentation and concept detection on them. The overall service functionality is described in detail in the previous deliverables D3.1 (Blume et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017).

Table 30: Complex concept detection performance (AP and MAP) of the linear SVM-GSU compared to the baseline linear SVM, Power SVM, and a LSVM extension for handling isotropic uncertainty using the MED15 (for training) and MED14Test (for testing) datasets.

Complex Concept	**Feature Configuration 1 (5055-D)**				
	LSVM	PSVM	LSVM-iso	LSVM-GSU (proposed)	McNemar Tests
E021	0.0483	0.0510	0.0500	**0.0515**	∗,⋄,∼
E022	0.0227	0.0310	**0.0350**	0.0277	∗,⋄,∼
E023	0.4159	0.4515	**0.6059**	0.6057	∗,⋄
E024	0.0071	0.0081	0.0097	**0.0105**	⋄
E025	0.0052	0.0052	**0.0074**	0.0068	
E026	0.0457	0.0459	0.0606	**0.0608**	⋄
E027	**0.1319**	0.1424	0.1174	0.1219	∗,⋄,∼
E028	0.4242	0.4125	0.3819	**0.4335**	∗,⋄,∼
E029	0.0812	0.0914	**0.1793**	0.1791	⋄
E030	0.0516	0.0551	0.0877	**0.0884**	
E031	0.4416	0.4425	0.4480	**0.4796**	∗,⋄,∼
E032	0.0280	0.0400	0.0870	**0.1196**	∗,⋄,∼
E033	0.3483	0.3614	0.3901	**0.4187**	∗,∼
E034	0.0583	0.0588	0.0599	**0.0614**	⋄
E035	0.3330	0.3419	**0.3500**	0.3369	∗,⋄,∼
E036	**0.0894**	0.0748	0.0695	0.0704	⋄
E037	0.0884	0.0880	**0.1981**	0.1968	∗,⋄,∼
E038	0.0261	0.0241	0.0212	**0.0291**	⋄
E039	0.2677	0.2698	**0.2959**	0.2757	∗,⋄,∼
E040	0.0421	0.0315	0.0375	**0.0377**	∗,⋄
MAP	0.1478	0.1513	0.1746	**0.1806**	–

Complex Concept	**Feature Configuration 2 (10110-D)**				
	LSVM	PSVM	LSVM-iso	LSVM-GSU (proposed)	McNemar Tests
E021	0.0829	0.0834	**0.1074**	0.0778	⋄,∼
E022	0.0674	0.0773	0.1023	**0.1429**	∗,⋄,∼
E023	0.7050	0.7236	0.7802	**0.7943**	∗,⋄,∼
E024	0.0187	0.0223	**0.0394**	0.0367	∗
E025	**0.0219**	0.0245	0.0161	0.0135	⋄
E026	0.0731	0.0745	0.0976	**0.1109**	∗,⋄,∼
E027	0.1152	0.0133	0.1254	**0.1812**	∗,⋄,∼
E028	0.1863	0.2214	**0.2700**	0.2278	∗,⋄,∼
E029	0.2046	0.1987	**0.2149**	0.1999	∗,⋄,∼
E030	0.1001	0.1276	0.1596	**0.1774**	∗,⋄,∼
E031	0.7595	0.7599	0.7422	**0.7697**	∗,⋄,∼
E032	0.0989	0.1011	0.1290	**0.1292**	∗,⋄
E033	0.4571	0.4789	0.5091	**0.5164**	∗
E034	0.3207	0.3214	0.3200	**0.3380**	∗,⋄,∼
E035	**0.3516**	0.3419	0.3252	0.3059	∗,⋄
E036	0.1156	0.1186	0.1064	**0.1288**	∗,⋄,∼
E037	0.1169	0.1257	0.1598	**0.1629**	∗,⋄,∼
E038	**0.0558**	0.0498	0.0557	0.0539	⋄
E039	0.4188	0.4219	**0.4349**	0.4271	∗,⋄,∼
E040	0.0837	0.0889	0.0856	**0.0902**	∗,⋄
MAP	0.2177	0.2187	0.2390	**0.2442**	–

In the previous version (see deliverable D3.1) the VIA supported a 323 concepts list (subset of TRECVID 2014 SIN task [74]). This concept list was complemented in the second year of the project by two additional

[74] http://www-nlpir.nist.gov/projects/tv2014/tv2014.html#sin

Figure 26: Sample fragments of lecture video titled "A Mosque in Munich Nazis, the CIA, and the Rise of the Muslim Brotherhood in the West", along with the corresponding transcripts and the extracted key terms.

<table>
<tr><th colspan="4">Lecture Video Title : “A Mosque in Munich Nazis, the CIA, and the Rise of the Muslim Brotherhood in the West”</th></tr>
<tr><th>Fragments</th><th>Keyframes</th><th>ASR generated transcripts</th><th>Key terms</th></tr>
<tr><td>Fragment # 01
00:00:03,880 → 00:05:04,730</td><td> </td><td>[...] about his new book mosque
entitled mosque in munich not sees the sea eight and the rise of the muslim brotherhood in the west
thank you thanks and happy to be here
i want to drop picture of any specific time in u s history
when times are fairly grim nuances in a battle against an implacable foe
the prospects are
for success are bleak
unless we can somehow win the hearts and minds of the muslim world
we've tried reaching out but it failed
are president is a man of faith churchgoer who is a non-muslim but has an affinity for islam
he wants to reach out again and is now willing to make allies with a group of muslim thinkers
that's his administration knows to be undemocratic maybe even
bordering on fascist but whom he's willing to court to bring victory [...]</td><td>years
thanks
president
mosque
islam
george w bush
victory
times
video
west</td></tr>
<tr><td>Fragment # 02
00:05:03,760 → 00:06:07,290</td><td> </td><td>[...] for example how or whether we should engage with groups that are antithetical to are ideals
i focus on the history of one mosque
islamic centre of munich
and there are two reasons for this
one is the mask itself is quite an important it was the first western base for the muslim brotherhood
and for many years
was a key centre for this group
the muslim brotherhood you may have heard of may be familiar with but it is these
islamic group if you think of political islam as a tree the muslim brotherhood might
be thought of as the trunk of the tree out of which many other groups
shot out over the years
muscle islamic centre in munich was a place a refuge forum brotherhood leaders
anne d for many years especially
seven easy eighties and nineties and into this decade when its board of directors was
a global who's who of political islam with people from south asia
they and the arab world and europe serving on its board of directors[...]</td><td>muslim brotherhood
today
political islam
munich
years
islamic centre
group
decade
years
trunk</td></tr>
<tr><td>Fragment # 03
00:06:07,290 → 00:16:11,630</td><td> </td><td>[...]analysts to the breakdown of the soviet union
first when the nazis came to power he ally themselves with them very closely
and ended up working fore
ministry called the office ministry and the ministry for the occupied eastern territories
when world war two started
the nazis set a plan to colonize large swaths of the soviet union end in the first year of the war
up to nineteen first year for six months or so
the germans took three million soviet prisoners of war be these nazis first thought that these were all
enemies from end and others like a mini
in the government realized that many of them are potential allies because they were russians
who could fight for the germans so under the leadership of unmanned and others like him
several and these are somebody one the soviet soldiers
were organised into units that fought for the germans in world war two
this is the patch one of these
groups that fought with access units and the reason this is important is that from endo also set up
a propaganda operation using some the better educated soviet minorities
that's a set set up newspapers radio broadcasts in the the soviet union all trying
to convince people not to fight for the soviet union ant to lay down their[...]</td><td>germany
the soviet union
war
world
the mosque
the germans
munich
hand
units</td></tr>
</table>

ones and their respective pre-trained DCNN models: a list of 365 place/location related concepts [75] and a list of 5,000 Imagenet [76] concepts. The alternative concept lists can be selected by altering a paramenter in the REST call. Further to this, the service's stability and reliability were improved by continuous testing during the second year, which revealed different possibilities for small improvement in robustness and time efficiency, and led to various bug fixes and upgrades.

3.9 SciFis: a scientific search engine

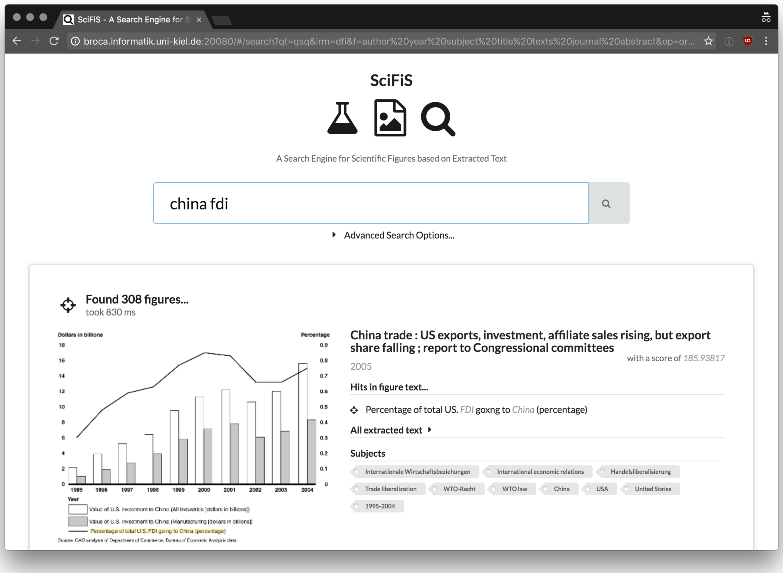
3.9.1 Problem statement

Nowadays, most search engines (general and specific) only take the textual content or the metadata of documents (e.g. scientific publications or webpages) into account. Content in non-ASCII format (e.g. images) is usually ignored. However, scholarly figures, as an example for such content, contain valuable information since they are used to present core research results in scientific publications. Extracting the textual content of these scholarly figures is one option to make these images accessible to a search engine. We have reported our results on text extraction from scholarly figures using our TX pipeline in D3.1 Section 3.3. This text extraction allows a search engine to either search directly for images or enhance the search of the surrounding documents. However, when searching for the text from scholarly figures, one has to address different challenges, e.g. sparse and short text, abbreviations, or recognition errors.

[75] http://places2.csail.mit.edu/
[76] http://www.image-net.org/

3.9.2 Implementation

We have developed a prototypical search engine called Scientific Figure Search (SciFiS)[77] to address the aforementioned challenges. SciFiS is working on a subset of about 7,800 scholarly figures which were semi-randomly extracted (i.e. taking size restrictions into account) from the millions of figures contained in the scientific publications of our Open Access Economics dataset collected from EconBiz[78], the ZBW publication portal for economics and business studies. The extracted figures were processed by the TX pipeline which extracts the text from the images. In the background, our SciFiS search engine relies on Elasticsearch[79] for storing and querying the textual content of the images and the metadata of the corresponding papers. Figure 27 shows the user interface.


SciFiS - A Search Engine for S
broca.informatik.uni-kiel.de:20080/#/search?qt=qsq&irm=dfi&f=author%20year%20subject%20title%20texts%20journal%20abstract&op=or...
Advanced Search Options...
Dollars in billions
Percentage
18
16
14
12
10
8
6
4
2
0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
Year
Value of U.S. investment to China (All Industries [dollars in billions])
Value of U.S. investment to China (Manufacturing [dollars in billions])
Percentage of total U.S. FDI going to China (percentage)
Source: GAO analysis of Department of Commerce, Bureau of Economic Analysis data.

SciFiS

A Search Engine for Scientific Figures based on Extracted Text

china fdi

Found 308 figures...
took 830 ms

China trade : US exports, investment, affiliate sales rising, but export share falling ; report to Congressional committees
with a score of *185.93817*
2005

Hits in figure text...

Percentage of total US. *FDI* goxng to *China* (percentage)

All extracted text

Subjects

Internationale Wirtschaftsbeziehungen · International economic relations · Handelsliberalisierung · Trade liberalization · WTO-Recht · WTO law · China · USA · United States · 1995-2004

Figure 27: User interface of SciFiS.

Our data model holds different metadata of the surrounding publication of a figure (author, year, keywords, title, journal/conference, and abstract) as well as the information about the figure itself (e.g. text extracted from the figure) which can be queried separately or in conjunction. Additionally, we can vary the information retrieval (IR) model that is used to retrieve the documents. The IR models available are TF-IDF (Salton, 1989), BM25 (Robertson & Walker, 1994), Divergence from Randomness (DFR) (Amati & Van Rijsbergen, 2002), Divergence from Independence (DFI) (Din, 2012), and Dirichlet Language Models (DLM) (Zhai & Lafferty, 2001). To account for recognition errors and abbreviations, the implemented retrieval method is based on trigrams and an editing distance. The Elasticsearch backend of SciFiS is wrapped with a custom REST API which is called by the SciFiS frontend. The SciFiS web-frontend looks like a common search engine, but allows to search in the textual content of scholarly figures. Its advanced options allow selecting the data fields to which the query is matched as well as the selection of the IR model. Additionally, a custom query language is supported which allows to specify in detail how tolerant/fuzzy the retrieval should be. The query language also offers the option to boost specific terms. The result list consists of scholarly figures from the scientific publications together with information about the corresponding publication and a link to the publication on EconBiz. Furthermore, the text inside a figure that matches the query is highlighted.

3.9.3 Experimental evaluation and comparison

We conducted a comparative evaluation of different configurations. Due to missing evaluation datasets, we created an artificial gold standard from 121 manually annotated scholarly figures. We created a set of queries from the STW[80] thesaurus, by sampling terms from the thesauri which appeared in the gold standard and use the order of results of the figures based on the manually annotated text as baseline. Given this baseline, we

[77] `https://www.econbiz.de/eb/en/beta/scifis-figure-search/`, lastly accessed on 01/02/2018
[78] `https://www.econbiz.de/`, lastly accessed on 01/02/2018
[79] `https://www.elastic.co/products/elasticsearch`, lastly accessed on 01/02/2018
[80] `http://zbw.eu/stw/version/latest/about.en.html`, lastly accessed on 01/02/2018

computed the nDCG (Järvelin & Kekäläinen, 2002) of the result list based on the extracted text. We compared 8 different configurations which differ with respect to the text analyzer (standard vs custom trigram), the search fields (only text or text with metadata) and the data (manually annotated vs extracted text). Table 31 shows the different configurations. We executed these configurations with five different IR models (TF-IDF, BM25, DFR, DFI, and DLM).

Table 31: Overview of the different configurations of SciFiS using either gold standard (GS) or extracted text (TX) and using either the simple analyzer (SA) or custom analyzer (CA) on the figure text (T) or figure text with metadata (TM)

SA vs CA	**GS**	**TX**	**GS vs TX**	**SA**	**CA**
T	I.	III.	**T**	V.	VII.
TM	II.	IV.	**TM**	VI.	VIII.

The results for the comparison of the analyzers are presented in Table 32 and the results for the comparison of the text quality is shown in Table 33. The results show a drop from 20 % to 80 % given the different configurations when comparing those on the manually annotated texts and the extracted texts. They also demonstrate that the addition of metadata information improves the performance. However, no clear favorite for the IR model could be identified. The largest impact on the performance of the SciFiS search engine has the quality of the extracted text from the figures. Thus, although the results are generally promising, the system is not ripe enough to be integrated into the MOVING platform.

Table 32: Average nDCG between Custom Analyzer and Simple Analyzer.

IR models	**Configurations**			
	V.	VI.	VII.	VIII.
BM25	0.807	0.878	0.205	0.628
TF-IDF	0.807	0.879	0.205	0.628
DFR	0.805	0.878	0.205	0.627
DFI	0.808	0.880	0.206	0.629
DLM	0.807	0.878	0.205	0.628

Table 33: Average nDCG comparing gold standard with extracted text.

IR models	**Configurations**			
	V.	VI.	VII.	VIII.
BM25	0.203	0.484	0.172	0.509
TF-IDF	0.203	0.484	0.171	0.511
DFR	0.197	0.474	0.162	0.490
DFI	0.200	0.477	0.172	0.509
DLM	0.199	0.472	0.170	0.504

4 User logging and data analysis dashboard

Both the user logging solution and the data analysis dashboard have already been deployed in the MOVING platform, as described in deliverables D3.1 (Blume et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017). However, continuous maintenance is necessary to ensure they fulfill their role. The user logging solution need to support the needs of any new module and further improvements that require sophisticated interaction information, while ensuring the captured interaction is reliable and remains unaffected by any changes made to the MOVING platform. The data analysis dashboard needs to constantly evolve, improve the offered analyses possibilities, and meet consortium partners' expectations. In this section we explain the extension to the captured data and the documentation of the platform interactions on the user logging (Section 4.1), as well as the longitudinal evaluation and the new assisted pattern mining of WevQuery (Section 4.2).

4.1 Logging of user interaction data

4.1.1 Problem statement

The UCIVIT[81] (Apaolaza, Harper, & Jay, 2013) interaction capture solution is currently capturing Web interaction data from the MOVING platform, allowing the ATS module to keep track of users' interaction.

However, this user logging solution needs to continuously support the needs of not only the ATS module, but any other MOVING module that may make use of the captured data. The development of other MOVING features, such as the recommender system based on user interaction (as described in the upcoming deliverable D2.2: *Updated curricula and prototypes for adaptive training support and introductory MOVING MOOC for community building*), requires additional information about the captured interaction.

For instance, in order to cluster similar users and provide effective recommendations, a way to easily retrieve all the documents selected from the search was found to be necessary. The number of times a user selects a result, as well as the nature of this result (including what database resource is linked to) is necessary.

The reliability of the captured interaction data also needs to be ensured over time. As MOVING is an ever-changing platform susceptible to modifications by various consortium partners, the necessary code annotations in the Web pages providing meaning to the captured data can be accidentally altered. Maintaining a periodic review of the key interface elements helps to ensure users' interaction with the platform is correctly captured.

4.1.2 Extension to the captured data

The interaction data by UCIVIT provides generic Web interaction data lacking information that would allow researchers to link the interaction events with the content shown in the screen. Extending the gathered information about the events has been considered necessary in order to create user profiles according to the content they consume, as well as to match these profiles to be able to serve recommendations. UCIVIT has been modified to capture more details about the interface elements, such as the HTML class, and the IDs from the parent interface elements. This way, MOVING modules can retrieve not only interaction with particular interface elements, but also with certain categories of elements. For example, a prospective recommender system would be particularly interested in retrieving all mouse clicks on the items returned from a search (HTML elements with class name *result-item*). Furthermore, we are planning to use the ID field from the result elements to stablish a link with the corresponding document in the database. This way, requesting the clicks on result elements would also provide the necessary link to the actual content that would make the recommender system possible.

4.1.3 MOVING platform interaction documentation

Captured interaction data is combining triggered interaction events (e.g. mouse click, key press) and information about the interface elements where the event took place. This way, interaction events taking place with a specific interface element can be retrieved (e.g. mouse click on a *submit* button). Therefore, the information provided by the captured interaction events is highly dependent on interface annotations. As the project is in the state of development, accidental changes to annotations can happen, compromising the reliability of the captured events.

To prevent such issues, a document listing all interactive elements in the platform has been designed. Periodically all elements are checked, and all additional interaction events are recorded. For example, as new visualisations are added, interaction events specific to these modules can be recorded. Table 34 shows the

[81] `https://github.com/aapaolaza/UCIVIT-WebIntCap`, last accessed: 20/03/2018

current state of the recorded interface elements. For each recorded interface element the following information is recorded:

Source Indicating the module where the element is located. Depending on the Web page to be shown to the user, the MOVING platform may combine multiple modules. It can be a common page element which is loaded in every page in the platform, or it can be specific to a Web page, such as the Advanced Search page. However, the page shown to the user can be a merge of several pages. For example, after carrying out a search the user is shown the "Search Results" page merged with the "Search" page containing the keywords used.

Description Short description to help locating the element within the Web page shown to the user.

Location The actual location of the element in the Web page shown to the user.

Type The HTML type of the element. It can be used to retrieve interaction events from the WevQuery REST service.

ID Unique identifier assigned to the element. This is the common key to be used when retrieving events.

Class As opposed to the *ID*, the class is not unique to a single element. Interaction with groups of interface elements can be retrieved this way. For example, retrieving all mouse clicks on HTML elements with class name *result-item* would return all the clicks on documents shown to the user as a result of a search.

Last tested Indicates the last date in which the interaction with this element was tested. It encourages a periodical test for this element. This cannot be automated as testing the interaction with an element is not trivial. There can be external elements (such as other interface elements obscuring the element) that can prevent the normal behaviour of the capture.

Notes Known issues for a particular element, or notes useful when retrieving the events.

This document is available to all consortium members, and will be kept up to date throughout the project. Additional interface elements from other MOVING Web pages as well as interaction with particular MOVING modules (such as visualisations) will be included in the document.

Table 34: Table listing all interface elements annotated for interaction capture.

Source	Description	Location	Type	ID	Class	Last tested	Notes
Common page elements	Navigation toggle (Mobile)	Top	button	navbar_toggle_button		07/12/2017	
	MOVING Logo	Top	link	home_link		07/12/2017	
	Search	Top	link	search_link		07/12/2017	
	Projects	Top	link	projects_link		07/12/2017	
	Learning Environment	Top	link	learning_link		07/12/2017	
	Contacts	Top	link	community_link		07/12/2017	
	My account	Top	link	account_link		07/12/2017	
	Sign in	Top	link	signin_link		NA	Tracking is not active before login
	Contact	Bottom	link	contact_link		07/12/2017	
	Terms of Service	Bottom	link	terms_link		07/12/2017	
	Privacy Policy	Bottom	link	privacy_link		07/12/2017	
	Imprint	Bottom	link	impressum_link		07/12/2017	
	sidebar-toggle-left	Bottom	link	toggle_leftside_link		07/12/2017	
	sidebar-toggle-right	Bottom	link	toggle_rightside_link		07/12/2017	
Login	Register	Middle	link	register_link			
	Login	Middle	button	login_button			
	Lost password	Middle	button	lost_password_button			
	Email	Middle	input	username			
	Password	Middle	input	password			
Home	Links to latest projects	Middle	link		link_to_project		
	Latest news	Middle	link		link_to_news		
Search	Simple Search	Middle	link	search_simple_link		10/01/2018	
	Advanced Search	Middle	link	search_advanced_link		10/01/2018	
	Search the web	Middle	link	search_web_link		NA	Button was removed
	Container for search items	Middle	form	search_form		10/01/2018	Interaction with search_domain dropdown items get registered here, registering the selected value.
	Research/Learning/Funding dropdown group button	Middle	button	search_domain_button		10/01/2018	Doesn't trigger 'change' events like regular dropdowns, see 'search_form' to retrieve selected value.
	Search button	Middle	button	search-button		10/01/2018	
	Search text input	Middle	input	q		10/01/2018	
Advanced Search	Simple Search	Middle	link	search_simple_link		10/01/2018	
	Advanced Search	Middle	link	search_advanced_link		10/01/2018	
	Search the web	Middle	link	search_web_link		NA	Button was removed
	Search button	Middle	button	search-button		10/01/2018	
	Research/Learning/Funding dropdown select	Middle	input	search_domain			Triggers 'change' event like regular dropdowns, keeping track of the new value.
	Title	Middle	input	advanced_query_title		10/01/2018	Keypress events keep track of the text content up to that point.
	Abstract	Middle	input	advanced_query_abstract		10/01/2018	Keypress events keep track of the text content up to that point.
	Fulltext	Middle	input	advanced_query_fulltext		10/01/2018	Keypress events keep track of the text content up to that point.
	Person	Middle	input	advanced_query_person		10/01/2018	Keypress events keep track of the text content up to that point.
Search Results	Results	Above result list	link	search-tab-results			
	Concept Graph	Above result list	link	search-tab-concept-graph		10/01/2018	
	uRank	Above result list	link	search-tab-urank		10/01/2018	
	Tag cloud	Above result list	link	search-tab-tag-cloud		10/01/2018	
	Top concepts	Above result list	link	search-tab-top-concepts		10/01/2018	
	Top sources	Above result list	link	search-tab-top-sources		10/01/2018	
	Date mentions	Above result list	link	search-tab-date-mentions		10/01/2018	
	Result items	Middle, list of results	link		result-item	10/01/2018	Result elements will be augmented with document ID
	Result items authors	Middle, list of results	link		result-item-author	10/01/2018	Result elements will be augmented with document ID
	Result items concepts	Middle, list of results	link		result-item-concept	10/01/2018	Result elements will be augmented with document ID
	Pagination elements	Below result list	link	pagination		10/01/2018	The text field of the node is stored indicating which pagination item was the target
	Sort by date	Above result list (Right)	link	sort_by_date		10/01/2018	
	Sort by relevance	Above result list (Right)	link	sort_by_relevance		10/01/2018	
	Remove Filters	Left side	button	remove_facet_filters_button		10/01/2018	
	Faceted category collapse	Left side	link	collapse_facet_CATEGORYNAME	collapse-facet	10/01/2018	
	Faceted checkboxes label	Left side	div	faceted_checkbox_CATEGORYNAME	checkbox	10/01/2018	These IDS are duplicated for each checkbox label
	Faceted checkboxes	Left side	input	filters_CATEGORYNAME_	filters-checkbox	10/01/2018	These IDS are duplicated for each checkbox

4.2 WevQuery

4.2.1 Problem statement

The role of the interaction analysis dashboard in the MOVING platform is twofold: support the access and use in production of users' interaction data by other MOVING partners and their modules, and further expanding its interaction data analysis features.

WevQuery[82] (Apaolaza & Vigo, 2017) is currently an integral part of the MOVING platform, providing access to the captured users' interaction data. As a way of ensuring it complies with the current needs, consortium partners are asked to answer questionnaires inquiring how useful the tool was, and the purpose it was used for. Regarding the extension of data analysis capabilities, we have explored the use of pattern mining features to support the extraction of common behaviours from the platform.

4.2.2 Longitudinal evaluation

So far, two users from the MOVING platform have provided feedback about their use of WevQuery. From their feedback, we identified two main uses of the tool so far:

1. Retrieve occurrences of users' carrying out a search and the content of the search keywords. Although already supported, providing the occurrences of the search along with all the search keywords, WevQuery users expressed a desire to be able to include the content of the search as part of the query.

2. Identify specific uses of search functionalities. The combination of mouse interaction and the ID of the specific search functionality provided the desired results.

Both users have reported on the usability of WevQuery through the USE usability questionnaire (Lund, 2001). This questionnaire requires users to express their opinion of thirty statements about various usability aspects of the tool using a 5-point Likert scale, ranging from "Strongly Disagree" to "Strongly Agree". The results for each user are reported in Tables 35 and 36. So far, it has been seen that although users agreed that the tool is useful (mean=3.88 SD=0.81) the ease of use is reportedly lower (mean=3.23 SD=1.07).

Table 35: USE questionnaire results for User 1.

Aspect	**Mean**	**SD**
Usefulness	3.50	0.76
Ease of Use	3.27	0.90
Ease of learning	3.75	0.50
Satisfaction	3.14	0.90
Total	3.37	0.81

Table 36: USE questionnaire results for User 2.

Aspect	**Mean**	**SD**
Usefulness	4.25	0.71
Ease of Use	3.18	1.25
Ease of learning	3.50	0.58
Satisfaction	3.57	0.79
Total	3.60	1.00

This longitudinal evaluation of the tool with MOVING partners is still at an early stage, and we will use the provided feedback to improve WevQuery while keeping track of partners' opinion regarding its usefulness.

4.2.3 Assisted pattern mining

WevQuery currently supports the extraction and analysis of user-defined sequences of events. We extended its capabilities by supporting pattern mining analysis of the extracted queries. As far as pattern mining is concerned, given a set of user interface events, patterns containing sequences or most frequent itemsets (i.e. events taking place together) can be obtained (Mooney & Roddick, 2013). This approach is not without challenges though: pattern mining algorithms generate a large number of resulting patterns that require being filtered to facilitate decision making. This filtering would be particularly useful when high-frequency fine-granularity events (i.e. low-level elements, such as mouse hovers) are used as input to the pattern mining algorithm.

Patterns resulting from the use of events such as *mouse click on a button* will provide less context (and less actionable insights) than *submit search query*. This suggests that the process of filtering out short sequences and itemsets should be selective and requires a good understanding of the context of use. Therefore designers' domain knowledge is of utmost importance and we cannot rely entirely upon set rules if we want to identify results that might convey actionable information. What is more, previous works suggest that "user ratings" on the likelihood of a found pattern to be representative of a frequent user task should be used to evaluate the relevance of the results (Dev & Liu, 2017). Furthermore, the relevance of particular results might not only

[82] https://github.com/aapaolaza/WevQuery, last accessed: 20/03/2018

depend on the output or the system under evaluation, but also on the specific goal of the evaluation. For example, designers who already know the most common tasks but would like to focus on outlying behaviours.

Several works have explored the extraction of event sets from user interface event logs in order to be used for visualisation purposes or further analysis to isolate the regularities exhibited by users (Dev & Liu, 2017; Perer & Wang, 2014; Zgraggen, Drucker, Fisher, & DeLine, 2015). Understandably, designers are provided with these digested preprocessed event sets for the mentioned uses. Access to the original dataset is unusual, and even if this was possible, designers might not have the skills for selecting, preprocessing and transforming the dataset, which prevents agile iterations on the analysis. This agility is desirable when designers reformulate their hypotheses about user interaction as a result of the initial analysis. For instance, a follow-up analysis may require the inclusion of interaction events of higher abstraction levels or additional events that were not initially considered. Consequently, if designers are not actively involved in these workflows, there is the risk of not taking advantage of the insights acquired in earlier analyses. For example, let's assume that a team of designers is exploring the use of the scroll bar while users browse a Web system under analysis. Eventually they realise that the interaction with the browser window should be included in the event set as resizing the browser window might be one of the causes for scrolling. Giving designers direct access to the log data would allow them to refine their exploration using these new insights. Similarly, giving access to interaction log data also helps designers to support or refute their hypotheses about user interaction and iteratively substantiate their choice of the typical tasks on empirical ground. In a way, formulating hypotheses about user interaction filters the noise and generates results that are adapted to the purpose of the hypotheses testing process.

Our approach empowers designers by providing them with the means to select and preprocess their own events sets from raw interaction data. This way we alleviate the dependencies on third parties and make possible agile analyses of interaction data. Consequently, once the data cleansing burden is removed, if appropriate tool support for pattern mining algorithms is provided, it is straightforward for designers to incorporate these algorithms into the analyses workflows. This infrastructure makes feasible data-driven analysis of interaction data that may bring about further hypotheses which could then feed back into the iterative analysis. These hypotheses might be weak or could even be considered *expectations* of the designers. Nevertheless, they serve as starting point to guide the exploration of the data and move iteratively from expectations to consolidated hypotheses, which can be employed in experiments and A/B tests.

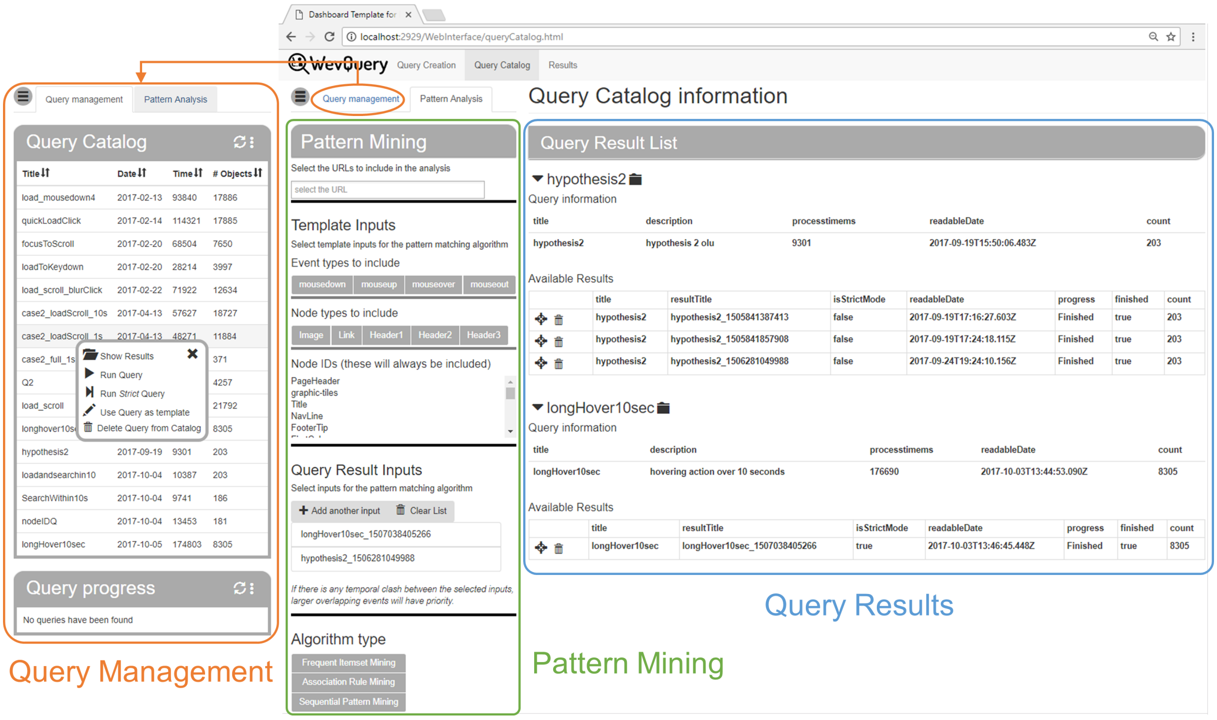
Addressing the Challenges Our approach helps to tackle several problems inherent to the analysis of interaction data and the use of pattern mining algorithms. The fine-granularity of log data provides extensive details about the interaction. Unfortunately, this granularity implies that the number of available events is large, making the selection of events sets from raw data a challenge. Subsetting is pertinent when there are particular events that might not be relevant for the evaluation of the task at hand or fall outside the scope of the interface to be evaluated. For example, if the object of the evaluation is a widget on a website, all interaction beyond the boundaries of said widget would not be of interest. Similarly, designers may want to transform the original events and map them into more suitable, semantically meaningful events. For instance, transforming the mouse clicks on a specific DOM node of a Web page into the higher abstraction event "Add item to cart". Subsetting and transforming the input for pattern mining help to tackle the problem of **noise** of the logs. These preprocessing steps demand expertise in data wrangling. For this reason, as we discussed earlier, designers receive the processed dataset and have no chance to carry out subsetting and transformation activities. We provide individuals with little data wrangling skills with tools for preprocessing their own event sets. This way, they can not only effortlessly retrieve up to date event sets from raw interaction logs, but they can also customise the resulting event set by defining higher abstraction events. The burden for data wrangling expertise is addressed in that designers are provided with a user interface to manipulate raw interaction data and create the mentioned mappings.

Due to the high number of results, handling the output of pattern mining algorithms can be challenging. Discerning useful patterns is non-trivial, and the usefulness of the resulting patterns depends on the nature of the application (Dev & Liu, 2017). Therefore, domain knowledge is necessary to filter the results and determine which patterns are useful. Our approach allows designers to guide the execution of the pattern mining step with their own hypotheses, specifying which sequences should be taken into account. Including the designers in this workflow addresses the domain knowledge problem and helps to filter those patterns that are not relevant for the purpose of the evaluation.

Current approaches lack support for identifying and understanding outlying behaviours. Task-based approaches focus on the most frequent uses of Web applications, and pattern mining techniques favour reoccurring scenarios. Consequently, the results follow a majority rule, where the most common tasks can be identified for further optimisation. However, designers might want to focus on less frequent (but still relevant) activities. Previous research has shown that unexpected interaction patterns may indicate usability problems, and unusual and not envisioned uses of the user interface (Akers, Simpson, Jeffries, & Winograd, 2009). Our approach gives designers the possibility of not only isolating these specific sequences, but also discovering how outlying sequences are intertwined with frequently exhibited activities.

We propose a tool-supported workflow that provides designers with functionalities to select the event sets to be used for pattern mining, allowing them to transform the input by mapping the events into semantically more meaningful ones. This way, designers can reduce the cardinality of events, remove the noise, and focus on the interactions that are relevant to their goals. The discovery of frequent tasks can be cumbersome, and designers have to rely on their domain knowledge to associate the output of the pattern mining with actual tasks (Dev & Liu, 2017). An iterative refinement of the event sets addresses this problem by gradually adding/removing those events that are not significant for the formation of relevant tasks.

Pattern mining The *pattern mining* component in Figure 28 allows users to select the event sets that are going to be used as input of the pattern mining algorithms. There are two ways that facilitate the creation of event sets. On the one hand, users can use the "Query Result Inputs" widget to select as input the results of any of the previously executed queries from the "Query Results" area – these queries may have been created by other users of the platform. On the other hand, the system automatically generates a set of *Template Inputs* under the corresponding header in the "Pattern Mining" area. These template inputs are generated by the system by extracting the occurrences of a set of Web interaction events. The Web interaction events to be extracted are a combination of the user interface event itself (`mouseover`, `mouseout`, `mousedown` and `mouseup` in Figure 28) and the user interface element upon which the event was triggered. The list of all ID attributes assigned to interface elements on the analysed Web page are extracted and considered as targets by default. As long as any event is selected, every interface element with its associated ID triggering such an event will be included. Our rationale was that if the Web page designer considered those interface elements to be relevant enough to assign them an ID, then they should be included in the analyses of the interaction by default. Other common Web interface element types are also included as targets, providing the user a human readable name, and then mapping them to their corresponding HTML tag: `image`, `link`, `Header1`, `Header2`, and `Header3`. For example, if the user selects `mousedown` and `mouseup` as the event types, and `image` as the node type, all interface elements with an ID that trigger such events (these are included by default) as well as any element with the `img` HTML tag (the tag used to place images) will be included in the resulting event set.



Query Catalog

Title	Date	Time	# Objects
load_mousedown4	2017-02-13	93840	17886
quickLoadClick	2017-02-14	114321	17885
focusToScroll	2017-02-20	68504	7650
loadToKeydown	2017-02-20	28214	3997
load_scroll_blurClick	2017-02-22	71922	12634
case2_loadScroll_10s	2017-04-13	57627	18727
case2_loadScroll_1s	2017-04-13	48271	11884
case2_full_1s			371
Q2			4257
load_scroll			21792
longhover10s			8305
hypothesis2	2017-09-19	9301	203
loadandsearchin10	2017-10-04	10387	203
SearchWithin10s	2017-10-04	9741	186
nodeIDQ	2017-10-04	13453	181
longHover10sec	2017-10-05	174803	8305

Query progress

No queries have been found

Pattern Mining

Select the URLs to include in the analysis

select the URL

Template Inputs

Select template inputs for the pattern matching algorithm

Event types to include

Node types to include

Node IDs (these will always be included)

PageHeader
graphic-tiles
Title
NavLine
FooterTip

Query Result Inputs

Select inputs for the pattern matching algorithm

longHover10sec_1507038405266

hypothesis2_1506281049988

If there is any temporal clash between the selected inputs, larger overlapping events will have priority.

Algorithm type

Query Catalog information

Query Result List

hypothesis2

Query information

title	description	processtimems	readableDate	count
hypothesis2	hypothesis 2 olu	9301	2017-09-19T15:50:06.483Z	203

Available Results

	title	resultTitle	isStrictMode	readableDate	progress	finished	count
✥ 🗑	hypothesis2	hypothesis2_1505841387413	false	2017-09-19T17:16:27.603Z	Finished	true	203
✥ 🗑	hypothesis2	hypothesis2_1505841857908	false	2017-09-19T17:24:18.115Z	Finished	true	203
✥ 🗑	hypothesis2	hypothesis2_1506281049988	false	2017-09-24T19:24:10.156Z	Finished	true	203

longHover10sec

Query information

title	description	processtimems	readableDate	count
longHover10sec	hovering action over 10 seconds	176690	2017-10-03T13:44:53.090Z	8305

Available Results

	title	resultTitle	isStrictMode	readableDate	progress	finished	count
✥ 🗑	longHover10sec	longHover10sec_1507038405266	true	2017-10-03T13:46:45.448Z	Finished	true	8305

Figure 28: User interface of the 'Query Catalog': the 'Query management' and 'Pattern Analysis' tabs have been highlighted in this figure.

Once the corresponding inputs (both *Template Inputs* and the *Query Result Inputs*) are selected, users can choose which pattern mining algorithms to run. The parameters for the execution, such as minimum support and minimum confidence thresholds, can be modified, but are given a default value that has been found to be sufficiently appropriate for the nature of the Web interaction log data in use. When users launch the pattern mining algorithms, all the selected inputs are retrieved from the database and put together into an event set that is pipelined into the component running the pattern mining algorithms. Each entry of the event set contains all the occurrences on all the selected inputs for each unique user during a single episode (i.e. a session on the website).

We integrated the SPMF library[83] (Fournier-Viger et al., 2016), an open source data mining library containing 133 algorithms, into our tool-supported workflow in order to incorporate pattern mining algorithms in the workflows for the analysis of interaction logs. SPMF takes a formatted text file as input, and prints the outcome of the selected pattern mining algorithm into another text file. In the evaluation of our approach, we have included the Apriori algorithm (Agrawal, Srikant, & others, 1994) for frequent itemset mining and the PrefixSpan algorithm (Han et al., 2001) for sequential pattern mining. When the execution of the algorithm is completed, a new tab opens up next to the top tabs *Query Creation*, *Query Catalog* and *Results*, where SPMF's output is channelled into the new tab showing the patterns found ranked in descending order according to their frequency.

[83] http://www.philippe-fournier-viger.com/spmf/

5 Visualisation technologies

5.1 Interactive network visualisation

Users that search through the MOVING platform retrieve a list of search results. “Browsing through a long list of documents and then reading parts of the content to locate the needed information can be a mentally exhausting task” (Chau, 2011). Hence, data visualisation concepts can help to find valuable information in search results easier as “the human visual system has enormous power to perceive information from visualised data” (Ware, 2012).

Visualisations included in the MOVING platform include:

1. Graph Visualisation Framework (GVF)[84] for discovery and exploration of relationships.
2. uRank tool for interest-driven exploration of search results. uRank, which was developed previously, has been adapted to and integrated into the MOVING platform.
3. Additional charts and simple visualisations for representing statistical distributions of metadata and topics, such as tag clouds and bar charts of top concepts and keywords.

The visualisations are integrated into the MOVING platform, which allows the user to switch between them and explore data in a variety of ways. This can be seen in Figure 29.

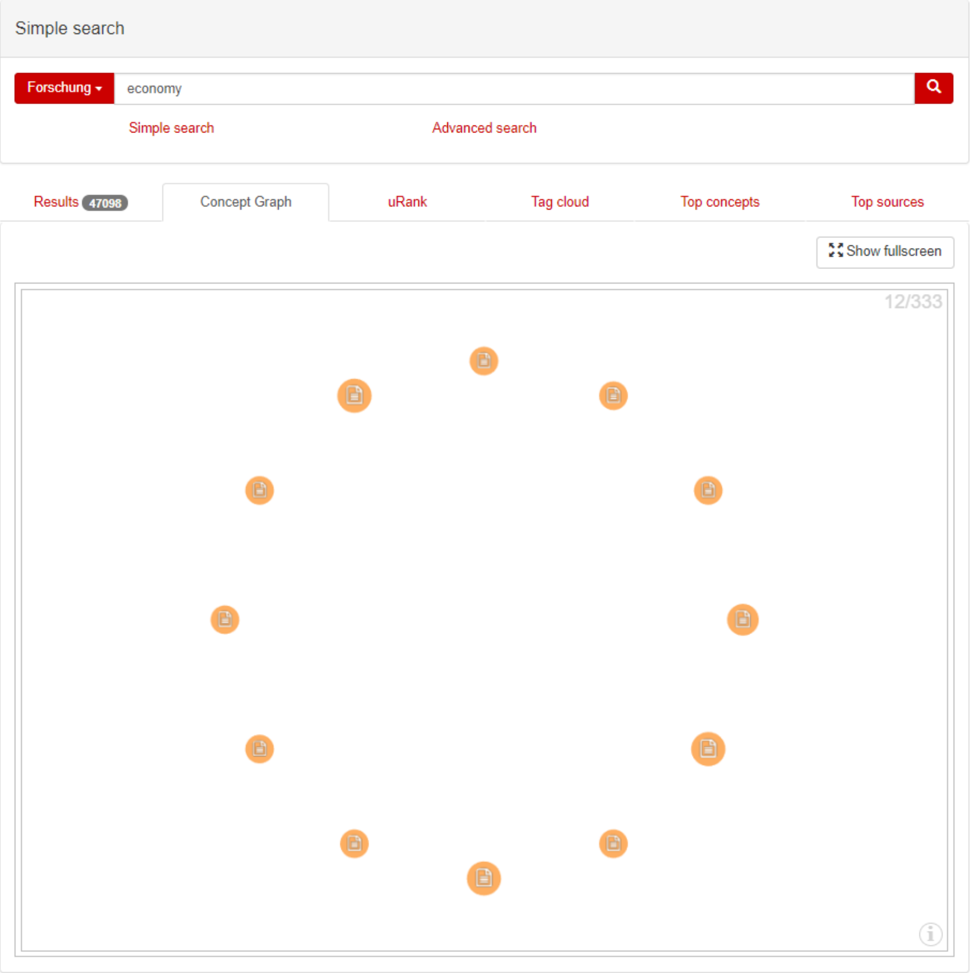


Figure 29: A preview of the initial display of nodes in the concept graph. The visualisations for the retrieved data can be changed by clicking on the respective tabs.

[84] https://github.com/PeterHasitschka/gvf_core, last accessed: 27/03/2017

5.1.1 Problem statement

GVF is a web-based graph visualisation framework designed to support interactive analysis of large, complex networks which may consist of documents, topical concepts, authors, venues, locations and other named entities as well as relationships which arise from co-occurrences, hierarchies, discourses, reading orders etc. To ensure scalability and provide smooth animated transitions, GVF is implemented using WebGL[85]-based rendering.

Graphs or node-link-diagrams are used to represent different entities and relations between them. Entities are visualised as nodes which are connected by links representing relations. Each graph is represented by a specific visual layout, which specifies the positions of the nodes, e.g. force-directed placement algorithms like Fruchterman and Reingold (Fruchterman & Reingold, 1991), and the geometry of the links, such as edge bundling methods (Holten & van Wijk, 2009). Different types of entities and relations as well as metadata can be visualised through different visual variables.

The basic rendering engine of the GVF was developed as part of the AFEL project[86] to represent relationships between large amounts of nodes of a single type (e.g. documents or persons). The rendering engine was developed further in the MOVING project by adding support for nodes and relations of different types resulting in the "Concept Graph" visualisation. For this purpose, the framework had to be specifically tailored to the data model provided by MOVING's search engine. Those adaptations included explicitly defining and implementing the nodes and edges in the graph.

We focus on visual representation of metadata and novel graph aggregation metaphors (Kienreich, Wozelka, Sabol, & Seifert, 2012) conveying relevant properties of nodes and relations in sub-graphs. For this purpose, a new navigational element was added, the ring-menu, supporting navigation in sub-graphs of specific depth and node type. Note that visual metaphors for aggregation and navigation are still being developed with the original draft designs introduced in deliverable D3.1 (Blume et al., 2017). More powerful interaction models are also under development, which include filtering, visual querying of the data and graph statistics.

5.1.2 User interface

Figure 30 shows the current version of the initial layout of the concept graph. In contrast to the first proposed version in deliverable D3.1 (Blume et al., 2017) the user interface of the concept graph was slightly adapted. Most notably, the sidebars on the left and right side were removed to provide more space for the visualisation, as the graph is integrated into a MOVING platform web-page which limits the amount of available screen space. The content of the right sidebar, which showed additional information about a node, was embedded into the graph as a tool tip when hovering over a node. The left sidebar, which displayed information about the platform, was removed entirely. Additional minor changes include the addition of a graph completion indicator and an information icon. The indicator, placed in the upper right of the graph plane, shows the currently displayed number of nodes in comparison to the total number of available nodes. The information icon (Figure 31), in the lower right, explains the usage of the graph to new users.

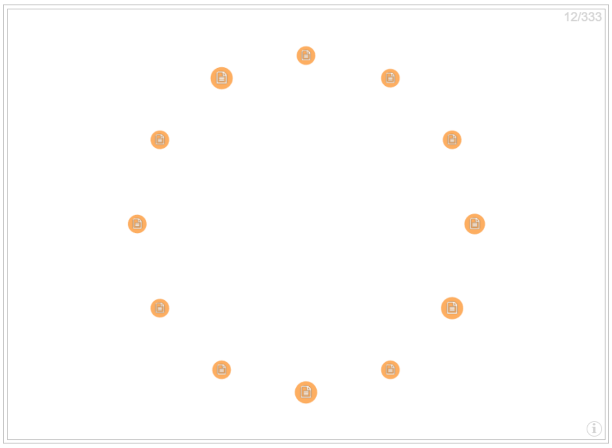
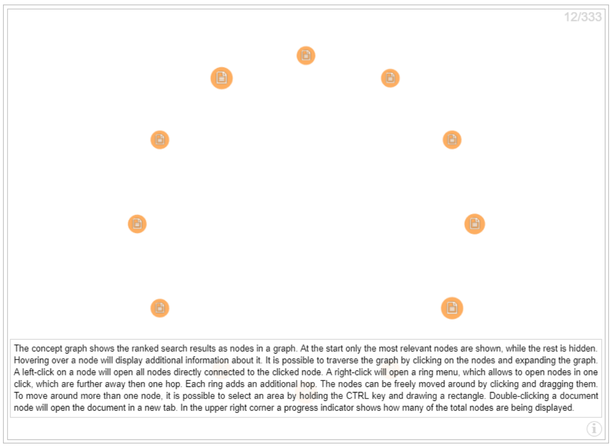


Figure 30: Initial layout of nodes in the Concept Graph. The twelve most relevant nodes are ordered in a circular manner.

[85] `https://www.khronos.org/registry/webgl/specs/latest/`, last accessed: 24/03/2017
[86] `http://afel-project.eu`



The concept graph shows the ranked search results as nodes in a graph. At the start only the most relevant nodes are shown, while the rest is hidden. Hovering over a node will display additional information about it. It is possible to traverse the graph by clicking on the nodes and expanding the graph. A left-click on a node will open all nodes directly connected to the clicked node. A right-click will open a ring menu, which allows to open nodes in one click, which are further away then one hop. Each ring adds an additional hop. The nodes can be freely moved around by clicking and dragging them. To move around more than one node, it is possible to select an area by holding the CTRL key and drawing a rectangle. Double-clicking a document node will open the document in a new tab. In the upper right corner a progress indicator shows how many of the total nodes are being displayed.

Figure 31: Clicking on the information icon on the bottom right reveals a tooltip which explains how the graph can be used.

After entering a search query, the twelve most relevant results are displayed in a circular manner in the graph plane. The order of the nodes in the clock-wise direction corresponds to the relevance provided by the search engine. The graph can be explored by expanding the nodes: clicking on a node will display it's neighborhood. Compact information about a node can be viewed by hovering over a node, which will display a tool tip. The resource represented by the node can be opened by double-clicking on it. Currently the graph differs between four types of nodes, namely "document", "author", "year" and "affiliation". Each node has a different color and icon, as illustrated in Figure 32

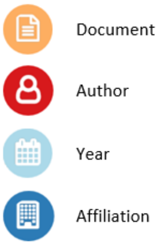


Figure 32: Each node has its own label and coloring. Documents are orange, authors are red, years are light blue and affiliations are dark blue.

In addition to zooming and panning, the option to select nodes and move them around the visualisation plane was added to support exploration. This feature allows the user to inspect a smaller section of the graph in detail, or to move nodes away from a cluttered area. The user can either move a single node, or a group of nodes. A single node can be moved directly by simply clicking on a node and moving it around while holding the left mouse button pressed (drag and drop), while a group of nodes has to be first selected. This can be done by holding the `CTRL` key pressed, while selecting an area with the mouse (rectangular selection). Figure 33 shows how a group of nodes can be moved.

To simplify the expansion of the nodes, a ring-menu was added which allows the user to expand a whole sub-graph with a single click. The ring-menu, shown in Figure 34, can be accessed by right-clicking on a node. Different rings allow the user to display nodes that are associated with the original node through other intermediary nodes. The further the ring is positioned from the original node, the more intermediary nodes will be expanded.

Note that in a previous version of GVF, each ring was subdivided into segments representing node types, as seen in Figure 35. In this representation each ring was effectively a doughnut chart enabling the user to understand the distribution of node types in the neighborhood. Also, clicking on a segment would restrict expansion to include only nodes of the corresponding type. However, it appears that such a representation is too complex for most users. For this reason, we have now disabled the doughnut chart rings and will explore simpler possibilities for type-restricted expansion.

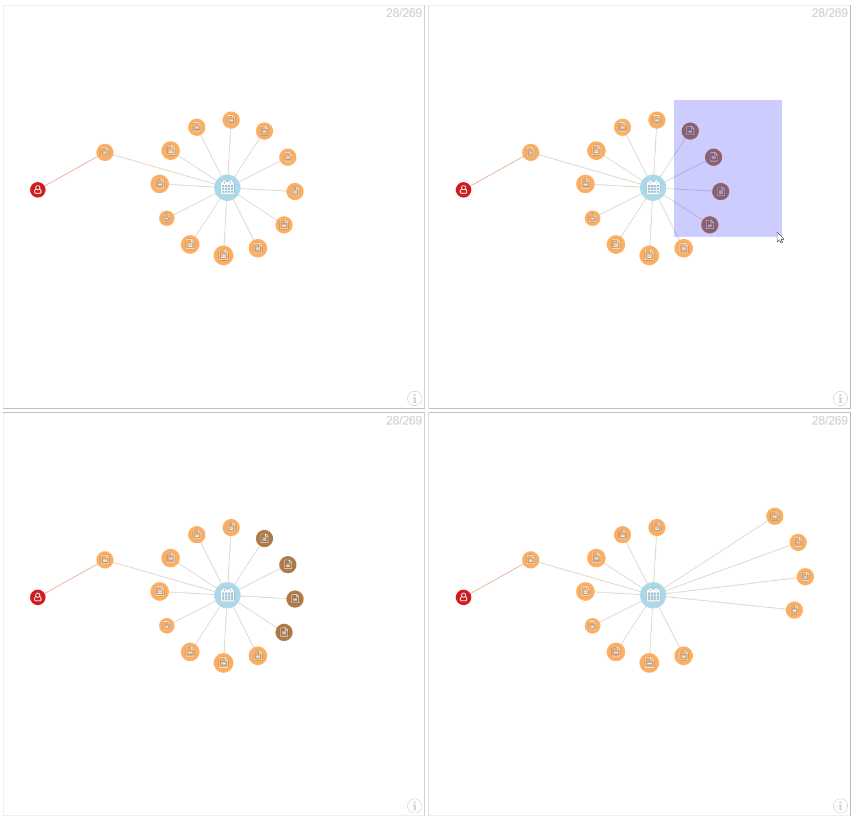


Figure 33: Moving a group of nodes in the concept graph. In the upper left picture a sub-graph is displayed. From this sub-graph some nodes can be selected (upper right). The lower left picture shows the marked nodes and the lower right the result of the movement.

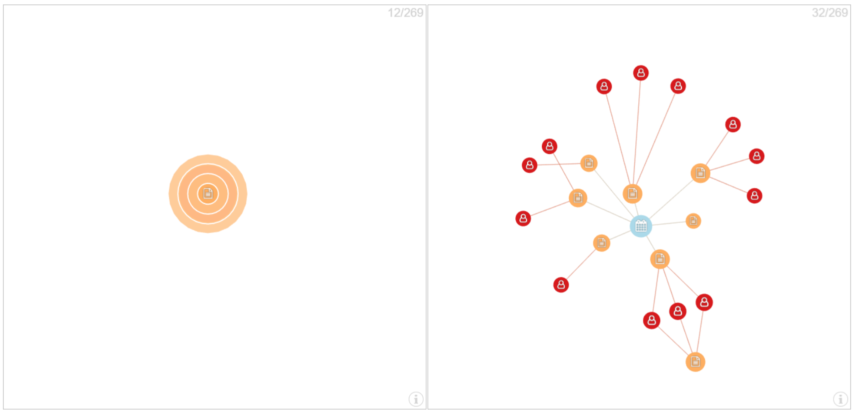


Figure 34: In the ring-menu (left), the outer most ring was clicked to generate a sub-graph (right). The farther away a ring is from the center, the more intermediate nodes will be used to generate a sub-graph. Every ring adds one additional intermediate node. The original node, on the right is the most central document node (over the light blue year node) and as it can be seen, the farthest node, from the original node has 3 other nodes in between.

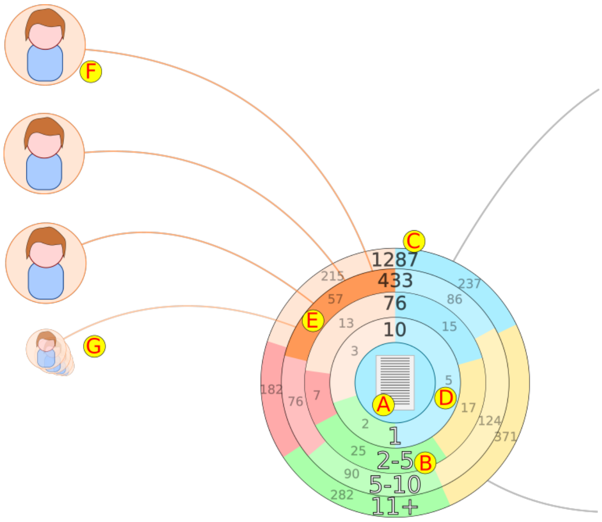


Figure 35: Concept of visualising the aggregated network around the focused node. (A): A document node is the current focus of interest of the user. Only a few other connections are visible (grey edges). (B): Concentric rings surround the node, each representing distance from the focus-node. For example the first ring (1) represents the immediate neighbours, while the second ring (2-5) summarises nodes which have a shortest path to the focus-node between two and five hops. The last ring represents additional, potentially relevant nodes in the graph, which are further away than 10 hops. (C): The number of nodes represented by each ring. (D): Each ring has segments. Each of them represents a different node type. They are colour encoded, thus the user can identify the type. For example the "5" in the first ring indicates that 5 documents (blue) are directly connected to the focus-node. "3" in the same rings indicates that three persons (beige) are mentioned in that document. (E): Interactive elements allow the user to navigate. Hovering over a segment (dark beige in the third ring) shows a handful nodes which correspond to the segment (in the sample: persons which can be reached by following five to ten hops). (F): The example shows that 57 persons meet the distance restrictions. Since showing all of them might overwhelm the user, only the three most relevant are shown. This means, that a ranking depending on parameters like the distance (five might be more interesting than an author that is ten nodes away) or similarities between those nodes and, e.g. the currently focused one. (G): The user can also expand all the other authors which are collapsed in a further meta node.

5.1.3 Use example 1 - exploring the graph click-by-click

To show the benefits of the concept graph, we defined a simple use case. Figure 36 shows the initial placement of nodes in the graph after searching for the term "economy".

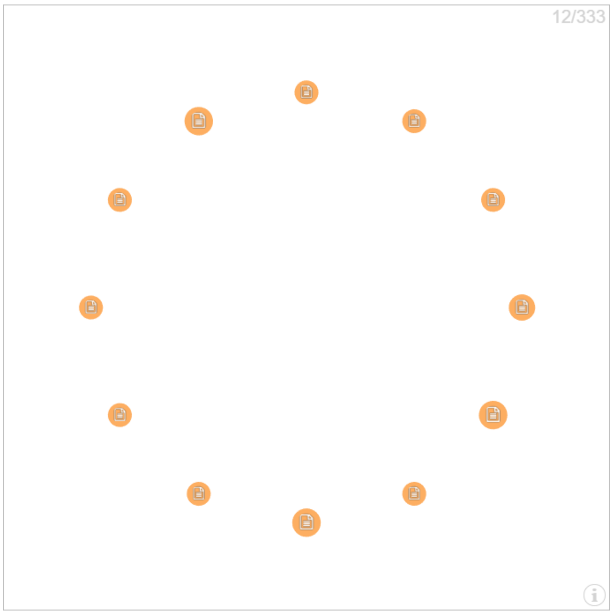
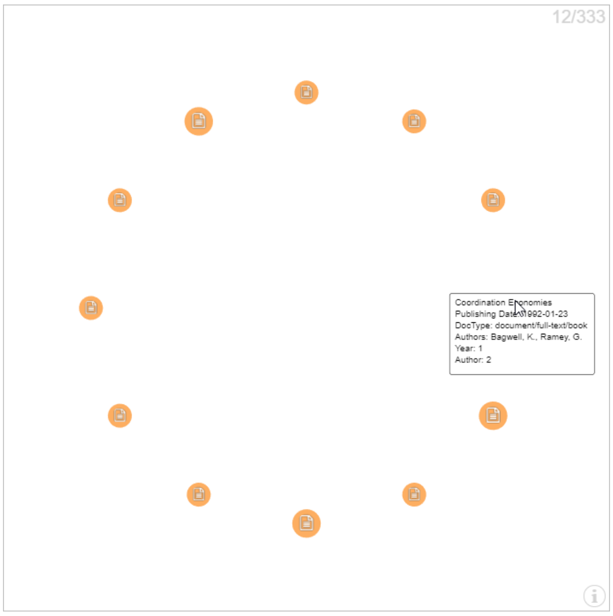


Figure 36: Initial placement of nodes for the search term "economy".

The node with the title "Coordination Economies", which is highlighted in Figure 37, is inspected. Additional information for the node includes the authors Bagwell, K. and Ramey, G., how many nodes are connected to it, the publication year, and the document type.



Coordination Economies
Publishing Date: 1992-01-23
DocType: document/full-text/book
Authors: Bagwell, K., Ramey, G.
Year: 1
Author: 2

Figure 37: Additional information for node of interest show as tool tip.

To be able to better inspect the node of interest, we drag it into an empty area (Figure 38). This also accommodates new nodes which might be added due to expansion. Note that we are currently exploring local space distortion techniques to reserve space for expanded nodes, while preserving most of the graph layout.

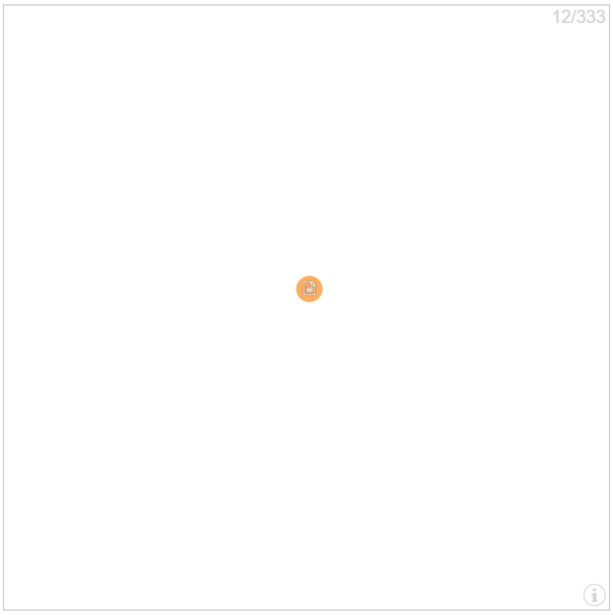


Figure 38: Node moved to empty area.

By clicking on the node, we expands it and show the authors of this resource and the publication year (Figure 39).

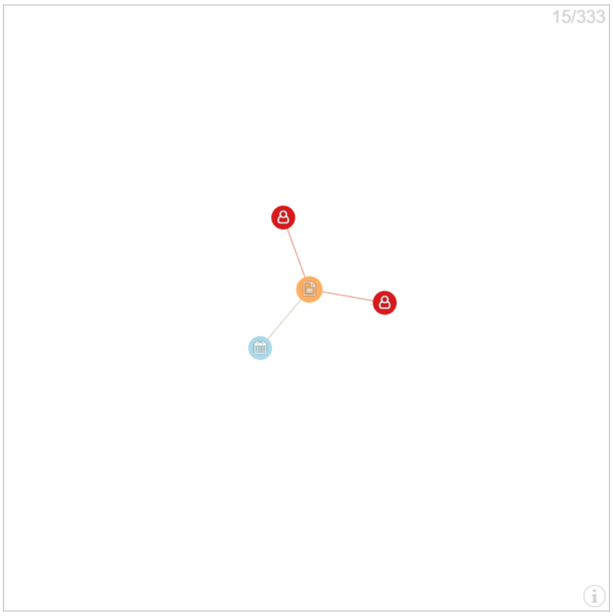


Figure 39: Node expanded revealing authors and publication year.

Hovering over an author node reveals the author name and the number of connected documents (Figure 40). One of those documents is the document from which the author node was expanded.

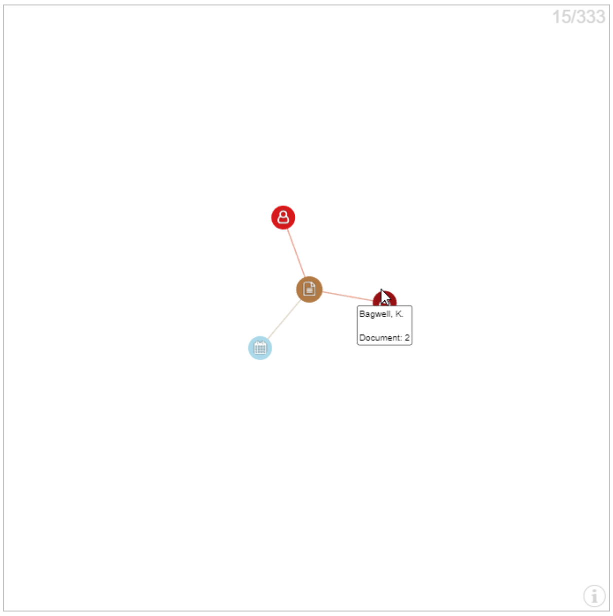


Figure 40: Information about an author node.

Clicking on this author node reveals the other document the author wrote. As it can be seen in Figure 41 this new document is also connected to the other author from the original document node. Therefore, those two authors published two books together.

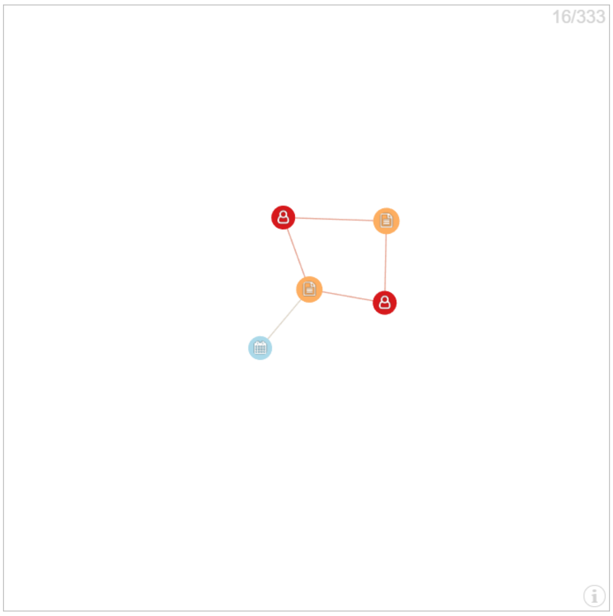
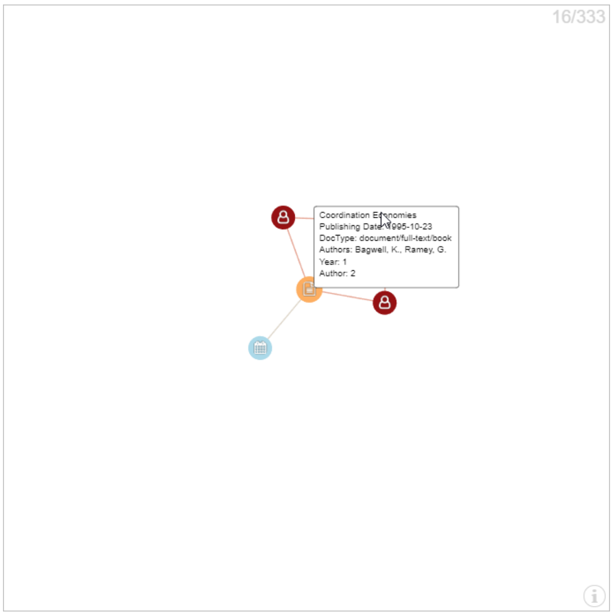


Figure 41: Additional document node reveals that the author published another document with the same co-author.

Hovering over the newly expanded node (Figure 42) reveals that it has the same title as the original node, yet the publishing date is different. The first one was published in 1992 (Figure 37) and the second 1995 (Figure 42). It is likely that the authors extended the previous work. Thus, the user is better advised to open the resource with the later date.

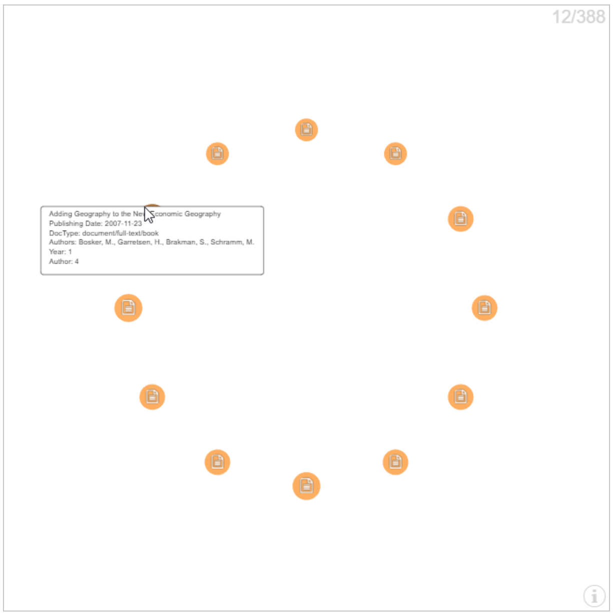


Coordination Economies
Publishing Date: 1995-10-23
DocType: document/full-text/book
Authors: Bagwell, K., Ramey, G.
Year: 1
Author: 2

Figure 42: The new document node has the same title as the original node, yet the publication date is different.

5.1.4 Use example 2 - exploring the graph with the ring-menu

The second use example showcases the benefit of the ring-menu. Again, a search was performed, but this time the search term "geography" was used. Figure 43 shows the initial placement of nodes, where the node with the title "Geography to the New Economic Geography" is highlighted for further inspection.



Adding Geography to the New Economic Geography
Publishing Date: 2007-11-23
DocType: document/full-text/book
Authors: Bosker, M., Garretsen, H., Brakman, S., Schramm, M.
Year: 1
Author: 4

Figure 43: The initial placement of nodes for the search term "geography".

Instead of expanding a sequence of nodes one by one, we can expand a whole sub-graph in just two clicks through the provided ring-menu. First, a right click on the node of interest opens the ring-menu (Figure 44). Each ring in ring-menu represents how far the new sub-graph should expand. Clicking on a node only opens the nodes that are directly connected to the clicked node. Each ring in the ring-menu adds an intermediate node which is used to display additional nodes. Therefore, clicking on the innermost ring not only shows the nodes directly connected to the original node but also displays the nodes connected to the expanded ones. In a similar fashion, clicking on the second or third ring adds one additional intermediate node. In deliverable D3.1 (Blume et al., 2017), a similar ring-menu was proposed, which fetched nodes that were further away. For example, the innermost ring used 1 intermediate node, while the second used 3-5 and the third even more. Due to the provided data, this approach resulted in very sparse graphs, the more intermediate nodes were used. Therefore, in a second approach this was drastically reduced. In future, this will be solved by using meta data to build additional edges in the graph as well as keywords and concepts of the search results provided by the search engine. Initially, more rings were used, but due to user feedback this was also reduced to three.

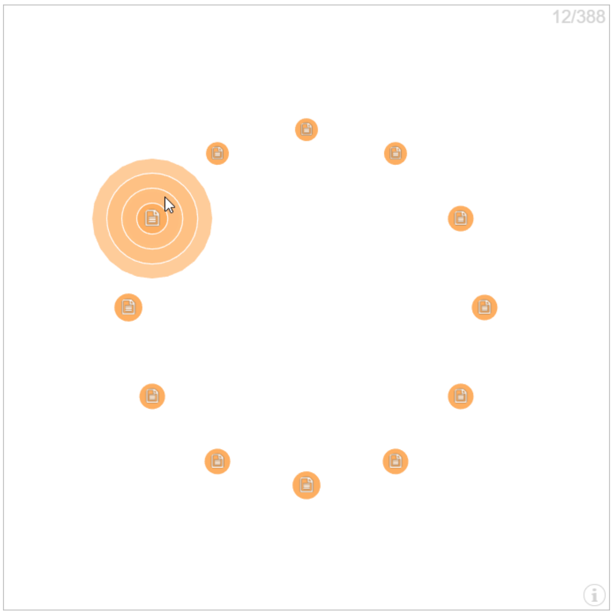


Figure 44: The opened ring-menu for the node of interest.

Clicking on a the innermost ring in the ring-menu of a document shows the authors of this document, and all additional documents published by the same authors (Figure 45).

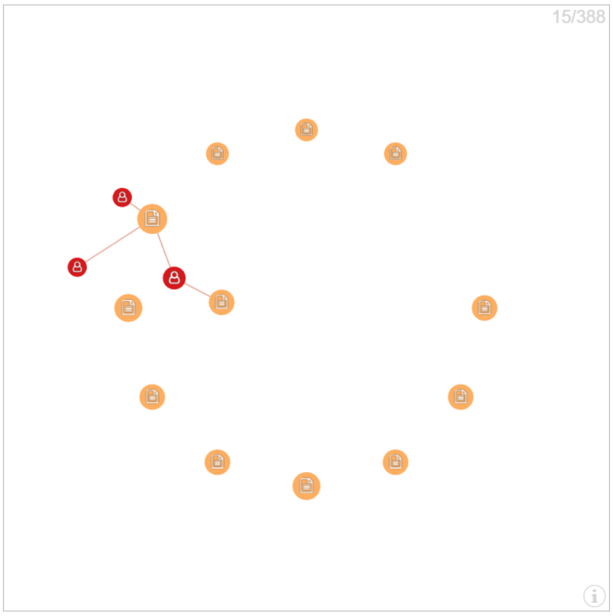


Figure 45: The expanded sub-graph create from using the ring-menu.

In this particular example, the expansion creates a connection to an already existing node. The original node of interest is connected over the author Garretsen, H. (Figure 46) to the node with the title "Geography Rules Too! Economic Development and the Geography of Institutions" (Figure 47). The user can discover that two results from the top 12 were published by the same author. This information might be helpful to the user when further exploring the results or adapting the search query.

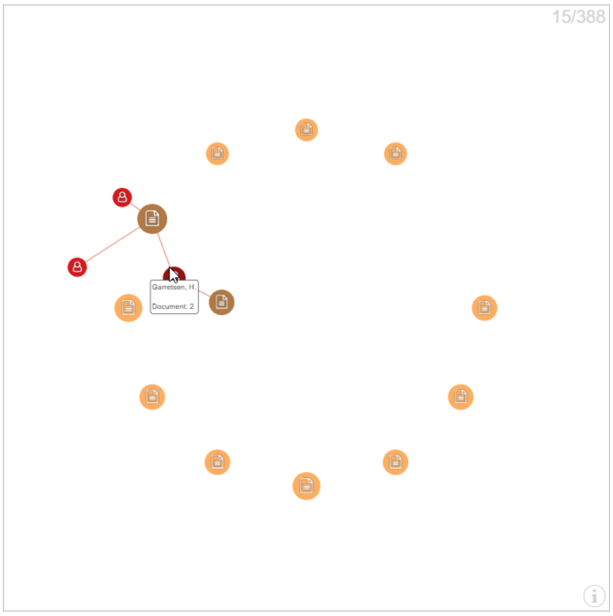
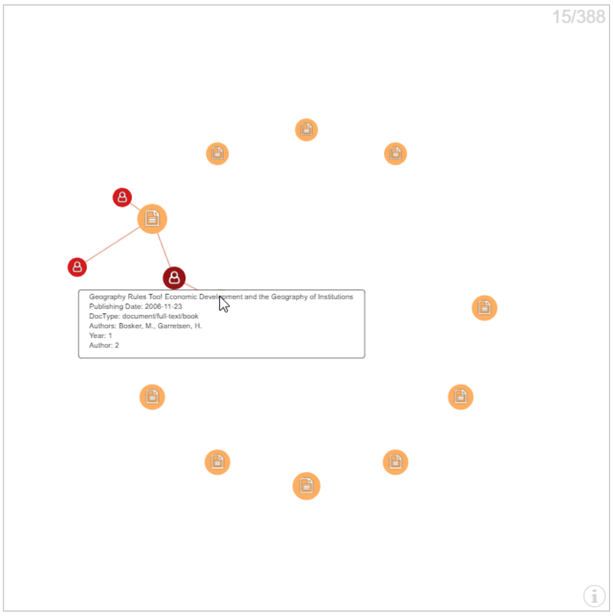


Figure 46: The name of the author which connects two document nodes.



Geography Rules Too! Economic Development and the Geography of Institutions
Publishing Date: 2006-11-23
DocType: document/full-text/book
Authors: Bosker, M., Garretsen, H.
Year: 1
Author: 2

Figure 47: The title of the connected document node.

5.2 uRank: interest-based result set exploration

5.2.1 Problem statement

Searching and browsing are the core activities when users gather and organise new information. As the exploration process unfolds and new knowledge is acquired, interest drifts occur inevitably and need to be accounted for. Despite the advances in retrieval and recommendation algorithms, real-world interfaces have remained largely unchanged: results are delivered in a relevance-ranked list. However, it quickly becomes cumbersome to reorganise resources along new interests, as any new search brings new results.

uRank (di Sciascio, Sabol, & Veas, 2016) is a visual analytics approach that combines lightweight text analytics and a visually augmented ranked list to assist in exploratory search of textual search result sets. The fully web-based tool provides automatic and interactive mechanisms that when combined enable users to explore a document collection and refine information needs in terms of topical keywords. The typical uRank workflow is summarised as follows:

1. uRank receives a set of textual document surrogates, i.e. titles and abstracts, from the search engine.
2. The keyword extraction module analyzes titles and abstracts and returns: (i) a list of weighted representative terms for each document, and (ii) a set of keywords that describe the whole collection.
3. The UI displays a list of documents along with the extracted collection keywords.
4. The users explore the documents and keywords. During this process, they can identify possible key topics or relations between documents and keywords.
5. When the users find interesting terms, they can select them individually or as group via drag and drop.
6. The document list is re-ranked according to the relevance to the selected keywords, and augmented with stacked-bars visualising document scores related to each keyword.
7. The users can select a single document to access more detailed information on it.

User-driven actions (4, 5, and 7) highly depend on the user's search strategy, thus they are rather iterative and interchangeable.

Note that uRank was first developed as part of the EEXCESS[87] project. The concept and the implementation were adapted for and integrated into the MOVING platform. Currently, uRank's keyword extraction performs Natural Language Processing (NLP) on the retrieved results in the web browser. This NLP processing includes the stemming and the removal of stop words for the English language. Optionally, uRank

[87] http://eexcess.eu/

can also rely on server-side keyword extraction. uRank adaptations for MOVING include the changes needed to accommodate the data model provided by MOVING's search engine, initial handling of German language (mainly stopword elimination), changes in the look & feel of uRank to fit the platform, and numerous small adjustments to the UI because of the limited space and user feedback. Additionally, more complex social features, such as organising documents in collections or bookmarking, are available but currently disabled. However, in future they can be enabled if needed.

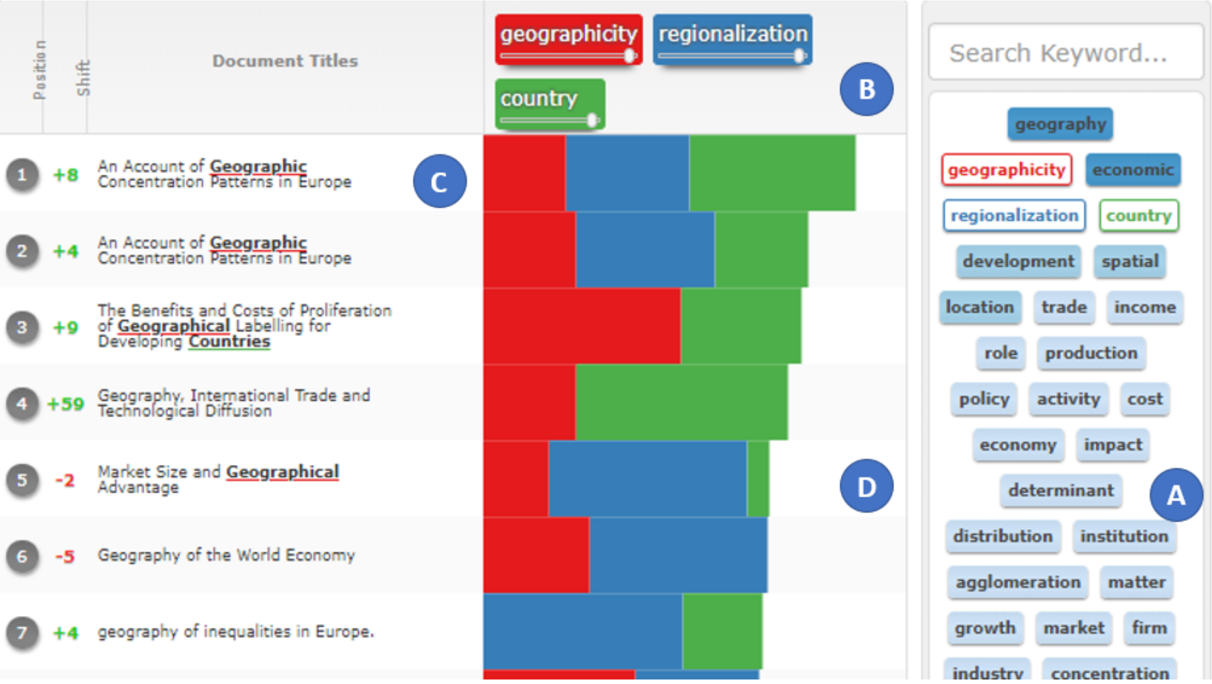
5.2.2 User interface

The UI layout is arranged in a multiview fashion that displays different levels of abstraction of a document collection:

Collection overview The Tag Box (Figure 48.A) summarises the entire collection through keyword tags.

Documents overview The Document List shows titles along with ranking information and the Ranking View displays stacked bar charts depicting document relevance scores (Figure 48.C and D, respectively). The list and ranking visualisation are updated as the user manipulates keyword tags in the Query Box (Figure 48.B).

Document detail view For a document selected in the list, the Document Viewer displays the title and snippet with colour-augmented keywords. The detailed document view can be accessed by clicking on one of the search results.

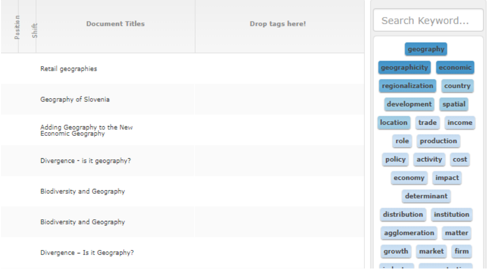


Position	Shift	Document Titles
1	+8	An Account of Geographic Concentration Patterns in Europe
2	+4	An Account of Geographic Concentration Patterns in Europe
3	+9	The Benefits and Costs of Proliferation of Geographical Labelling for Developing Countries
4	+59	Geography, International Trade and Technological Diffusion
5	-2	Market Size and Geographical Advantage
6	-5	Geography of the World Economy
7	+4	geography of inequalities in Europe.

Figure 48: uRank User Interface displaying documents related to "geography", with ranking updated to match the keywords "geographicity", "regionalization" and "country". (A) The Tag Box presents a keyword-based summary of the search results, (B) the Query Box contains keywords selected by the user, (C) the Document List and (D) the Ranking View present a list with document titles augmented with stacked bars indicating relevance scores.

5.2.3 Use example

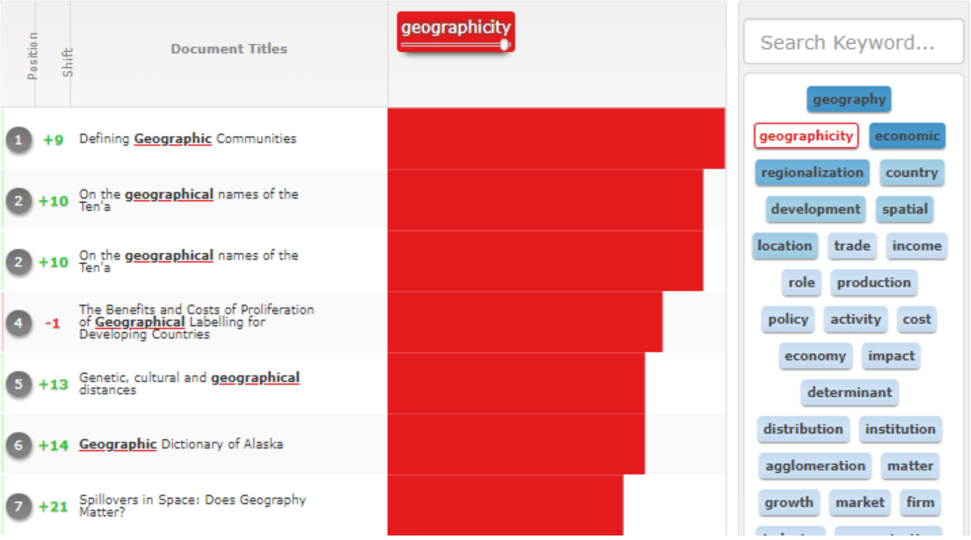
Figure 49 shows the initial ranking given by the search engine after the term "geography" was entered as query. Here, the ranking is based only on the relevance score given by the search engine itself.



Position	Shift	Document Titles	Drop tags here!
		Retail geographies	
		Geography of Slovenia	
		Adding Geography to the New Economic Geography	
		Divergence - is it geography?	
		Biodiversity and Geography	
		Biodiversity and Geography	
		Divergence – Is it Geography?	

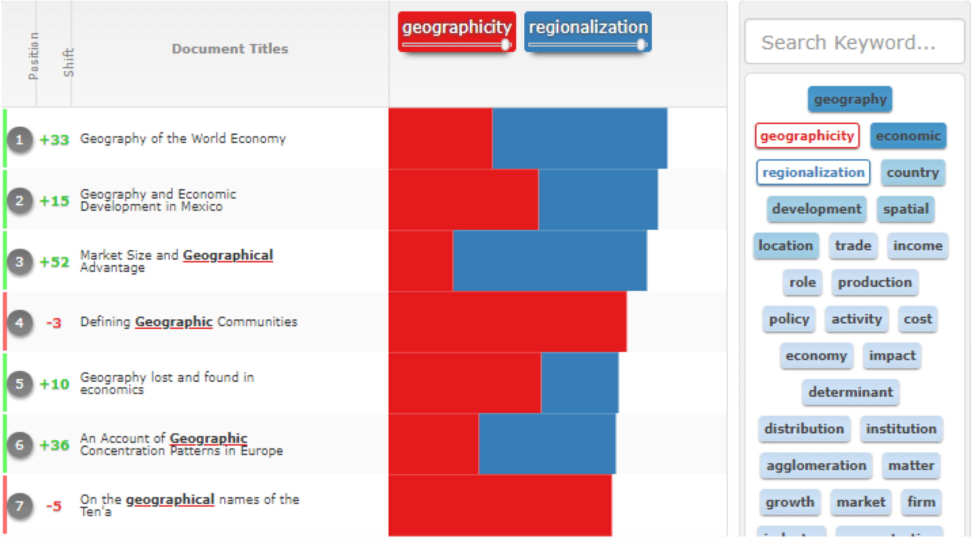
Figure 49: uRank: Initial ranking.

The tag cloud on the right side contains the most relevant keywords extracted from the retrieved documents. Selecting keywords of interest automatically re-ranks the results. Figures 50, 51, and 52 show how much the ranking can change based on the user's interest in certain topics. The shift column on the left side displays by how many positions the rank of the document has changed, depending on the last selected keyword.



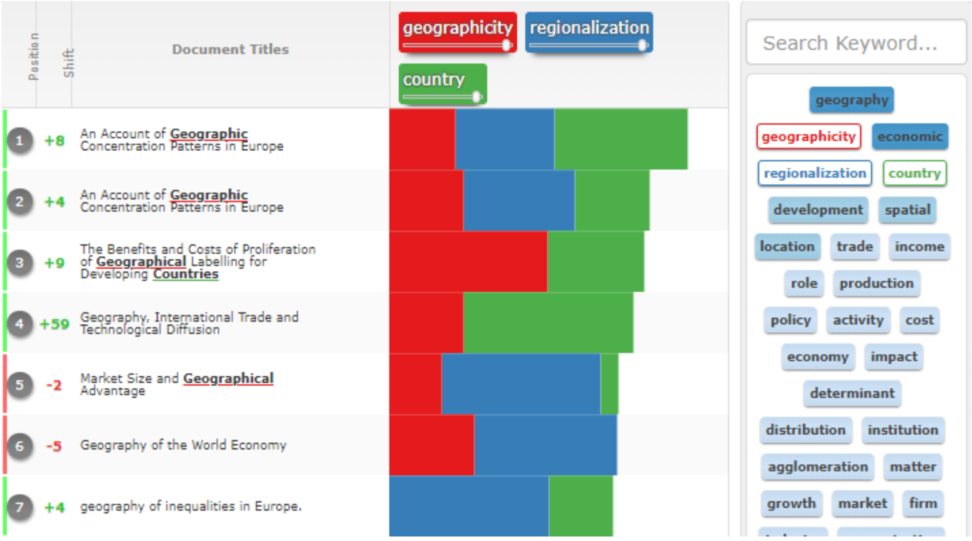
Position	Shift	Document Titles	geographicity
1	+9	Defining Geographic Communities	
2	+10	On the geographical names of the Ten'a	
2	+10	On the geographical names of the Ten'a	
4	-1	The Benefits and Costs of Proliferation of Geographical Labelling for Developing Countries	
5	+13	Genetic, cultural and geographical distances	
6	+14	Geographic Dictionary of Alaska	
7	+21	Spillovers in Space: Does Geography Matter?	

Figure 50: uRank: Results re-ranked after selecting "geographicity".



Position	Shift	Document Titles	geographicity	regionalization
1	+33	Geography of the World Economy		
2	+15	Geography and Economic Development in Mexico		
3	+52	Market Size and Geographical Advantage		
4	-3	Defining Geographic Communities		
5	+10	Geography lost and found in economics		
6	+36	An Account of Geographic Concentration Patterns in Europe		
7	-5	On the geographical names of the Ten'a		

Figure 51: uRank: Results re-ranked after additionally selecting "regionalization".

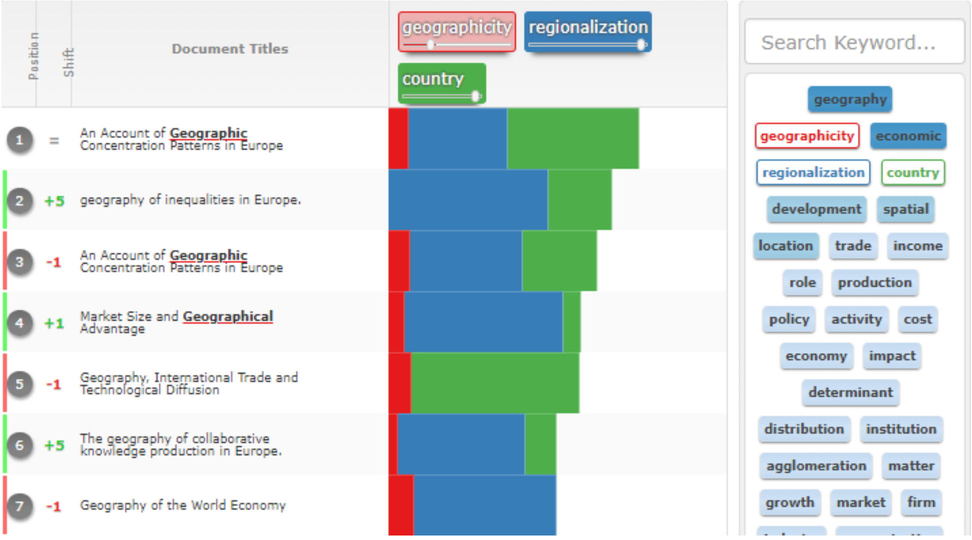


Position	Shift	Document Titles
1	+8	An Account of Geographic Concentration Patterns in Europe
2	+4	An Account of Geographic Concentration Patterns in Europe
3	+9	The Benefits and Costs of Proliferation of Geographical Labelling for Developing Countries
4	+59	Geography, International Trade and Technological Diffusion
5	-2	Market Size and Geographical Advantage
6	-5	Geography of the World Economy
7	+4	geography of inequalities in Europe.

Figure 52: uRank: Results re-ranked after additionally selecting "regionalization".

The ranking can be fine-tuned by adjusting the slider below the selected keywords. This slider reduces or increases the importance of that keyword for the ranking. By default, the slider is positioned in the middle. Moving the slider to the left decreases the keyword's importance, while moving it to the right increases the importance.

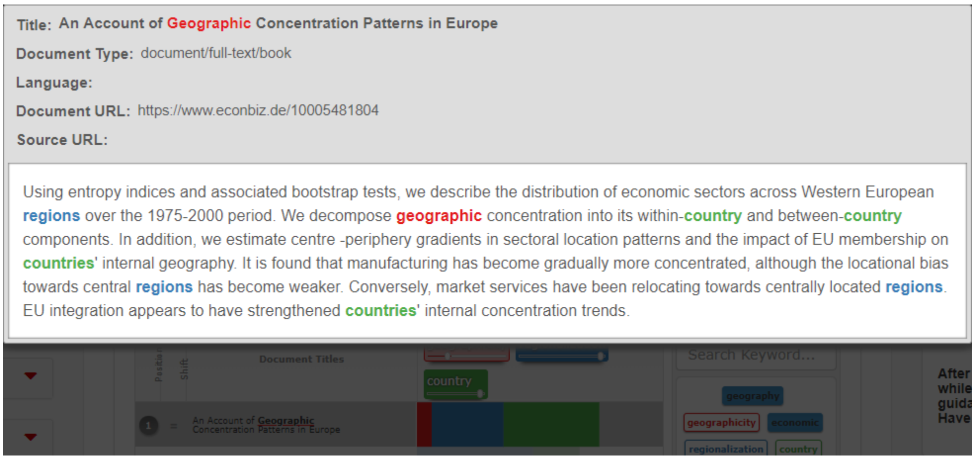
Figure 53 shows the effect of reducing the weight of a keyword. As it can be seen from the stacked bar charts, the keyword still has some influence on the ranking, yet the influence was minimised.



Position	Shift	Document Titles
1	=	An Account of Geographic Concentration Patterns in Europe
2	+5	geography of inequalities in Europe.
3	-1	An Account of Geographic Concentration Patterns in Europe
4	+1	Market Size and Geographical Advantage
5	-1	Geography, International Trade and Technological Diffusion
6	+5	The geography of collaborative knowledge production in Europe.
7	-1	Geography of the World Economy

Figure 53: uRank: Results re-ranked after the weight of "geographicity" was reduced.

Clicking on a result opens the detailed view of the corresponding document (Figure 54), where additional information can be inspected. Selected keywords are highlighted in corresponding colours.



Title: An Account of Geographic Concentration Patterns in Europe

Document Type: document/full-text/book

Language:

Document URL: https://www.econbiz.de/10005481804

Source URL:

Using entropy indices and associated bootstrap tests, we describe the distribution of economic sectors across Western European **regions** over the 1975-2000 period. We decompose **geographic** concentration into its within-**country** and between-**country** components. In addition, we estimate centre -periphery gradients in sectoral location patterns and the impact of EU membership on **countries**' internal geography. It is found that manufacturing has become gradually more concentrated, although the locational bias towards central **regions** has become weaker. Conversely, market services have been relocating towards centrally located **regions**. EU integration appears to have strengthened **countries**' internal concentration trends.

Figure 54: uRank: Document detailed view shows additional information about a document.

5.3 Content and statistics-based result filtering

5.3.1 Problem statement

Having the document results ranked by relevance does not give the user insights about important metadata describing documents. Therefore, simple visualisations, such as a bar chart and a tag cloud were added to help users understand the distribution of content and metadata in the result set, in order to adjust their search parameters accordingly.

5.3.2 Bar chart component

In the current version of the MOVING platform, the bar chart visualisation is used for three different purposes. Figure 55 shows the two bar charts, which are generated upon clicking on the "Top concepts" tab in the search window. The bar chart on the left shows the most frequently occurring concepts in the retrieved results. On the other hand, the bar chart on the right shows the most often occurring keywords in the retrieved results. The keywords summarises the content of the retrieved result, and in this visualisation they are obtained from metadata provided by the search engine. The concepts, also provided by the search engine, are a more structured form of category to which the retrieved results belong to. Therefore, inspecting the highest ranked keywords might help the user refine his search, while inspecting the top concepts gives the user an idea of the area to which most of the retrieved results belong.

Clicking on the "Top sources" tab in the search window generates a bar chart, as it can be seen in Figure 56. Similarly to the "Top concepts" and "Top keywords", this visualisation shows the most frequently occurring sources in the retrieved results. Additional metadata types, such as venues and location, will be added later on.

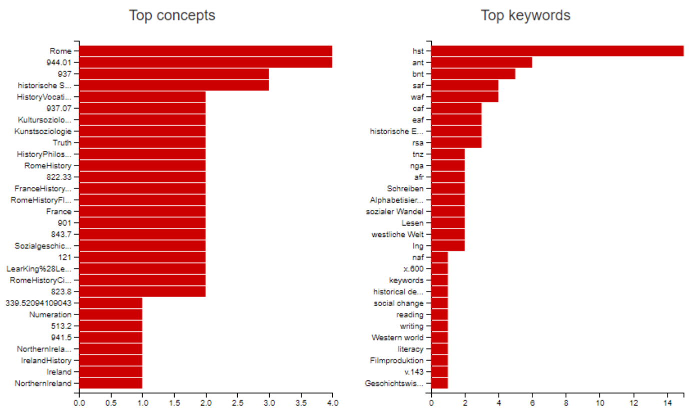


Figure 55: The bar charts display the most often occurring concepts (left) and keywords (right) for the retrieved search results (search query "history").

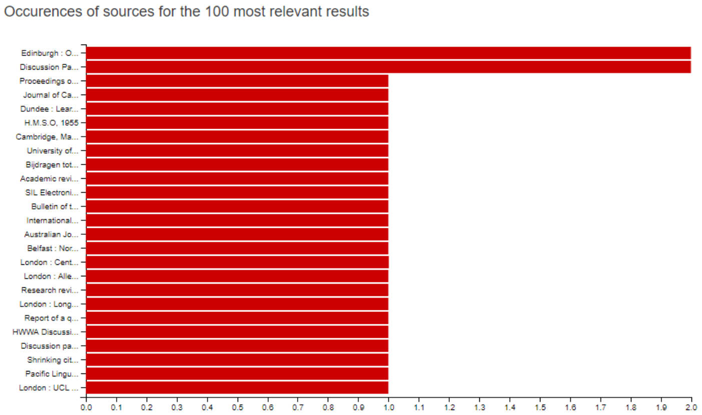


Figure 56: The bar chart shows the most frequently occurring sources in the results when searching for the term "geography".

5.3.3 Tag cloud component

Figure 57 shows the current version of the tag cloud in the MOVING platform. The tag cloud, similar to the bar chart visualising the keywords, displays the most relevant keywords of the retrieved search results. In contrast to the bar chart visualisation, the tag cloud does not rely on the keywords retrieved by the search engine: it relies on the uRank's extraction of keywords from the retrieved content, to generate a richer variety of keywords for the current result set. The size of the tags is based on their relevance (number of occurrences). The tag cloud orders the tags randomly, however, they can be sorted by ascending or descending occurrence. We plan to map properties to colours in a future version. Additionally, tags can be searched and filtered to find out if a specific tag occurs in the cloud.

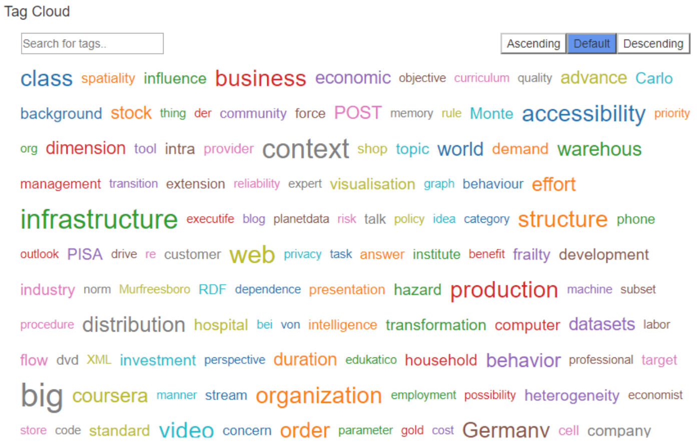

Tag Cloud
Search for tags..
Ascending
Default
Descending
class spatiality influence business economic objective curriculum quality advance Carlo
background stock thing der community force POST memory rule Monte accessibility priority
org dimension tool intra provider context shop topic world demand warehous
management transition extension reliability expert visualisation graph behaviour effort
infrastructure executife blog planetdata risk talk policy idea category structure phone
outlook PISA drive re customer web privacy task answer institute benefit frailty development
industry norm Murfreesboro RDF dependence presentation hazard production machine subset
procedure distribution hospital bei von intelligence transformation computer datasets labor
flow dvd XML investment perspective duration edukatico household behavior professional target
big coursera manner stream organization employment possibility heterogeneity economist
store code standard video concern order parameter gold cost Germany cell company

Figure 57: The tag cloud shows the most prominent keywords in the retrieved search results for the term "data".

5.4 Future work

The technical foundation for all the visualisations has been set up. Future work will include the extension of metaphors and the functionalities, primarily of the graph visualisation and the underlying GVF framework. We plan to add on-the-fly extraction of relations based on co-occurrences in the result set, resulting in additional nodes and edges for users to explore. Metadata filtering is also planned in order to simplify exploration of potentially complex graphs.

We will extend the graph visualisation metaphor in several ways to improve its readability. Edges will convey the relevance and the type of connections through thickness and stroke, respectively. We plan to add edge bundling to minimise edge crossings and a priority-based node and relation labelling to prevent text overdraw. As already mentioned, techniques for local space distortion will be employed to seamlessly reserve additional space when expanding nodes, still preserving most of the graph layout.

The current ring menu will be extended to include additional information on the neighbourhood of a node, such as the distribution of node and edge types depending on distance. This metaphor - probably a simplified version of the previously proposed nested doughnut charts - will also support focused and context-based navigation. For example, the user will be able to expand the graph along relations and nodes of a particular type (focused navigation), with the optional constraint that new, expanded nodes must relate to other already visible, user-selected nodes (context-based navigation).

Finally, the work on visual graph aggregation shall continue. In particular we will implement metaphors for aggregating graph regions which are currently not in the focus of the user's exploration. The resulting visual summary will convey most important information on the corresponding sub-graph, such as its size, density, or node and edge distribution, while significantly reducing the overall complexity of the visualisation.

6 Conclusion

We presented the progress on the set of techniques for data acquisition, data processing, data visualisation and user logging. We provided a new version of the common data model to improve the data processing, augment its expressiveness and better support the user requirements. The new data integration service ensures a minimum quality when integrating new data.

Regarding data acquisition, we improved the crawlers. SEC has been updated to extract and handle embedded videos included in the crawled webpages. FDC has been updated to include "allow" and "block" patterns defined by the user and inherited the SEC's functionality to handle video. SSM now manages not only web pages but also media items. The new FLuID model enables a universal representation of schema level indices and is a basic building block which unifies existing approaches. An extended evaluation and discussion of additional schema level indices modeled with FLuID has been provided, focusing not only on approximation quality but also on storage requirements.

We studied the evolution of vocabulary terms in knowledge graphs. We showed that changes on terms have a large impact on the real data although they are not too frequent. Most of the newly coined terms are adopted in less than one week after their publishing date, although some are only adopted after several months. Surprisingly, there are also terms adopted before their official publishing date. Often, deprecated terms are still in use, i.e. they are not really deprecated in practice.

Regarding data processing, we improved some techniques and we investigated new ones. In the context of author alignment, we focused on author name blocking. We proposed and evaluated a new blocking scheme called *entropy-isolate* that generalises the idea behind existing schemes and allows tuning a threshold parameter. Our evaluation enabled us to estimate to which extent previous work, which was only evaluated within name blocks, has neglected a blocking-related loss of recall. We also presented a technique for duplicate detection, reusing the effective similarity measure deployed in our work on author disambiguation to distinguish duplicate pairs from non-duplicate ones. As future work, we need to determine an optimal set of features that can be reliably applied in MOVING. For the ranking task, we built an effective title-based information retrieval model that provides competitive results compared to a retrieval model operating on full-text. L2R outperformed the other title-based statistical ranking models and the gap between the best full-text-based retrieval model and the best title-based retrieval model is only 6.6%. We also addressed the problem of conducting a semantic multi-label classification on SKOS thesauri by using only the documents' titles. We ran an extensive series of experiments to compare various methods for document classification. The results showed that it is possible to reach a competitive performance for semantic annotation using solely the title of documents.

About video processing, we extended the initial set of technologies with a new machine learning method for concept annotation with complex concept labels, applied it to non-lecture videos and developed a first approach to the problem of performing temporal fragmentation on the lecture videos based on their transcripts. Furthermore, we improved the Video Analysis service (VIA). Regarding image processing, we have developed a prototypical search engine called SciFiS to take non-textual content into account. The text extraction allows for either searching directly for images or enhancing the search of the surrounding documents. We compared various text analyzers, search fields (only text or text with metadata) and data (manually annotated versus extracted text).

Both the user logging solution and the data analysis dashboard have already been deployed in the MOVING platform. We extended user logging to capture more data and we documented the platform interactions. We conducted a longitudinal evaluation of WevQuery and introduced the new assisted pattern mining. The longitudinal evaluation is still at an early stage, and we will use the provided feedback to improve WevQuery.

Finally, we have introduced an adapted UI for the GFV and also new visualisations. Most notably, GFV is optimised to show the graph in a web-page which limits the amount of available screen space and to reduce its latency. uRank combines lightweight text analytics and a visually augmented ranked list to assist in exploratory search of textual search result sets. It enables users to explore a document collection and refine information needs in terms of topical keywords. Additional bar charts and a tag cloud are also integrated to better filter the results.

7 Appendix - Common data model

Listing 3: Common data model v1.1 to integrate full-texts, metadata, HTML content and video data.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",

  "definitions": {
        "personName" : {
                "type": "string",
                "pattern": ".{2,}, ((.\\.( )?){0,1})*"
        },
        "date" : {
                "type": "string",
                "pattern": "[0-9]{4}-[0-9]{2}-[0-9]{2}"
        },
        "organisationName" : {
                "type": "string",
                "pattern": ".*"
        },
        "searchDomain" : {
                "type": "string",
                "enum": ["research", "learning", "funding"]
        },
        "source" : {
                "type": "string",
                "enum": ["SocialMediaWeb", "ZBWEconomics", "BTC2014", "videolectures.net",
                 "GESIS-SSOAR", "GESIS-SOLIS", "GESIS-SOFIS", "Laws_and_Regulations"]
        },
        "documentType" : {
                "type" : "string",
                "enum": ["project", "video/lecture", "video/debate", "video/demonstration",
                 "video/discussion_or_debate", "video/interview", "video/introduction",
                 "video/course", "video/opening", "video/invitation", "video/announcement",
                 "video/keynote", "video/self_introduction", "video/best_paper",
                 "video/press_conference", "video/video_conference_or_advertisment",
                 "video/advertisement", "video/invited_talk", "video/panel", "video/poster",
                 "video/promotional_video", "video/thesis_proposal", "video/thesis_defense",
                 "video/external_lecture", "video/event", "video/event_section",
                 "video/event_toc", "video/event_course", "video/project",
                 "video/project_group", "video/session", "video/referenced_course",
                 "video/curriculum", "video/default", "video/playlist", "video",
                 "video/tutorial", "video/summary", "document/full-text/summary",
                 "document/full-text/tutorial", "document/full-text/press_conference",
                 "document/full-text/book", "document/full-text/article",
                 "document/full-text/journal_article", "document/full-text/book_article",
                 "document/full-text/working_paper", "document/full-text/short_survey",
                 "document/full-text/report", "document/full-text/thesis",
                 "document/full-text/essay", "document/full-text/collection",
                 "document/full-text/textbook", "document/full-text/congress_report",
                 "document/full-text/commentary", "document/full-text/survey",
                 "document/full-text/review_article", "document/full-text/case_study",
                 "document/full-text/multi-volume_publication",
                 "document/full-text/goverment_document",
                 "document/full-text/law-regulation", "document/full-text/news_article",
                 "document/RDF/summary", "document/RDF/tutorial",
                 "document/RDF/press_conference", "document/RDF/book",
                 "document/RDF/article", "document/RDF/journal_article",
                 "document/RDF/book_article", "document/RDF/working_paper",
                 "document/RDF/short_survey", "document/RDF/report", "document/RDF/thesis",
                 "document/RDF/essay", "document/RDF/collection", "document/RDF/textbook",
                 "document/RDF/congress_report", "document/RDF/commentary",
                 "document/RDF/survey", "document/RDF/review_article",
                 "document/RDF/case_study", "document/RDF/multi-volume_publication",
                 "document/RDF/goverment_document", "document/RDF/law-regulation",
                 "document/RDF/news_article", "document/PDF/summary",
                 "document/PDF/tutorial", "document/PDF/press_conference",
                 "document/PDF/book", "document/PDF/article", "document/PDF/journal_article",
                 "document/PDF/book_article", "document/PDF/working_paper",
                 "document/PDF/short_survey", "document/PDF/report", "document/PDF/thesis",
                 "document/PDF/essay", "document/PDF/collection", "document/PDF/textbook",
                 "document/PDF/congress_report", "document/PDF/commentary",
                 "document/PDF/survey", "document/PDF/review_article",
                 "document/PDF/case_study", "document/PDF/multi-volume_publication",
                 "document/PDF/goverment_document", "document/PDF/law-regulation",
                 "document/PDF/news_article", "document/full-text", "document/RDF",
                 "document/PDF", "document", "website/organisation", "website/media-news",
                 "website/social-media-post/twitter", "website/social-media-post/google+",
                 "website/slides", "website/funding/travel-grant",
                 "website/funding/scholarship", "website/funding/project-grant",
                 "website/learning/moocs", "website/learning/webinar",
                 "website/social-media-post", "website/funding", "website/learning",
                 "website"]
        },
        "entityType" : {
                "type": "string",
                "enum": ["person", "organisation", "location", "socialmediaaccount",
                 "webauthor"]
        },
        "statisticLabel" : {
                "type": "string",
                "enum": ["retweets", "citations"]
        },
        "role": {
                "type": "string",
                "enum": ["author", "contributor", "editor", "related", "legislator"]
        },
        "locationRole": {
                "type": "string",
                "enum": ["law_applicable_region"]
        },
    "location": {
                "type": "object",
                "properties": {
                        "identifier": { "type": "string" },
                        "mentionID": { "type": "string" },
                        "URIs": {
                                "type": "array",
                                "items": { "type": "string", "format": "uri" }
                        },
                        "name":{ "type": "string" }
                },
                "required" : ["name"],
                "additionalProperties": false
        },
        "organisation": {
                "type" : "object",
                "properties": {
                        "identifier": { "type": "string" },
                        "mentionID": { "type": "string" },
                        "URIs": {
                                "type": "array",
                                "items": { "type": "string", "format": "uri" }
                        },
                        "name":{ "type": "string" },
                        "location": { "$ref": "#/definitions/location" }
                },
                "required": ["name"],
                "additionalProperties": false
        }
  },

  "type": "object",
  "properties": {
                "identifier": { "type": "string" },
                "sourceURLs": {
                        "type": "array",
                        "items": { "type": "string", "format": "uri" }
                },
                "documentURLs": {
                        "type": "array",
                        "items": { "type": "string", "format": "uri" }
                },
                "title": { "type": "string" },
                "abstract": { "type": "string" },
                "fulltext": { "type": "string" },
                "thumbnailURL": { "type": "string", "format": "uri" },
                "isPartOf": {
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "parentID": { "type": "string" },
                                        "position": { "type": "integer" }
                                },
                                "required": ["parentID"],
                                "additionalProperties": false
                        }
                },
                "hasParts": {
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "childID": { "type": "string" },
                                        "position": { "type": "integer" }
                                },
                                "required": ["childID"],
                                "additionalProperties": false
                        }
                },
                "metadata_persons": {
                        "type": "array",
                        "items": {
                                "type" : "object",
                                "properties": {
                                        "identifier": { "type": "string" },
                                        "mentionID": { "type": "string" },
                                        "URIs": {
                                                "type": "array",
                                                "items": { "type": "string", "format": "uri" }
                                        },
                                        "name": { "$ref": "#/definitions/personName" },
                                        "rawName": { "type": "string" },
                                        "roles":{
                                                "type": "array",
                                                "items": {
                                                        "$ref": "#/definitions/role"
                                                }
                                        },
                                        "email": { "type": "string", "format": "email" },
                                        "affiliations":{
                                                "type": "array",
                                                "items": { "$ref":
                                                 "#/definitions/organisation" }
                                        }
                                },
                                "required": ["name", "roles"],
                                "additionalProperties": false
                        }
                },
                "metadata_organisations":{
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "roles":{
                                                "type": "array",
                                                "items": { "$ref": "#/definitions/role" }
                                        },
                    "identifier": { "type": "string" },
                    "mentionID": { "type": "string" },
                    "URIs": {
                        "type": "array",
                        "items": { "type": "string", "format": "uri" }
                    },
                    "name":{ "type": "string" },
                    "location": { "$ref": "#/definitions/location" }

```

```
                                },
                                "required": ["name", "roles"],
                                "additionalProperties": false
                        }
                },
                "metadata_location":{
                        "type": "array",
                        "items": {
                                "type": "object",
                            "properties": {
                    "identifier": { "type": "string" },
                    "mentionID": { "type": "string" },
                    "URIs": {
                        "type": "array",
                        "items": { "type": "string", "format": "uri" }
                    },
                    "name":{ "type": "string" },
                    "rawName": { "type": "string" },
                    "lat": { "type": "number" },
                    "lon": { "type": "number" },
                    "roles":{
                        "type": "array",
                        "items": { "$ref": "#/definitions/locationRole" }
                    }
                 },
                                "required": ["roles"],
                                "additionalProperties": false
                        }
            },
            "metadata_venue":{
                        "type" : "object",
                        "properties": {
                                "identifier": { "type": "string" },
                                "mentionID": { "type": "string" },
                                "URIs": {
                                        "type": "array",
                                        "items": { "type": "string", "format": "uri" }
                                },
                                "name": { "type": "string" },
                                "rawName": { "type": "string" },
                                "startDate": {
                                        "anyOf": [
                                                { "$ref": "#/definitions/date" },
                                                { "type": "string", "format": "date-time" }
                                        ]
                                },
                                "endDate": {
                                        "anyOf": [
                                                { "$ref": "#/definitions/date" },
                                                { "type": "string", "format": "date-time" }
                                        ]
                                },
                                "volume": { "type": "integer" },
                                "issue": { "type": "integer" },
                                "location": { "$ref": "#/definitions/location" }
                        },
                        "additionalProperties": false
            },
                "startDate": {
                        "anyOf": [
                                { "$ref": "#/definitions/date" },
                                { "type": "string", "format": "date-time" }
                        ]
                },
                "endDate": {
                        "anyOf": [
                                { "$ref": "#/definitions/date" },
                                { "type": "string", "format": "date-time" }
                        ]
                },
                "source": { "$ref": "#/definitions/source" },
                "license": { "type": "string" },
                "openAccess": {
                        "type": "integer",
                        "minimum": 0,
                        "maximum": 1
                },
                "docType": { "$ref": "#/definitions/documentType" },
                "language": {
                        "type" : "string",
                        "pattern" : "[a-z]{2}"
                },
                "concepts": {
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "label": { "type": "string" },
                                        "URL": { "type": "string", "format": "uri" },
                                        "relevanceScore":  { "type": "number" }
                                },
                                "required": ["label"],
                                "additionalProperties": false
                        }
                },
                "sectors":{
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "label": { "type": "string" },
                                        "URL": { "type": "string", "format": "uri" },
                                        "relevanceScore":  { "type": "number" }
                                },
                                "required": ["label"],
                                "additionalProperties": false
                        }
                },
                "subjects":{
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "label": { "type": "string" },
                                        "URL": { "type": "string", "format": "uri" },
                                        "relevanceScore":  { "type": "number" }
                                },
                                "required": ["label"],
                                "additionalProperties": false
                        }
                },
                "keywords":{
                        "type": "array",
                        "items": { "type": "string" }
                },
                "references":{
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "identifier": { "type": "string" },
                                        "rawText": { "type": "string" }
                                },
                                "additionalProperties": false
                        }
                },
                "entities":{
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "identifier": { "type": "string" },
                                        "mentionID": { "type": "string" },
                                        "URIs": {
                                                "type": "array",
                                                "items": { "type": "string", "format": "uri" }
                                        },
                                        "label": { "type": "string" },
                                        "confidence": {
                                                "type": "number",
                                                "minimum": 0,
                                                "maximum": 1
                                        },
                                        "type": { "$ref": "#/definitions/entityType" }
                                },
                                "required": ["label"],
                                "additionalProperties": false
                        }
                },
                "external_statistics": {
                        "type": "array",
                        "items": {
                                "type": "object",
                                "properties": {
                                        "label": { "$ref": "#/definitions/statisticLabel" },
                                        "value": { "type": "number" }
                                },
                                "required": ["label", "value"],
                                "additionalProperties": false
                        }
                },
                "searchDomains" : {
                        "type": "array",
                        "items": { "$ref": "#/definitions/searchDomain" }
                },
        "lawSpecific_metadata" : {},
        "temporal_metadata": {}
    },
        "required": ["title", "source", "docType"],
        "additionalProperties": false

}
```

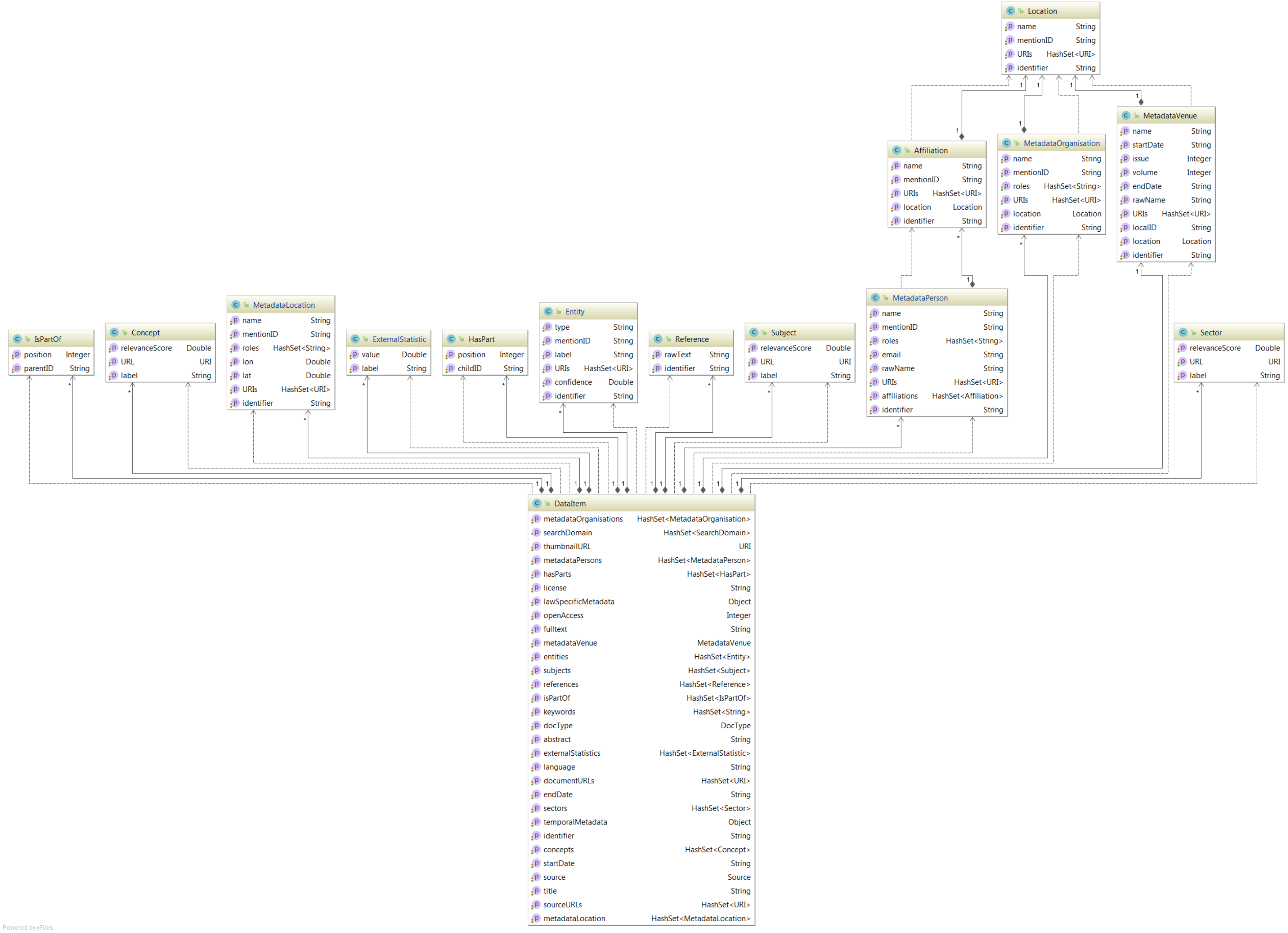


Figure 58: Diagram of the common data model.

References

Abdel-Qader, M., & Scherp, A. (2017, September). Towards Understanding the Evolution of Vocabulary Terms in Knowledge Graphs. *ArXiv e-prints*.

Abdel-Qader, M., & Scherp, A. (2016). Qualitative analysis of vocabulary evolution on the linked open data cloud. In *Profiles@ eswc*.

Agrawal, R., Srikant, R., & others. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487–499).

Ai, Q., Yang, L., Guo, J., & Croft, W. B. (2016a). Analysis of the paragraph vector model for information retrieval. In *Proceedings of the 2016 acm on international conference on the theory of information retrieval* (pp. 133–142).

Ai, Q., Yang, L., Guo, J., & Croft, W. B. (2016b). Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *Proceedings of the 39th international acm sigir conference on research and development in information retrieval* (pp. 869–872).

Akers, D., Simpson, M., Jeffries, R., & Winograd, T. (2009). Undo and erase events as indicators of usability problems. In *Proceedings of the 27th international conference on human factors in computing systems* (pp. 659–668). ACM. Retrieved 2012-03-22, from http://doi.acm.org/10.1145/1518701.1518804 doi: 10.1145/1518701.1518804

Amati, G., & Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 357–389. doi: 10.1145/582415.582416

Apaolaza, A., Harper, S., & Jay, C. (2013). Understanding users in the wild. In *Proc. of the 10th international cross-disciplinary conference on web accessibility* (pp. 13:1–13:4). Retrieved from http://doi.acm.org/10.1145/2461121.2461133 doi: 10.1145/2461121.2461133

Apaolaza, A., & Vigo, M. (2017). WevQuery: Testing hypotheses about web interaction patterns. , *1*(1), 4:1–4:17. Retrieved from http://doi.acm.org/10.1145/3095806 doi: 10.1145/3095806

Arias, M., Fernández, J. D., Martínez-Prieto, M. A., & de la Fuente, P. (2011). An empirical study of real-world SPARQL queries. *arxiv.org*. Retrieved from http://arxiv.org/abs/1103.5043

Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., … Weinberger, K. (2010). Learning to rank with (a lot of) word features. *Information retrieval*, *13*(3), 291–314.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer vision–eccv 2006* (pp. 404–417). Springer.

Benedetti, F., Bergamaschi, S., & Po, L. (2014). Online index extraction from linked open data sources. In *LD4IE.*

Benedetti, F., Bergamaschi, S., & Po, L. (2015). Exposing the underlying schema of LOD sources. In *Joint IEEE/WIC/ACM WI and IAT.*

Bi, J., & Zhang, T. (2004). Support vector classification with input data uncertainty. In *Nips.*

Bienia, I., Fessl, A., Günther, F., Herbst, S., Maas, A., & Wiese, M. (2017). *Deliverable 1.1: User requirements and specification of the use cases* (Tech. Rep.). MOVING. Retrieved from http://moving-project.eu/wp-content/uploads/2017/04/moving_d1.1_v1.0.pdf

Blume, T., Böschen, F., Galke, L., Saleh, A., Scherp, A., Schulte-Althoff, M., … Gottron, T. (2017). *Deliverable 3.1: Technologies for MOVING data processing and visualisation v1.0* (Tech. Rep.). MOVING. Retrieved from http://moving-project.eu/wp-content/uploads/2017/04/moving_d3.1_v1.0.pdf

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010.* Springer.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade.* Springer.

Bousquet, O., & Bottou, L. (2008). The tradeoffs of large scale learning. In *Advances in neural information processing systems.*

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Wadsworth.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on machine learning* (pp. 89–96).

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on machine learning* (pp. 129–136).

Chau, M. (2011). Visualizing web search results using glyphs: Design and evaluation of a flower metaphor. *ACM Transactions on Management Information Systems*, *2*(1), 2. doi: 10.1145/1929916.1929918

Chawuthai, R., Takeda, H., Wuwongse, V., & Jinbo, U. (2016). Presenting and preserving the change in taxonomic knowledge for linked data. *Semantic Web*, *7*(6), 589–616.

Chen, R.-C., Spina, D., Croft, W. B., Sanderson, M., & Scholer, F. (2015). Harnessing semantics for answer sentence retrieval. In *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval* (pp. 21–27). ACM.

Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, *24*(9), 1537–1555.

Christodoulou, K., Paton, N. W., & Fernandes, A. A. A. (2013). Structure inference for Linked Data sources using clustering. In *Joint edbt/icdt.*

Christopher, D. M., Prabhakar, R., & Hinrich, S. (2008). Introduction to information retrieval. *An Introduction To Information Retrieval*, *151*, 177.

Ciglan, M., Nørvåg, K., & Hluchý, L. (2012). The SemSets model for ad-hoc semantic list search. In *Www.*

Cohen, D., Ai, Q., & Croft, W. B. (2016). Adaptability of neural networks on varying granularity ir tasks. *arXiv preprint arXiv:1606.07565*.

Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice* (Vol. 283). Addison-Wesley Reading.

Dang, V. (2010). Ranklib-a library of ranking algorithms.. `https://sourceforge.net/p/lemur/wiki/RankLib/`. (Online; accessed 19 January 2018)

De Giacomo, G., & Lenzerini, M. (1996). TBox and ABox reasoning in expressive description logics. In *Aaai technical reports.*

Dev, H., & Liu, Z. (2017). Identifying frequent user tasks from application logs. In *Proceedings of the 22nd international conference on intelligent user interfaces* (pp. 263–273). ACM. Retrieved 2017-04-03, from `http://doi.acm.org/10.1145/3025171.3025184` doi: 10.1145/3025171.3025184

di Sciascio, C., Sabol, V., & Veas, E. (2016). Rank as you go: User-driven exploration of search results. In *Proceedings of the 21st international conference on intelligent user interfaces* (pp. 118–129). USA / Vereinigte Staaten: Association of Computing Machinery.

Din, B. T. (2012). IRRA at TREC 2012 : Divergence From Independence (DFI) The Heuristic Approach for Early Precision. , 1–6.

Ding, L., Shinavier, J., Shangguan, Z., & McGuinness, D. L. (2010). SameAs networks and beyond: Analyzing deployment status and implications of owl:sameAs in Linked Data. In *Iswc.*

Dividino, R., Scherp, A., Gröner, G., & Grotton, T. (2013). Change-a-lod: does the schema on the linked data cloud change or not? In *Proceedings of the fourth international conference on consuming linked data-volume 1034* (pp. 87–98).

Dividino, R. Q., Scherp, A., Gröner, G., & Grotton, T. (2013). Change-a-lod: Does the schema on the linked data cloud change or not? In *COLD* (Vol. 1034). CEUR-WS.org.

European Mathematical Society. (2014). *Equivalence relation.* Retrieved from `http://www.encyclopediaofmath.org/index.php?title=Equivalence_relation&oldid=35990`

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 363–370). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `https://doi.org/10.3115/1219840.1219885` doi: 10.3115/1219840.1219885

Fournier-Viger, P., Lin, J. C.-W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016). The SPMF open-source data mining library version 2. In *Machine learning and knowledge discovery in databases* (pp. 36–40). Springer, Cham. Retrieved 2017-09-23, from `https://link.springer.com/chapter/10.1007/978-3-319-46131-1_8` doi: 10.1007/978-3-319-46131-1_8

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research.*

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, *14*(771-780), 1612.

Fruchterman, T. M., & Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, *21*(11), 1129–1164. doi: 10.1002/spe.4380211102

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Ijcai* (Vol. 7, pp. 1606–1611).

Galke, L., Mai, F., Schelten, A., Brunsch, D., & Scherp, A. (2017, May). Using Titles vs. Full-text as Source for Automated Semantic Document Annotation. In *Proceedings of the 9th international conference on knowledge capture (k-cap).*

Galke, L., Saleh, A., & Scherp, A. (2017). Evaluating the impact of word embeddings on similarity scoring in practical information retrieval. In *Informatik 2017, 47. jahrestagung der gesellschaft für informatik.*

Ganguly, D., Roy, D., Mitra, M., & Jones, G. J. (2015). Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 795–798).

Garofalakis, M. N., Gehrke, J., & Rastogi, R. (Eds.). (2016). *Data stream management.*

Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, *49*(3), 291–304.

Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial intelligence*, *40*(1-3), 11–61.

Gkalelis, N., Mezaris, V., Kompatsiaris, I., & Stathaki, T. (2013). Mixture subclass discriminant analysis link to restricted gaussian model and other generalizations. *Neural Networks and Learning Systems, IEEE Trans. on*, *24*(1), 8–21.

Goldman, R., & Widom, J. (1997). DataGuides: Enabling query formulation and optimization in semistructured databases. In *Vldb.*

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations (3rd ed.).* Baltimore, MD, USA: Johns Hopkins University Press.

Goossen, F., Ijntema, W., Frasincar, F., Hogenboom, F., & Kaymak, U. (2011). News personalization using the CF-IDF semantic recommender. In *Web intelligence, mining and semantics.*

Gottfried, S., Grunewald, P., Pournaras, A., Collyda, C., Fessl, A., Hasitschka, P., … Scherp, A. (2017). *Deliverable 4.1: Definition of platform architecture and software development configuration* (Tech. Rep.). MOVING. Retrieved from http://moving-project.eu/wp-content/uploads/2017/02/moving_d4.1-v1.0.pdf

Gottfried, S., Pournaras, A., Collyda, C., Mezaris, V., Backes, T., Wertner, A., … Saleh, A. (2017). *Deliverable 4.2: Initial responsive platform prototype, modules and common communication protocol* (Tech. Rep.). MOVING. Retrieved from http://moving-project.eu/wp-content/uploads/2016/02/moving_d4.2_v1.0.pdf

Gottron, T., Knauf, M., & Scherp, A. (2015). Analysis of schema structures in the linked open data graph based on unique subject uris, pay-level domains, and vocabulary usage. *Distributed and Parallel Databases*, *33*(4), 515–553.

Gottron, T., Scherp, A., Krayer, B., & Peters, A. (2013). LODatio: using a schema-level index to support users infinding relevant sources of linked data. In *K-cap.*

Große-Bölting, G., Nishioka, C., & Scherp, A. (2015). A comparison of different strategies for automated semantic document annotation. In *Knowledge capture.*

Guha, R. V., Brickley, D., & Macbeth, S. (2016). Schema. org: Evolution of structured data on the web. *Communications of the ACM*, *59*(2), 44–51.

Hall, M. A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *proceedings of the 17th international conference on data engineering* (pp. 215–224).

Hecht-Nielsen, R. (1988). Theory of the backpropagation neural network. *Neural Networks*, *1*(Supplement-1), 445–448.

Heß, A., Dopichaj, P., & Maaß, C. (2008). Multi-value classification of very short texts. In *Advances in artificial intelligence.* Springer.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, *abs/1207.0580*. Retrieved from http://arxiv.org/abs/1207.0580

Holten, D., & van Wijk, J. J. (2009). Force-directed edge bundling for graph visualization. In *Proceedings of the 11th eurographics/IEEE-VGTC conference on visualization* (pp. 983–998). Berlin, Germany: The Eurographs Association; John Wiley; Sons, Ltd. doi: 10.1111/j.1467-8659.2009.01450.x

Hose, K., Schenkel, R., Theobald, M., & Weikum, G. (2011). Database foundations for scalable rdf processing. In *Reasoning web. semantic technologies for the web of data: 7th international summer school 2011, galway, ireland, august 23-27, 2011, tutorial lectures* (pp. 202–249). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-23032-5_4

Huang, M., Névéol, A., & Lu, Z. (2011). Recommending MeSH terms for annotating biomedical articles. *Am. Medical Informatics Association*, *18*(5).

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data..

J. A. Nelder, R. W. M. W. (1972). Generalized linear models. *Royal Statistical Society*, *135*(3). Retrieved from http://www.jstor.org/stable/2344614

Järvelin, K., & Kekäläinen, J. (2002, October). Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, *20*(4), 422–446. Retrieved from http://doi.acm.org/10.1145/582415.582418 doi: 10.1145/582415.582418

Jett, J., Nurmikko-Fuller, T., Cole, T. W., Page, K. R., & Downie, J. S. (2016). Enhancing scholarly use of digital libraries: A comparative survey and review of bibliographic metadata ontologies. In *JCDL*.

Jiang, L., Yu, S.-I., Meng, D., Mitamura, T., & Hauptmann, A. G. (2015). Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *Acm int. conf. on multimedia retrieval*.

Jiang, Y.-G., Bhattacharya, S., Chang, S.-F., & Shah, M. (2013). High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval*, *2*(2), 73–101.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *ICML* (pp. 143–151). Morgan Kaufmann.

Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., & Hogan, A. (2013). Observing linked data dynamics. In *Eswc* (Vol. 7882).

Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., & Hogan, A. (2013). Observing linked data dynamics. In *Extended semantic web conference* (pp. 213–227).

Käfer, T., Umbrich, J., Hogan, A., & Polleres, A. (2012). Towards a dynamic linked data observatory. *LDOW at WWW*.

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Systems, Man, and Cybernetics*, *15*(4), 580–585.

Kienreich, W., Wozelka, R., Sabol, V., & Seifert, C. (2012). Graph visualization using hierarchical edge routing and bundling. In *Proceedings of the 3rd international eurovis workshop on visual analytics* (pp. 97–101). Vienna, Austria: The Eurographics Association. doi: 10.2312/PE/EuroVAST/EuroVA12/097-101

Kim, K., Sefid, A., & Giles, C. L. (2017). Scaling author name disambiguation with cnf blocking. *arXiv preprint arXiv:1709.09657*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*. Retrieved from http://arxiv.org/abs/1412.6980

Konrath, M., Gottron, T., Staab, S., & Scherp, A. (2012). SchemEX - efficient construction of a data catalogue by stream-based indexing of Linked Data. *J. Web Sem.*, *16*, 52–58.

Levin, M., Krawczyk, S., Bethard, S., & Jurafsky, D. (2012). Citation-based bootstrapping for large-scale author disambiguation. *Journal of the Association for Information Science and Technology*, *63*(5), 1030–1047.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Machine Learning Research*, *5*.

Lin, M., Chau, M., Cao, J., & Nunamaker Jr, J. F. (2005). Automated video segmentation for lecture videos: A linguistics-based approach. *International Journal of Technology and Human Interaction (IJTHI)*, *1*(2), 27–45.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, *3*(3), 225–331.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. the proceedings of the seventh ieee international conference on* (Vol. 2, pp. 1150–1157).

Lund, A. M. (2001). Measuring usability with the use questionnaire12. *Usability interface*, *8*(2), 3–6.

Luo, Y., Fletcher, G. H. L., Hidders, J., Wu, Y., & Bra, P. D. (2013). External memory k-bisimulation reduction of big graphs. In *Cikm*.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*. Cambridge.

Markatopoulou, F., Galanopoulos, D., Mezaris, V., & Patras, I. (2017). Query and keyframe representations for ad-hoc video search. In *Proceedings of the 2017 acm on international conference on multimedia retrieval* (pp. 407–411). ACM. Retrieved from http://doi.acm.org/10.1145/3078971.3079041 doi: 10.1145/3078971.3079041

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *Aaai-98 workshop on learning for text categorization* (Vol. 752, pp. 41–48).

McHugh, J., Abiteboul, S., Goldman, R., Quass, D., & Widom, J. (1997). Lore: a database management system for semistructured data. *SIGMOD Record*, *26*(3), 54–66.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153–157.

Menestrina, D., Whang, S. E., & Garcia-Molina, H. (2010). Evaluating entity resolution results. *Proceedings of the VLDB Endowment*, *3*(1-2), 208–219.

Metzler, D., & Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, *10*(3), 257–274.

Metzler, D., & Kanungo, T. (2008). Machine learned sentence selection strategies for query-biased summarization. In *SIGIR Learning to Rank Workshop* (pp. 40–47).

Meusel, R., Bizer, C., & Paulheim, H. (2015). A web-scale study of the adoption and evolution of the schema. org vocabulary over time. In *Proceedings of the 5th international conference on web intelligence, mining and semantics* (p. 15).

Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., & Gómez-Pérez, A. (2016). Collaborative ontology evolution and data quality-an empirical analysis. In *International experiences and directions workshop on owl* (pp. 95–114).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Milojević, S. (2013). Accuracy of simple, initials-based methods for author name disambiguation. *Journal of Informetrics*, *7*(4), 767–773.

Mooney, C. H., & Roddick, J. F. (2013, March). Sequential pattern mining – approaches and algorithms. *ACM Comput. Surv.*, *45*(2), 19:1–19:39. Retrieved from `http://doi.acm.org/10.1145/2431211.2431218` doi: 10.1145/2431211.2431218

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml.*

Nam, J., Kim, J., Mencía, E. L., Gurevych, I., & Fürnkranz, J. (2014). Large-scale multi-label text classification. revisiting neural networks. In *Machine learning and knowledge discovery in databases.* Springer.

Nestorov, S., Ullman, J. D., Wiener, J. L., & Chawathe, S. S. (1997). Representative Objects: Concise representations of semistructured, hierarchial data. In *ICDE.*

Neumann, T., & Moerkotte, G. (2011). Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins. In *Icde.*

Nishioka, C., Große-Bölting, G., & Scherp, A. (2015). Influence of time on user profiling and recommending researchers in social media. In *iknow.* ACM.

Over, P., Awad, G., Fiscus, J., Michel, M., Joy, D., Smeaton, A. F., … Ordelman, R. (2015). TRECVID 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. of trecvid 2015.*

Papadakis, G., Svirsky, J., Gal, A., & Palpanas, T. (2016). Comparative analysis of approximate blocking techniques for entity resolution. *Proceedings of the VLDB Endowment*, *9*(9), 684–695.

Perer, A., & Wang, F. (2014). Frequence: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on intelligent user interfaces* (pp. 153–162). ACM. Retrieved 2016-10-07, from `http://doi.acm.org/10.1145/2557500.2557508` doi: 10.1145/2557500.2557508

Qi, Z., Tian, Y., & Shi, Y. (2013). Robust twin support vector machine for pattern classification. *Pattern Recognition*, *46*(1), 305–316.

Qin, T., & Liu, T.-Y. (2013). Introducing LETOR 4.0 Datasets. *CoRR*. Retrieved from `http://arxiv.org/abs/1306.2597`

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *In proceedings of the lrec 2010 workshop on new challenges for nlp frameworks.*

Robertson, S. E., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In W. B. Croft & C. J. van Rijsbergen (Eds.), *Proceedings of the 17th annual international ACM-SIGIR conference on research and development in information retrieval. dublin, ireland, 3-6 july 1994 (special issue of the SIGIR forum)* (pp. 232–241). ACM/Springer.

Robertson, S. E., Walker, S., Beaulieu, M., & Willett, P. (1999). Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. *Nist Special Publication SP*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision (IJCV)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Sakr, S., & Al-Naymat, G. (2010). Graph indexing and querying: a review. *Int. Journal of Web Information Systems*, *6*(2), 101–120.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5).

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Sangiorgi, D. (2009). On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, *31*(4), 15:1–15:41.

Schaible, J., Gottron, T., & Scherp, A. (2014). Survey on common strategies of vocabulary reuse in linked open data modeling. In *European semantic web conference* (pp. 457–472).

Schaible, J., Gottron, T., & Scherp, A. (2016). TermPicker: Enabling the reuse of vocabulary terms by exploiting data from the Linked Open Data cloud. In *ESWC*.

Schmachtenberg, M., Bizer, C., & Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *International semantic web conference* (pp. 245–260).

Shah, R. R., Yu, Y., Shaikh, A. D., & Zimmermann, R. (2015). Trace: Linguistic-based approach for automatic lecture video segmentation leveraging wikipedia texts. In *Multimedia (ism), 2015 ieee international symposium on* (pp. 217–220).

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014a). A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd acm international conference on conference on information and knowledge management* (pp. 101–110).

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014b). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd international conference on world wide web* (pp. 373–374).

Shoval, P., & Kuflik, T. (2004). Effectiveness of title-search vs. full-text search in the web. *International Journal of Information Theories and Applications*.

Spahiu, B., Porrini, R., Palmonari, M., Rula, A., & Maurino, A. (2016). ABSTAT: ontology-driven Linked Data summaries with pattern minimalization. In *ESWC satellite events, revised selected papers*.

Spyromitros, E., Tsoumakas, G., & Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. In *Artificial intelligence*. Springer.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*(3).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE conf. on computer vision and pattern recognition, CVPR 2015, boston, ma, usa, june 7-12, 2015* (pp. 1–9).

Tanaka, E. A., Nozawa, S. R., Macedo, A. A., & Baranauskas, J. A. (2015). A multi-label approach using binary relevance and decision trees applied to functional genomics. *Journal of Biomedical Informatics*, *54*.

Tang, L., Rajan, S., & Narayanan, V. K. (2009). Large scale multi-label classification via metalabeler. In *WWW*. ACM.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology - volume 1* (pp. 173–180). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `https://doi.org/10.3115/1073445.1073478` doi: 10.3115/1073445.1073478

Tran, T., Haase, P., & Studer, R. (2009). Semantic search — using graph-structured semantic models for supporting the search process. In *Int. conf. on conceptual structures* (pp. 48–65). Springer. Retrieved from `http://dx.doi.org/10.1007/978-3-642-03079-6_5` doi: 10.1007/978-3-642-03079-6_5

Tran, T., Ladwig, G., & Rudolph, S. (2013). Managing structured and semi-structured RDF data using structure indexes. *IEEE Transactions on Knowledge and Data Engineering*, *25*(9), 2076-2089.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *IJDWM*, *3*(3).

Tzelepis, C., Gkalelis, N., Mezaris, V., & Kompatsiaris, I. (2013). Improving event detection using related videos and relevance degree support vector machines. In *Proc. of the 21st acm int. conf. on multimedia* (pp. 673–676).

Tzelepis, C., Mezaris, V., & Patras, I. (2017). Linear maximum margin classifier for learning from uncertain data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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.

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *Ieee international conference on computer vision*. Sydney, Australia.

Ware, C. (2012). *Information visualization: perception for design*. Elsevier.

Wu, Q., Burges, C. J., Svore, K. M., & Gao, J. (2010). Adapting boosting for information retrieval measures. *Information Retrieval*, *13*(3), 254–270.

Xu, J., & Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international acm sigir conference on research and development in information retrieval* (pp. 391–398).

Xu, Z., Yang, Y., & Hauptmann, A. G. (2015). A discriminative cnn video representation for event detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1798–1807).

Zamani, H., & Croft, W. B. (2016). Embedding-based query language models. In *Proceedings of the 2016 acm on international conference on the theory of information retrieval* (pp. 147–156).

Zgraggen, E., Drucker, S. M., Fisher, D., & DeLine, R. (2015). (s,qu)eries: Visual regular expressions for querying and exploring event sequences. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (pp. 2683–2692). ACM. Retrieved 2017-04-03, from http://doi.acm.org/10.1145/2702123.2702262 doi: 10.1145/2702123.2702262

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to Ad Hoc information retrieval. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, *22*(2), 334–342. Retrieved from http://portal.acm.org/citation.cfm?doid=383952.384019 doi: 10.1145/383952.384019

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Icml*.

Zhang, W., Stella, X. Y., & Teng, S.-H. (2012). Power SVM: Generalization with exemplar classification uncertainty. In *Computer vision and pattern recognition (cvpr), 2012 ieee conf. on* (pp. 2144–2151).

Zhang, Y., Rahman, M. M., Braylan, A., Dang, B., Chang, H.-L., Kim, H., … others (2016). Neural information retrieval: A literature review. *arXiv preprint arXiv:1611.06792*.

Zuccon, G., Koopman, B., Bruza, P., & Azzopardi, L. (2015). Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th australasian document computing symposium* (p. 12).