Deliverable 3.3: Technologies for MOVING data processing and visualisation v3.0

Iacopo Vagliano, Till Blume, Lukas Galke, Florian Mai, Ahmed Saleh/ZBW
Alexandros Pournaras, Nikolaos Gkalelis, Damianos Galanopoulos, Vasileios Mezaris/CERTH
Ilija Šimić, Vedran Sabol/KC
Aitor Apaolaza, Markel Vigo/UMAN
Andrea Zielinski, Peter Mutschke/GESIS

31/01/2019

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	31/01/2019
Actual date of delivery	31/01/2019
Deliverable number	3.3
Deliverable name	Technologies for MOVING data processing and visualisation v3.0
File	`MOVING_D3.3_v1.0.tex`
Nature	Report
Status & version	Final & v1.0
Number of pages	131
WP contributing to the deliverable	3
Task responsible	ZBW
Other contributors	CERTH, KC, UMAN, GESIS
Author(s)	Iacopo Vagliano, Till Blume, Lukas Galke, Florian Mai, Ahmed Saleh/ZBW Alexandros Pournaras, Nikolaos Gkalelis, Damianos Galanopoulos, Vasileios Mezaris/CERTH Ilija Šimić, Vedran Sabol/KC Aitor Apaolaza, Markel Vigo/UMAN Andrea Zielinski, Peter Mutschke/GESIS
Quality Assessors	Tanja Zdolšek Draksler
EC Project Officer	Hinano SPREAFICO
Keywords	Technologies, data acquisition, data processing, data visualisation, user data logging, common data model

Executive summary

This deliverable *D3.3: Technologies for MOVING data processing and visualisation v3.0* provides a final update on the common data model as well as on the set of data acquisition, data processing, user logging and data visualisation components used in the MOVING platform. The common data model has been subject to minor changes in order to adapt to the changes in the duplicate-detection component and to the new entity extraction module (Section 2). Advances in data acquisition and data processing are described in Section 3 and comprises different techniques. Specifically, the three crawlers for webpages and social media content (Section 3.1) have been further improved, while the flexible schema-level index for Linked Open Data to harvest additional metadata has been modified to support incremental updates, given the distributed nature of the data represented (Section 3.2). The duplicate detection service (Section 3.3) has been completed with the actual removal of duplicates, and two new components for entity extraction and linking have been integrated in the platform (Section 3.4). A new model for detecting and forecasting trending topics on data streams, such as Twitter posts, has been developed (Section 3.5). Further experiments have been conducted with new methods to use machine learning for multi-label classification relying only on titles (Section 3.6). An update on the improved techniques for video fragmentation, concept detection and video transcript analysis is also outlined (Section 3.7). The changes in logging of user interaction data as well as a user study to evaluate WevQuery are explained in Section 4. The new features of the visualisations are described in Section 5. Finally, we summarise the main contribution for each task highlighting the achievements of the third year of the project in Section 6.

Table of contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Explanation
ACL ARC	Association for Computational Linguistics Anthology Reference Corpus
ANN	Artificial neural networks
API	Application Programming Interface
ASR	Automatic Speech Recognition
ATS	Adaptive Training Support
AVS	Ad-hoc Video Search
BoW	Bag of Words
CBOW	Continuous Bag of Words
CBUQ	Component-based Usability Questionnaire
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CSE	Complex Schema Element
DIS	Data Integration Service
DR	Dimensionalty Reduction
EP	Eigenproblem
EQR	Equivalence Relation
FDC	Focused web-Domain Crawler
FLuID	Formal schema-Level Index model for the web of Data
GRU	Gated Recurrent Unit
GVF	Graph Visualisation Framework
HTML	HyperText Markup Language
ID	Identifier
IR	Information Retrieval
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LRDA	Linear Regression Discriminant Analysis
LS	Least Squares
LSTM	Long Short-Term Memory
MAP	Mean Average Precision
MeSH	Medical Subject Headings
ML	Machine Learning
MLP	Multi-Layer Perceptron
NER	Named Entity Recognition
NERC	Named Entity Classification
NERD	Named Entity Disambiguation
NEL	Named Entity Linking
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NoSQL	Not only SQL
NP	Noun Phrase
NZEP	Nonzero Eigenpairs
PAY	Paylod function
PC	Property Cluster
PDAQ	Perceived Difficulty Assessment Questionnaire
PDF	Portable Document Format
POC	Property-Object Cluster
POS	Part of Speech
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RNN	Recurrent Neural Network
RMSE	Root Mean Sqared Error
SEC	Search-Engine-based web Crawler
SD	Standard Deviation

Abbreviation	Explanation
SG	Schema Graph
SLI	Schema Level Index
SOLIS	Social Science Literature Information System
SPMF	Sequential Pattern Mining Framework
SQL	Structured Query Language
SRDA	Spectral Regression Discriminant Analysis
SSM	Social Stream Manager
STW	Standard-Thesaurus Wirtschaft
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SW	Semantic Web
TF-IDF	Term Frequency - Inverse Document Frequency
TM	Text Mining
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USE	Usefulness, Satisfaction, and Ease of use
UX	User Experience
VIA	Video Analysis Service
WP	Work package
XMLC	Extreme Multi-Label Classification

1 Introduction

1.1 History of the document

Table 2: Document history.

Date	Version
13/11/2018	v0.1: table of content ready for quality assurance
21/11/2018	v0.2: table of content's comments addressed
14/01/2019	v0.3: content ready for quality assurance
31/01/2019	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 deliverables D3.1: *Technologies for MOVING data processing and visualisation v1.0* (Blume et al., 2017) and D3.2: *Technologies for MOVING data processing and visualisation v2.0* (Vagliano et al., 2018). Some minor improvements in the services that implement the technologies described may be outlined in the upcoming D4.3: *Final responsive platform prototype, modules and common communication protocol*, due in month 36. In this report, we 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 used in the MOVING platform. In Section 2, we briefly present some minor improvements in the common data model. We describe the data acquisition and the data processing components in Section 3, and the user logging components in Section 4. Finally, we illustrate the data visualisation components in Section 5. Figure 1 shows an overview of the interaction between components developed in work package (WP) 3 as well as their interconnection to the MOVING web application developed in WP 4.

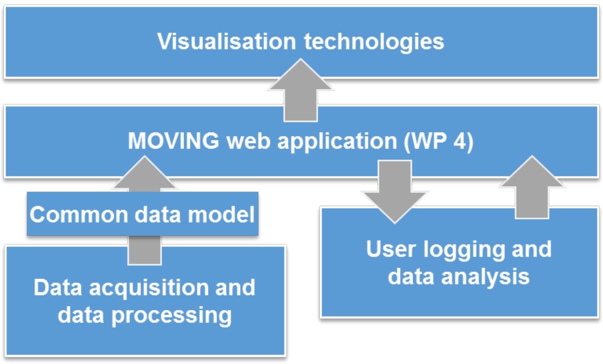

Visualisation technologies
MOVING web application (WP 4)
Common data model
Data acquisition and data processing
User logging and data analysis

Figure 1: Overview of the interaction between components developed in WP 3 and the MOVING web application (WP 4). When not indicated otherwise the components belongs to WP 3.

2 Common data model

The MOVING platform provides access to a large variety of documents coming from different data sources. The common data model is the foundation for data integration in the platform. 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 in deliverable D3.2 (Vagliano et al., 2018), which improved some fields from the previous version v1.0 and introduced some new specific metadata fields, namely, `videoSpecific_metadata`, `lawSpecific_metadata`, `temporal_metadata`, and `fundingSpecific_metadata`. This new common data model v1.1 has shown to be flexible enough to cope with all different data sources that we integrated. Thus, the overall model has not been changed.

In this final update (v1.2), we included a new specific metadata field, `duplicateSpecific_metadata`. Furthermore, we added `startChar` and `endChar` to the `entities` field in order to enable visual highlighting within a single document as well as computing dependencies between entities. The final common data model v1.2 can be viewed in the appendix Listing 4. In the list below, we briefly recall the main feature of the specific metadata fields introduced in v1.1 as well as describe the new `duplicateSpecific_metadata` field:

`videoSpecific_metadata` represents all the metadata produced by the video analysis and contains two basic sub-fields: `video_concepts` and `video_fragments`. The first represents the concepts related to the video that are produced either from visual analysis (for crawled videos) or transcript analysis (for lecture videos). It contains two sub-fields: `labels`, which is the name of the concept, and `relevanceScore`, which determines the likelihood that this concept is relevant. The `video_fragments` field represents the temporal fragmentation of the video and contains the following sub-fields: `URL`, which is a link to the start of the specific video fragment; `thumbnailURL`, a link to a thumbnail for the video fragment; `start` and `end`, which determine the starting and ending time of the fragment; and `video_concepts`, which represents the concepts for the fragment.

`fundingSpecific_metadata` contains metadata about the funding sources crawled. The metadata are specific to the Horizon 2020 topics, since we are crawling only those topics. `EU_pillar` and `category` are forms of categorisation of the different kinds of Horizon calls. The `callDeadline` field is the submission deadline of the call and `external_identifier` is an identifier assigned to every funding topic by European Commission.

`duplicateSpecific_metadata` contains information about the duplicates' cluster, showing which documents belong to the same cluster, and which documents are prioritised based on their metadata. The basic idea is that only the first, top-ranked document will be presented to the user. There are three basic sub-fields: `docID`, `cluster`, and `rank`. The first is the document's identifier, e. g. `AWBZXF2cRhS18e7PDcm2`, the second is the identifier for the cluster, e.g. `32438`, and the third is the ranking of the document in the cluster, e. g. `2`. Highest priority is given to documents with a low rank.

`lawSpecific_metadata` contains metadata about different laws and regulations. Each regulation consists of several articles and paragraphs, while each paragraph can consist of several sub paragraphs. This field contains information about the legislator, abbreviation, structuring number, status, sector, location, announcement date and commencement date of the law.

`temporal_metadata` Regulations are subject to continuous changes, each change consists of its own metadata. For this reason, we are representing the laws and regulations in two fields. The `temporal_metadata` field contains information about the changes of the respective law over time. In more details, this field consists of the following subfields: (1) `title` usually contains general information about the changed articles, (2) `abbreviation` is the abbreviation of the change, (3) `kind` represents if this is an initial version, partially altered or rewritten version of the regulation, (4) `announcement data`, `issuing date`, `announcement date`, `commencement date` and `end date` of the change, (5) `changedParts` contains more information about the changed parts of the regulation, its legislator(s) and ordinations, as well as the current status of this change.

3 Data acquisition and data processing

3.1 Crawling of social media, websites and videos

In this section, we will focus on the new features implemented for the crawlers. The MOVING crawlers consist of 3 individual crawlers:

- The Focused web-Domain Crawler (FDC) capable of crawling full domains.
- The Social Stream Manager (SSM) which monitors social media.
- The Search Engine-based Crawler (SEC) which harnesses the Google API to retrieve search results.

More details about the functionality of the crawlers can be found in the previous deliverables D3.1 (Blume et al., 2017) and D3.2 (Vagliano et al., 2018).

3.1.1 Crawling of funding sources

We implemented a mechanism to monitor the Horizon 2020 funding topics. The mechanism is part of the Social Stream Manager and it is scheduled to run once every day. It collects the data about the funding topics from a JSON file[1]. The structure of a single funding topic can be seen in Listing 1. The file is updated every time a new funding topic is inserted. The mechanism first checks the status of the call. If the call is closed, it skips the topic as there is no need to index an expired funding topic. The next step is to retrieve the full HTML page of the topic and transform its metadata into the common data model format. The resulting document looks similar to Listing 2. Last, the crawler checks if there are possible duplicates of the produced document already indexed before indexing it. This is a simple duplicate check which only takes into account the URL by looking for similar URLs in indexed documents. More advanced duplicate detection and removal techniques are addressed in Section 3.1.2. The indexing is done through the data integration service (DIS).

Listing 1: Funding topic data

```
{
"topicId":1269891,
"ccm2Id":31085907,
"subCallId":916870,
"topicFileName":"altfi-01-2017",
"callProgramme":"H2020",
"callFileName":"h2020-altfi-2017",
"callStatus":"Closed",
"plannedOpeningDate":"25 April 2017",
"plannedOpeningDateLong":1493078400000,
"publicationDate":"25 April 2017",
"publicationDateLong":1493078400000,
"mainSpecificProgrammeLevelCode":"EU.2.",
"mainSpecificProgrammeLevelDesc":"Industrial Leadership",
"deadlineDates":["07 September 2017"],
"deadlineDatesLong":[1504803600000],
"identifier":"ALTFI-01-2017",
"title":"Improving access by innovative SMEs to alternative forms of
   finance",
"tags":[
  "convertible loans",
  "trade finanae",
  "crowdfunding",
  "business angels",
  "equity finance",
  "venture debt",
  "factoring",
  "alternative finance"
```

[1] http://ec.europa.eu/research/participants/portal/data/call/h2020/h2020\char`_topics.json

```
],
"flags":["OpenInnovation"],
"actions":[
{
  "topicId":1269891,
  "types":["CSA Coordination and support action"],
  "statusCcm2Id":0,
  "callType":0,
  "plannedOpeningDate":"25 April 2017",
  "deadlineModel":"single-stage",
  "deadlineDates":["07 September 2017"]
}
],
"callIdentifier":"H2020-ALTFI-2017",
"callTitle":"Improving access by innovative SMEs to alternative forms of
    finance"
}
```

Listing 2: Funding topic document

```
{
'startDate': '2017-10-27',
'fulltext':'...',
'endDate': '2019-08-27',
'language': 'en',
'title': 'ERA-NET Co-Fund Enhanced cooperation in Digitalisation of Energy
    Systems and Networks',
'abstract': 'Research Participant Portal is your entry point for electronic
     administration of EU-funded research and innovation projects',
'sourceURLs': ['http://ec.europa.eu/research/participants/portal/desktop/en
    /opportunities/h2020/topics/lc-sc3-es-9-2019.html'],
'docType': 'website/funding',
'source': 'SocialMediaWeb',
'fundingSpecific_metadata': {
  'EU_pillar': 'Societal Challenges',
  'callDeadline': '2019-08-27',
  'external_identifier': 'LC-SC3-ES-9-2019',
  'category': {'label': 'EU.3.'}
},
'searchDomain': ['funding']
}
```

3.1.2 Duplicate filtering

The MOVING crawlers utilise two different duplicate filtering mechanisms. The first one, briefly mentioned in the end of section 3.1.1, looks for the URL of the document that is candidate for indexed in the database. If the URL does not exist the indexing of the document will proceed.

The second mechanism is more complicated and takes into account the full-texts of the documents. Near-duplicate web documents are abundant. Two such documents differ from each other in a very small portion that displays advertisements, for example. Such differences are irrelevant for web search. So the quality of the web crawler increases if it can assess whether a newly crawled web page is a near-duplicate of a previously crawled web page or not. Duplicates often have the same base URL and different query strings. For example in *http://example.com/over/there?name=ferret*, *http://example.com/over/there* is the base URL, and *?name=ferret* is the query string. In order to apply the method, we store two URLs for each web page indexed in the platform: the full URL, and the URL with the query string stripped. The reason for this is that we can easily retrieve web pages with similar URLs, which are likely duplicates (from now we call "similar URLs" two or more URLs with the same base URL). When there is a new crawled web page to be indexed, a query to retrieve all documents with similar URL is issued. Then, each of those documents is compared with the one to be indexed. First, the title is compared. If the title is different the documents are considered different. If the title is the same, we proceed with the full-text comparison. Before the comparison we strip the full-texts of all HTML and JavaScript elements. Then, we compare the two clean full-texts using the "gestalt pattern matching" approach [2] and compare the similarity ratio with a fixed threshold we have set, after testing the method with a number of documents.

3.1.3 Indexing temporal fragmentation metadata for crawled videos

In the previous deliverable D3.2 (Vagliano et al., 2018), we described the pipeline to extract visual concepts from crawled videos and add them as metadata to the indexed document. We extend the previous process by adding temporal fragmentation metadata to the document. The temporal fragmentation is performed by the Video Analysis Service (VIA). The VIA as well as the fragmentation and concept detection methods are described in detail in the previous deliverables D3.1 (Blume et al., 2017) and D3.2 (Vagliano et al., 2018). There are concepts detected for each fragment and for the whole video as well. All the metadata is inserted in a specific common data model field `videoSpecific_metadata`, which is explained in detail in Section 2

3.1.4 Adaptive crawling

Optimizing the crawling frequency is a complex task. Websites that change their content more often need a more frequent crawling strategy. For MOVING we have implemented a simple "adaptive crawling" mechanism and have applied it to the Focused web-Domain Crawler (FDC). The mechanism takes into account not only the amount of changed content in each domain crawl but also the users preferences in the form of log data from the platform.

When the crawler re-crawls a domain, it keeps track of the percentage of pages with changed content to the total number of pages. From this percentage we calculate a *"Modification Factor"*. This metric quantifies the amount of changed content in a domain. The higher it gets, the more often it needs to be crawled as the content changes regularly. We also employ MOVING platform's wevQuery to retrieve logs about user clicks on the search results. From the number of clicks we calculate the *"Use Factor"*. This metric shows the extent to which each domain's results are used by the platforms users. Domains that are more regularly queried should be updated on a more regular basis. From the two factors, we calculate a third *"Crawl Frequency Factor"* which we later normalise. The Frequency Factor determines how often to crawl a domain. The higher the Frequency Factor of a domain is, the sooner it gets recrawled. Typicaly recrawl periods vary from 20 days to two months.

3.2 Incremental schema-level index for distributed Linked Data retrieval

Semi-structured, schema-free data formats are used in many applications because their flexibility enables simple data exchange. Especially graph data formats like Resource Description Framework (RDF)[3] have become well established in the Web of Data[4]. The latter is a global dataspace of interlinked data where providers can publicly share their content, which is based on Linked Data[5], a set of best practices to publish structured data (including but not limited to RDF) on the Web. For the Web of Data, it is known that data instances are not only added, changed, and removed regularly, but that their schemas are also subject to enormous changes over time since there is no central authority that controls which data is published and how. Unfortunately, the collection, indexing, and analysis of the evolution of data schemas on the web is still in its infancy. To enable a detailed analysis of the evolution of Linked Data, we lay the foundation for the implementation of incremental schema-level indices for the Web of Data. Unlike existing schema-level indexes, incremental schema-level indices have an efficient update mechanism to avoid costly re-computations of the entire index. This enables us to monitor changes to data instances at schema-level, trace changes, and ultimately provide an always up-to-date schema level index for the Web of Data. In this chapter, we analyse in detail the challenges of updating arbitrary schema-level indices for the Web of Data. For this purpose, we extend our previously developed model FLuID, presented in deliverable D3.2 (Vagliano et al., 2018), to define any schema-level index. In addition, we outline an algorithm for performing the updates. Finally, we describe how we extended our implementation to continuously harvest bibliographic metadata from the Web of Data.

[2] https://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/DDJ/1988/8807/8807c/8807c.htm

[3] https://www.w3.org/standards/techs/rdf/

[4] https://www.w3.org/2013/data/

[5] http://linkeddata.org/

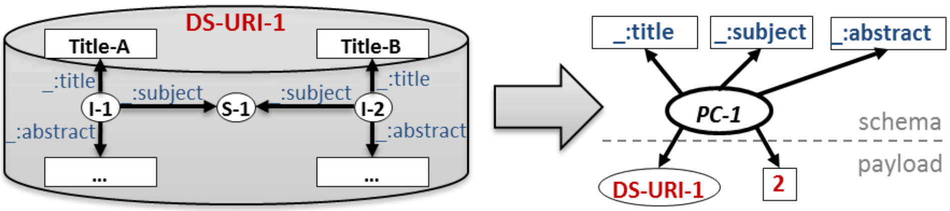


Figure 2: Two instances (I-1,I-2) using the same three properties can be summarised by the same Property Cluster (PC-1). PC-1 provides schema information (properties) and payload information, e. g., their data source URI and number of summarised instances.

3.2.1 Problem statement

The Web of Data is a valuable source for semi-structured, distributed graph data. Since the Web of Data is by its nature a decentralised database, without a central authority responsible for data management, there are different types of data schemas. In addition, data instances on the Web are not just regularly added, modified, and removed (Käfer, Abdelrahman, Umbrich, O'Byrne, & Hogan, 2013), but their schema is also subject to enormous changes over time (Dividino, Gottron, Scherp, & Gröner, 2014; Abdel-Qader, Scherp, & Vagliano, 2018). Consequently, data-driven applications that aim to make use of the Web of Data is useless if they rely on outdated data (Dividino, Gottron, & Scherp, 2015). The analysis of the evolution of data and data schemata can bring enormous advantages for the understanding of the data (Nishioka & Scherp, 2015) and help to keep local copies of the data up-to-date (Dividino et al., 2015). Many research works analyse the (co-)evolution of data (Hu, Cao, & Ke, 2014; Ohsaka, Akiba, Yoshida, & Kawarabayashi, 2016; Fan, Zhang, Wu, & Tan, 2016), but few analyse the (co-)evolution of Linked Data to understand dependencies of change behaviour such as frequently used vocabularies, network links and other (co-)evolution patterns of data instances (Dividino et al., 2014; Nishioka & Scherp, 2015). In order to perform large scale analyses on the Web of Data, efficient access to the dynamic data is required, with incremental indices playing a decisive role. The data instances from the Web must be continuously captured. This stream of data can only be indexed with an incremental algorithm, which allows efficient updates instead of costly re-computations of the entire index.

So far, there exist only incremental instance-level indices but no incremental schema-level indices. Instance-level indices allow to quickly retrieve nodes or answer questions about reachability, distance, or shortest path (Sakr & Al-Naymat, 2010). In contrast, schema level indices (SLI) index the schema computed from the data (Gómez, Etcheverry, Marotta, & Consens, 2018; Blume & Scherp, 2018a). For example, if there are instances in a data source, each described with three properties (see Fig. 2), an SLI can summarise these instances. An SLI stores only the combination of the three properties (schema information) and links them to the corresponding data source (payload information). The payload is needed to use an SLI in a concrete application context such as for data search (Gottron, Scherp, Krayer, & Peters, 2013), where the aim is to find data sources relevant to a user's query. There are numerous different variations of SLIs that index different schema information and store different payload information (Gómez et al., 2018; Blume & Scherp, 2018a).

In the related field of schema discovery in NoSQL databases, there are incremental algorithms (Baazizi, Ben Lahmar, Colazzo, Ghelli, & Sartiani, 2017; Wang et al., 2015). However, these are limited to document-oriented formats such as JSON and are designed for closed database systems. Therefore, they cannot be applied to an incremental schema-level index for open, distributed graph databases. When conducting large scale analyses on the Web of Data, a delay in the computation until the entire Web of Data is crawled is unacceptable, which is why the index computation must be able to process incomplete data. Let us assume that a data source DS-URI-2 is crawled at a certain point in time after data source DS-URI-1 has been indexed (Fig. 2). If the data source DS-URI-2 contains instances with the same schema as the instances in DS-URI-1, the payload of PC-1 must be updated. The crawler can also visit the data source DS-URI-1 again and provide changed information. This can trigger different types of updates. For example, if only one of two instances in Fig. 2 disappears, we have to update the payload with the new number of summarised instances (1 Instance). If both instances disappear we must also remove the data source and consider removing the schema element (PC-1) from the SLI since a matching query would not return any payload anyway. If, however, only the instance information changes, e.g. the title of I-2 changes to "Title-C", no update is required.

On the basis of this scenario, we can make certain assumptions about our data graph. First, we consider the crawling strategy as a black box. This means that all data sources in the Web of Data are visited (again) by the crawler at a certain point in time. Second, crawlers download the data source for a given data source URI entirely. Thus, we can assume that all statements fetched via one data source URI come next to each other in the data stream. Third, each data source can contain statements about instances stored in other data sources. Our experiments suggest that about 28% of the data instances are split over 2 or more data sources. Thus, an instance can be defined in a truly distributed setting.

Various update operators for schema-level indices are possible (Blume & Scherp, 2018b). To handle them, we rely on the FLuID model to describe SLIs in general (Blume & Scherp, 2018a; Vagliano et al., 2018), and we extend this model to describe and implement any SLI using only 11 building blocks (Blume & Scherp, 2018a; Vagliano et al., 2018).

3.2.2 Related Work

The approaches for indexing graph data can be divided in instance-level indices and schema-level indices. Instance-level indices store information about the actual data instances and their statements (Hose, Schenkel, Theobald, & Weikum, 2011). They support queries for specific resources such as finding all persons with the surname "Müller". Schema-level indices provide a concise summary of the data instances by memorizing the schema defined by the instances' statements. Schema-level indices satisfy information needs like "Find all data sources with information about persons who are actors and American presidents" (Gómez et al., 2018; Blume & Scherp, 2018a). There exists a plethora of schema-level indices (SLIs) (Schaible, Gottron, & Scherp, 2016; Gómez et al., 2018; Blume & Scherp, 2018a), which main difference is how they compute the schema. For example, 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. However, there exists no incremental schema-level index. Therefore, we discuss existing incremental instance-level indices and discuss incremental schema discovery algorithms in NoSQL databases.

There are few incremental instance-level indices that allow adding new data instances on the fly and seamlessly incorporating it into the index (Yuan, Mitra, Yu, & Giles, 2012, 2015). Yuan et. al. (Yuan et al., 2012) propose a sub graph mining algorithm to mine frequent and discriminative features in sub graphs in order to optimise a computed index and regroup sub graphs based on newly added features. Thereby the algorithm minimises the number of index lookups for a given type of query. The idea of sub graph mining for instance-level indices has been further improved by various works (Yuan et al., 2015; Kansal & Spezzano, 2017). Some works investigate the efficient sub graph matching problem on unlabelled and undirected graphs (Qiao, Zhang, & Cheng, 2017). Qiao et al. (Qiao et al., 2017) propose to compress the data graph G using a pattern graph P by extracting isomorphic sub graphs. They propose interesting techniques and address the problem of graphs too large for the main memory, but it needs to be evaluated whether they can also be applied on labelled directed graphs such as the Web of Data.

Wang et al. (Wang et al., 2015) propose a schema management framework for JSON document stores. They propose to store the schema in tree-like structures, which can be computed using each instance only once. However, their approach has several limitations. First, they assume that instances are always complete when the schema is computed. Second, their approach is only applicable for schema structures that consider properties used by the specific instance only and no information beyond the scope the instance is needed. Baazizi et al. (Baazizi et al., 2017) compute schemas for JSON documents to present information about optional fields and mandatory fields. However, incrementally discovering hidden schemas is not the same as computing exact schemas given by the data, they are limited to document orientated formats like JSON, and are designed for closed database systems.

In summary, all related approaches presented assume that the data is accessible at all time and that single instances are stored centralised and complete. To the best of our knowledge, the only approach handling decentralised graph data is proposed by Konrath et al. (Konrath, Gottron, Staab, & Scherp, 2012). They use a stream-based approach to compute the schema on-the-fly, which, however, does not support incremental updates.

3.2.3 Method description

3.2.3.1 Basic building blocks of Schema-level Indices

In our previous work, we developed the FLuID model, which is able to define arbitrary schema-level indices as combination of equivalence relations (Blume & Scherp, 2018a). A more extensive discussion of FLuID can be found in deliverable D3.2 (Vagliano et al., 2018). Formally, a schema-level index is a 3-tuple $(G, \mathrm{EQR}, \mathrm{PAY})$, 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. These equivalence relations can be defined using parameterised simple and complex schema elements (Blume & Scherp, 2018a; Vagliano et al., 2018). They basically define, how the schema is computed from the data graph, e. g., which types and properties are taken into account. In the following, we introduce common notions, define the data graph, and highlight the relevant aspects of FLuID's schema elements, their parametrisations, and the payload elements (Blume & Scherp, 2018a; Vagliano et al., 2018).

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. 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 linking instances in the data graph. We say the instance I_s with the subject URI s is of type c if there exists a triple $(s, p_t, c) \in G$, with $p_t \in P_{type}$. In the context of RDF, P_{type} contains only *rdf:type* and P_{rel} all $p \in P \setminus P_{type}$. Furthermore, we denote with $\delta^{+}(I_s)$ the out-degree of an instance I_s and with $\delta^{-}(I_s)$ the in degree.

3.2.3.2 Index Definition with FLuID

Below, we briefly describe FLuID's building blocks and subsequently describe, how FLuID-indices can be computed.

Simple Schema Elements summarise instances $I_1 \in G$ and $I_2 \in G$ by comparing all triples $(s_1, p_1, o_1) \in I_1$ and all triples $(s_2, p_2, o_2) \in I_2$. For simple schema elements, we distinguish property cluster (PC), object cluster (OC), and property-object cluster (POC). Each simple schema element compares triples following a different strategy, i. e., comparing only the properties, only the objects, or the combination of both. Furthermore, there are undirected versions of the three simple schema element, where additionally the incoming triples $(x, p, i) \in G$ with i as the subject of the instance in object position are considered.

Complex Schema Elements (CSE) partition the data graph by summarizing instances based on the three given equivalence relations $\sim^s$, $\sim^p$, and $\sim^o$. Therefore, they can be defined as 3-tuple $CSE := (\sim^s, \sim^p, \sim^o)$. While the simple schema elements summarise instances by comparing triples using the identity equivalence "$=$", the complex schema elements compare triples using arbitrary equivalences $\sim^s$, $\sim^p$, and $\sim^o$, e. g., simple schema elements. Thus, they can be considered as containers to combine any number of simple schema elements.

Parametrisations further specify our schema elements. There are four parametrisations defined in FLuID. Chaining parametrisation determines the size of the considered sub graph of a complex schema element CSE of up to k-hops, and is denoted by CSE_k. Label parametrisation allows restricting the SLI to consider only specific properties and can, for example, be used to define type cluster OC_{type}, where the object cluster only compares objects connected over the properties in P_{type}. Inference parametrisation Φ is applied on the data graph G and enables ontology reasoning using a schema graph SG. In practice this means that a schema graph SG_{RDFS} is constructed on-the-fly (or in a pre-processing step), which stores all hierarchical dependencies between types and properties denoted by RDF-Schema (RDFS)[6] properties found in the data graph G. Instance parametrisation σ allows merging equivalent instances, e. g., instances linked with *owl:sameAs*.

The label, inference, and instance parametrisations pose further restrictions or relaxations for the comparison of triples when computing the schema elements, thus, change the schema element's definition. For example, the label parametrisation allows ignoring a certain set of properties to determine the equivalence of two instances. The chaining parametrisations, however, affects the number of neighbouring instances that also need to be equivalent according to the complex schema element. Two instances I_1 and I_2 are considered equivalent by CSE_k, if for each neighbouring instances of I_1 in the data graph for each distance up to k, there is a neighbouring instance of I_2 in the same distance that is considered equivalent by CSE. Formally, this chaining of complex schema elements up to a length of k can be recursively defined as bisimulation (Blume & Scherp, 2018a; Vagliano et al., 2018).

Payload Elements are attached to schema elements and contain information about the summarised instances, for example their data source or the number of summarised instances. The payload is needed to implement a concrete application, e. g., a search engine(Gottron et al., 2013). For each type of payload, a mapping functions needs to be defined. One such function is the data source mapping function ds, which maps a schema element to the data sources of all summarised instances. The function ds 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.

[6] https://www.w3.org/TR/rdf-schema/

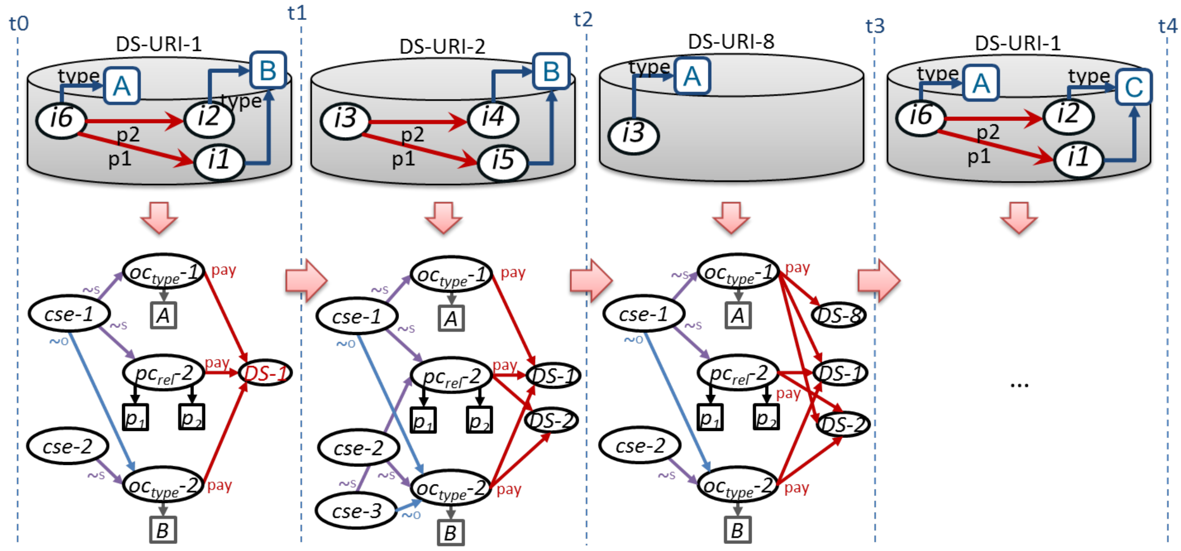


Figure 3: Computing and updating a schema-level index when data sources arrive over time with information about the data graph.

3.2.3.3 Index Construction with FLuID

In this section, we present our approach to compute schema-level indices modelled with FLuID. In the following section, we will extend this approach to an incremental indexing approach.

Computing an SLI can be described as a function $SC : (G, \mathrm{EQR}, \mathrm{PAY}) \rightarrow SLI$, which computes according to the equivalence relations given in EQR, and the n payload functions PAY the concrete schema-level index sli for a data graph G. Since the equivalence relations given in EQR can be defined with schema elements, we can treat EQR as a tuple of schema elements. In the following, we will denote schema elements in EQR as abstract schema elements. For example, the SLI TermPicker (Schaible et al., 2016) defines the schema structure that summarises instances that have a common set of types, a common set of properties, and a common set of combined types of all neighbouring instances. With the help of FLuID, this can be expressed with four abstract schema elements:

$$\mathrm{EQR}_{\mathrm{TermPicker}} := (OC_{type} \cap PC_{rel}, T, OC_{type}), \tag{1}$$

A complex schema element wraps the label parameterised object cluster using the set P_{type} denoted by OC_{type}, the label parameterised property cluster using the set P_{rel} denoted by PC_{rel}, and an arbitrary tautology denoted by T, which considers everything equivalent.

To compute a concrete schema-level index sli, the abstract schema elements are used as blueprints to compute (instantiate) schema elements for the instances in the data graph G. 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. More specifically, instantiating schema elements means creating a schema element $se\text{-}i \in sli$ with a certain set of specific attributes, e. g., the three properties {_:title, _:subject, _:abstract}.

There is a significant difference between computing simple schema elements and computing complex schema elements. The FLuID model defines simple schema elements as low-level building blocks that summarise instances by comparing triples using the identity equivalence ($=$) for properties and/or objects. Thus, simple schema elements can always be computed directly from the data without dependencies to other schema elements. In contrast, complex schema elements are deliberately designed to have dependencies either on complex schema elements or on simple schema elements. To explain this in more detail, we look at the example shown in Fig. 3. When we start with an empty SLI (interval $[t0, t1]$), to compute the schema of instances, we first need to compute the sub-schema structures (i. e., the simple schema elements $oc_{type}\text{-}1$, $oc_{type}\text{-}2$, $pc_{rel}\text{-}1$) according to the abstract schema definition (blueprint) given in Eq. (1). We can compute the complex schema elements $cse\text{-}1$ and $cse\text{-}2$ based on the simple schema elements. To compute the complex schema element for instance i6, we need to know the type information about instances i2 and i1 ($oc_{type}\text{-}2$).

The advantage of this feature is that with increasing complexity of the abstract schema definition, the computation costs increase only linearly since the index reuses already computed simple schema elements for several complex schema elements (Blume & Scherp, 2018a; Vagliano et al., 2018). In general this means, we have to instantiate at most all abstract schema elements for each instance $I_1 \in G$. We denote with s the number of abstract simple schema elements and with c the number of abstract complex schema elements. Please note that the chaining parametrisation is defined so that the pattern of an abstract complex schema element is applied recursively k times. When chaining a CSE k times, comparable to unrolling a loop, we can simply assume that we have to calculate $k \times c$ many abstract complex schema elements. However, no new abstract simple schema element is required, so the number s is not affected.

The payload information needs to be computed for a data graph G according to the payload functions PAY. Computing the payload elements follows the same principle as computing the schema elements. Thus, we omit the details here. Please note, in the following, we consider a computed SLI for a data graph G again as a graph. However, other data structures can be used as well.

To summarise, the total number of instances in G can be bounded by the number of nodes in the data graph G ($|G|$), with $|G| = v$. When computing an SLI, we have to compute $\alpha \leq s \times v$ many simple schema elements and $\beta \leq c \times v$ many complex schema elements. Furthermore, we need to consider p many payload mapping functions instantiating $\gamma \leq p \times v$ many payload elements for the data graph G. Thus, we know that $\alpha + \beta \leq v \times (s + c \times k)$ schema elements are instantiated in the final SLI. With a data graph where no two instances can be summarised, we end up with $\alpha + \beta = v \times (s + c \times k)$.

3.2.3.4 Update Operations on an Index built with FLuID

As described in the previous sections, computing an SLI means extracting schema and payload information from the data to compute schema and payload elements that are added to the SLI. In the following, we analyse the different cases of updates with respect to the expected complexity. We compare the costs implied with our incremental index to computing the index from scratch, as it is done so far (Konrath et al., 2012; Blume & Scherp, 2018a; Vagliano et al., 2018).

Updating Schema Elements There are six cases of updates possible for an SLI: a new instance is observed with a new schema (SE_{new}), a new instance is observed with a known schema (PE_{add}), a known instance is observed with a changed schema (SE_{mod}), a known instance is observed with only changed instance information (PE_{mod}), a instance with its schema and payload information no longer exists (PE_{del}), and no more instance with a specific schema exists in the data graph (SE_{del}). Please note, we denote with the schema of an instance the complete schema defined by all abstract schema elements in the index definition, e. g., the complex schema element. Sub-schemas are smaller parts of the schema, e. g., simple schema elements combined within a complex schema element.

Adding an instance to G means adding a set of triples (s_1, p_1, s_2) about a resource s_1 when there where no triple about this resource in G before, neither with s_1 as subject nor with s_1 as object. Since we do not force an order on the data, as soon as any triple about a resource is observed we have to assume it exists and treat it as an instance that may has no schema besides being a resource.

For new instances in G, we consider two cases. First, if the instance is observed with an entirely new schema (SE_{new}), all $s + c \times k$ (sub-) schema elements need to be computed and added to the index. However, it is possible that some sub-schema elements are already known even if the schema of the instance is new. For example, the specific combination of properties (pc-1) is known but not the specific combination of types (tc-8) and therefore also not the complex combination cse-3 using pc-1 and tc-8. This means, adding an instance with a new schema to G (SE_{new}) can require adding between 1 and $s + c \times k$ schema elements to the index. Second, if the instance is observed with a known schema (PE_{add}), also all sub-schema elements have to already exist in the index. Therefore, only the payload of existing schema elements may needs to be updated.

Modifying instances means adding and/or removing a set of triples $(s_1, p_1, s_2) \in G$ about the resource s_1 when there existed triples about this resource in G before. When the modifications are only on the instance-level (PE_{mod}), this requires no change on the schema elements but the payload may needs to be updated. However, the modifications can be on the schema-level (SE_{mod}). This means, that, for example, new properties are added or types are changed. When we modify an instance on the schema-level, analogously to adding an instance, between 1 and $s + c \times k$ schema-elements in the index may need to be updated or created. Furthermore, it is possible that for another instance I_2 a triple $(s_2, p_1, s_1) \in G$ was observed before, where s_1, s_2 are the subject URIs of I_1, I_2 respectively. This means, for each such instance I_2 in G, we may need to update the schema elements as well. Thus, for each incoming edge to I_1 in G, up to c many complex schema elements require an update. When we apply the chaining parametrisation here, the individual in-degree of each neighbour up to a distance k needs to be considered. To ease notation, we denote with $\delta^-(I_1^{max_k})$ the maximum in-degree of all instances within a k-hop distance from instance I_1. We can then simplify the estimation to $\delta^-(I_1^{max_k})^k \times c$

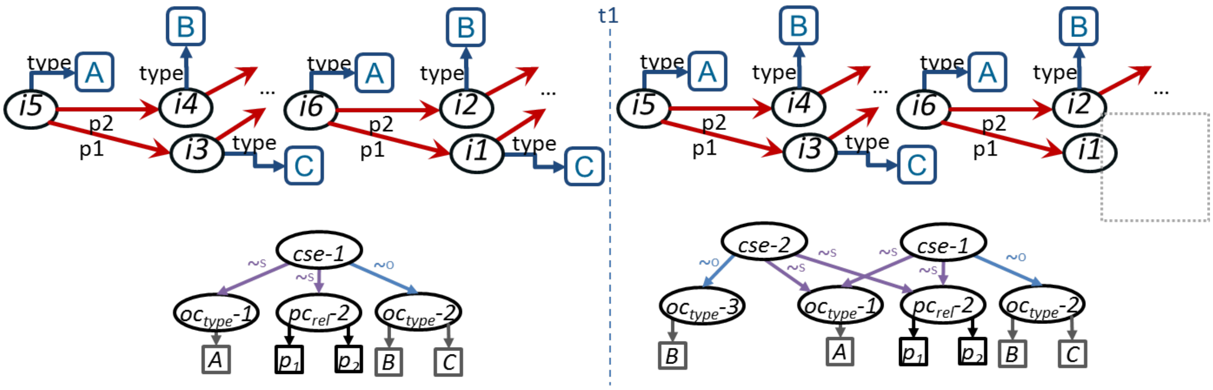


Figure 4: Deletion of the information about instance $i1$ effects the complex schema of the instance $i6$, which has a property linking to $i1$.

effected schema elements. Thus, SE_{mod} requires a maximum of $s+c\times k+\delta^{-}(I_1^{max_k})^k\times c$ updates on schema-level. Since s and c are fixed before computing the index, the only variable factor depending on the data is the in-degree δ^{-} of the instances. Please note, for existing SLIs we know that $c\leq 1$ and $s\leq 3$ (Blume & Scherp, 2018a; Vagliano et al., 2018). Furthermore, our results discussed in Section 3.2.4 indicate that the in-degree is on average only 5.

Deletion can be seen as an extreme case of modification where all instance information is deleted, i. e., all triples about the resource. Due to the decentrality of the Web of Data, there can still exist links to that instance in the data graph. Thus, the resource identifier may still appear in the data.

First, the deletion can mean that there are no further data instances in the data graph G with that schema (SE_{del}). This means, we basically have to revert all changes that where needed to add a new instance with a new schema (SE_{new}) and we have to delete between 1 and $s+c\times k$ schema elements from the index. Furthermore, deleting single schema elements may require an update of depending schema-elements when (chained) complex schema elements are used analogously to the modification SE_{mod}. Considering Fig. 3, the schema is computed based on types extracted from instances linked via properties. This means, when we remove instance i4 and i5 after $t4$, updates on depending schema elements are necessary. We delete the (sub-) schema elements of cse-2 and thus, also have to update the schema element cse-1 since it can no longer reference oc_{type}-2. Thus, analogously to the modification, if we have to consider the in-degrees of neighbouring instances and possibly update up to $\delta^{-}(I_1^{max_k})^k\times c$ complex schema elements.

Second, the deletion can mean that there are still instances in the data graph G with that schema left (PE_{del}). This means, also all sub-schema elements have to remain in the index. Thus, there is no deletion of schema information required and only the payload of existing schema elements may needs to be updated. For this case, although no schema element is deleted, we may still have to update $\delta^{-}(I_1^{max_k})^k\times c$ complex schema elements. Let us consider the example illustrated in Fig. 4. When we compute indices using (chained) complex schema elements, e. g., TermPicker, we do not delete even a sub-schema element but we rather need to add schema elements cse-2 and oc_{type}-3. Thus, up to $\delta^{-}(I_1^{max_k})^k\times c+s$ schema elements need an update.

3.2.3.5 Updating Payload Elements

With p many payload functions, up to p new payload-elements need to be instantiated for each new instance I_1. Making general assumptions about the payload is not possible since there can be a unlimited number of different payload functions. Thus, we focus on two often used payload functions, the data source function and the instance count function. Adding or deleting an instance always triggers an update on the instance count payload. Modifying instance does not trigger a change on the instance counts. All these three kinds of instance-level change can trigger an update of the data source payload if a new data source is added or an existing instance is moved to another data source.

3.2.4 Discussion

As we discussed in the previous sections, to update schema-level indices some information about the data graph needs to be preserved to coordinate the right update operation to the correct schema elements. Thus, we propose to store two kinds of link information, instance URI to schema element and instance URI to data source URI. Furthermore, if complex schema elements are used, we need to memorise a subset of incoming

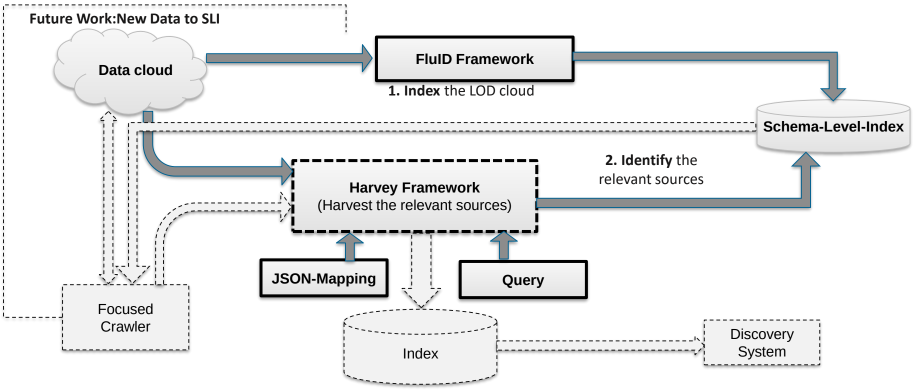


Figure 5: High-level view of the Data Integration Service.

edges for each instance. The amount of storage space required to store all these links for the Web of Data may be undesirably large. We plan to conduct detailed analyses using real-world datasets crawled from the Web of Data. From our analysis in Section 3.2.3.4, one major factor is the in-degree of instances in the data graph. Our preliminary results for the first four snapshots crawled by the Dynamic Linked Data Observatory (DyLDO)[7] indicate large variety in the link distribution and instance distribution. About 28% of the data instances are split over 2 to 5800 (average 2.2) data sources. Furthermore, about $\frac{1}{3}$ of the instances have dependencies on 1 to 172.000 instances. However, on average instances only have dependencies on 5 instances. Storing all links sums up to about 33% of the number of links in the data graph. These first results suggest that implementing an efficient incremental schema-level index using a complex schema element for the Web requires storing significantly more data permanently. Thus, we will also analyse possible optimisation and approximation strategies.

3.2.5 Implementation, APIs and integration

We implemented the incremental index algorithm in our FLuID Framework[8]. The FLuID Framework is being integrated in LODatio+, which is used by the Data Integration Service. This enables us, as illustrated in Fig. 5, step 1, to continuously index the Web of Data (Data cloud). We use this index to identify the currently relevant data sources (step 2). These identified data sources are used as seedlist for our focused crawler that crawls and harvest the bibliographic metadata. We parse and transform the bibliographic metadata using the provided JSON-Mappings file, as described in deliverable D3.1 (Blume et al., 2017). A more extended description of the harvesting, processing, and integration of bibliographic metadata will be available in the upcoming deliverable D4.3.

3.3 Duplicate detection and removal

3.3.1 Problem Statement

The MOVING platform contains a large and continuously expanding document collection from many different sources. In such a setting, *finding* and *eliminating* duplicate records is important to avoid information overload by users. The identification of candidate duplicates, i.e. documents that are completely identical or exhibit only small differences with respect to their metadata entries, has been a major topic in deliverable D3.2 (Vagliano et al., 2018), Section 3.4. We developed a deduplication method for identifying duplicates based on the similarity of metadata entries with a low computational complexity which has been successfully evaluated against a gold standard of annotated duplicates taken from the DFG Project POLLUX[9]. In this section, we describe novel work which is devoted to the ranking of duplicates with the aim to be able to have a criterion for eliminating entries that are incomplete or outdated, while the best-ranked document are chosen as the primary record to be kept in the index.

[7] http://km.aifb.kit.edu/projects/dyldo/
[8] https://github.com/t-blume/fluid-framework
[9] http://www.pollux-fid.de/

3.3.2 Method description

The main heuristics we applied for ranking duplicates within a cluster of identical documents have been: a) the relevance of the source, b) the relevance of the document type, and c) the timestamp of the document. In the following, we provide additional information for each heuristic.

- **Relevance of the source repository**. We give priority to quality-controlled MOVING repositories with rich metadata, in particular ZBW Economics (EconBiz), GESIS fulltext (SSOAR) and GESIS metadata (SOLIS). Here, we give ZBW Economics a higher rank than SSOAR according to the ranking of Open Access repositories[10]. Specifically, we give to the SSOAR rank 2 and SOLIS rank 3. As regards the Web sources ('Social Media and the Web', 'Web of Data') we give 'Web of Data' a higher rank, since this source provides more structured records than 'Social Media and the Web'. This results in the following ranking of MOVING data sources: 1. ZBW Economics, 2. GESIS fulltext, 3. GESIS metadata, 4. Web of Data, 5. Social Media and the Web. VideoLectures.NET is a non-competitive source, since it provides a quite different type of information (videos) than the other sources. Thus, results from VideoLectures.NET will be kept in any case (apart from deduplication within a set of documents from VideoLectures.NET). The same applies for the MOVING source 'Laws and Regulations'.
- **Relevance of the document type**. Full paper documents are favored over records for which only an abstract exist. As regards video lectures, those videos are given a higher rank which represent scientific full paper presentation (such as keynote, best paper, invited talk, thesis defense), in contrast to videos which present information about events (such as opening, introduction, summary) or promotional material.
- **Timestamp of the document**. Up-to-date records are given preference.

3.3.3 Implementation, APIs and integration

All in all, 51,899 duplicate clusters have been discovered, with a total amount of 110,082 documents, among them 24,303 identical elements. As expected, the distribution follows a typical power-law, i.e. there is a huge number of clusters containing just two elements (see Figure 6). It is important to note that all clusters are disjoint so that our technique only requires comparing each object with every other object in the cluster. For the non-identical elements, the heuristics has been applied whenever Source, Document Type or Timestamp values differed in at least one value, indicating that nearly the half of the ranking problem could be solved by a repository ranking. Lower-ranked duplicates have been removed from the search index of the platform to avoid users being confused by duplicates. However, the duplicates and their ranking values have been inserted into an additional index (non-searchable) to avoid losing information.

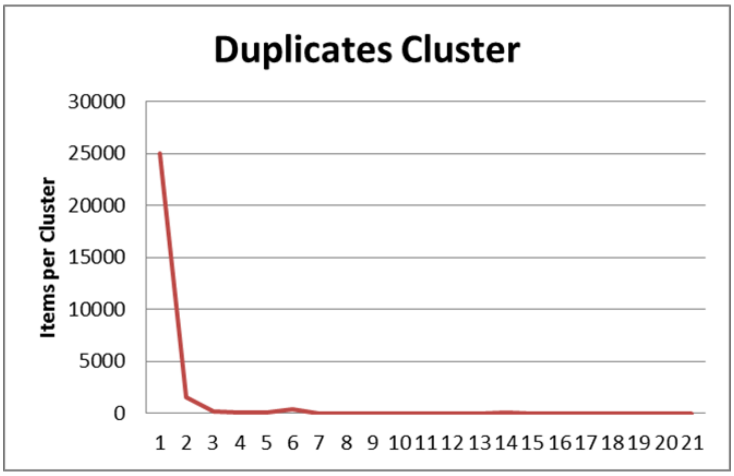


Figure 6: Statistics on Duplicates

[10] http://repositoryranking.org/

3.4 Entity extraction and linking

In this section, we describe the work on entity extraction and linking. As described in the deliverable D1.1 (Bienia et al., 2017), the MOVING platform targets two kind of users, young researchers and auditors, supporting two different use cases.

During the requirement analysis described in D1.1 (Bienia et al., 2017), very different needs in terms of entity of interest emerged for auditors and researchers. For example, the former are rather interested in organisations, locations, persons, while the latter, although depending on the research area, tend to focus more on research methods, tools, measurements, etc. In addition to these different needs, also the documents to which apply entity extraction and linking are not the same. While auditors can benefit more from the Economics documents, Web pages crawled from organisations' Web site, and laws which are available in the platform, researches primary refer to publications or other educational material, as well as funding. Not only these various types of documents contains distinct types of entities, but also the extraction and linking procedures varies in the implementation as well as in when needs to be run because of the different data addressed. For example, in the case of crawled data it is convenient that this extraction occur in an online fashion, while the crawlers integrate the new documents through the DIS. In fact the platform continuously crawls new data from the Web. For publications or educational resources, which are typically integrated offline by their data providers in bulks, it is better to first index the data then extract and link the entities. In this way, although integrating many new data altogether, it is guaranteed that the platform is still responsive to users because separating the two tasks significantly reduces the computational resources required. Despite the separation of the two tasks, the entity extraction and linking module is implemented in such a way that can still be executed incrementally (processing only the new documents).

For these reasons, we implemented two different modules for entity extraction and linking, one for the researchers' use case and one for the auditors. It is worthy to note that the two modules can also be combined for documents which have both entities interesting for auditors (persons, locations, organisations) and researchers (methods, tools, measurements). Section 3.4.1 describes the problem of entity extraction and linking for the researchers' use case, outlines the method provided in the platform to solve such problem, and provides the evaluation of that method. Instead, Section 3.4.2 addresses the same problem for the auditors' use case, explaining how it has been solved in the MOVING platform. Further implementation details on the two modules will be given in the upcoming deliverable D4.3.

3.4.1 Young researchers' use case

3.4.1.1 Problem Statement

Entity Extraction and Linking is a subtopic of *Information Extraction* and aims to semantically annotate unstructured sources such as textual corpora using techniques of Machine Learning (ML), Natural Language Processing (NLP), Text Mining (TM), Information Retrieval (IR) and Semantic Web (SW). This section highlights the fundamental differences between different related tasks within the field and gives an example for each task:

1. **Named Entity Recognition (NER)**: Recognizing the boundaries of a named entity in a text.

   Input: Architect Irving Morrow constructed the Golden Gate Bridge.

   Output: Irving Morrow, Golden Gate Bridge

2. **Named Entity Classification (NERC)**: The noun phrase refers to an entity of a specific class, e.g., person, organisation, location or more-fine-grained.

   Input: Architect Irving Morrow constructed the Golden Gate Bridge.

   Output: Irving Morrow: person, Golden Gate Bridge: location

3. **Named Entity Linking/Disambiguation (NEL/NERD)**: Linking named entities to an external knowledge base such as a thesaurus or ontology.

   Input: University of California offers programming courses in Java

   Output: University of California: `http://dbpedia.org/page/University_of_California,_Berkeley`, Java: `https://dbpedia.org/page/Java_(programming_language)`[11]

[11] Note that 'University of California' and 'Java' are ambiguous and can refer to various entities and concepts.

NER and NERC are well studied tasks in NLP and key components in a wide range of natural language understanding tasks(Tjong Kim Sang & De Meulder, 2003; Pradhan et al., 2013). Many high-level applications such as entity linking can be built on top of a NERC system (Moro, Raganato, & Navigli, 2014). For the NEL task, an external knowledge base with definitions and descriptions needs to be provided (Ji, Nothman, Hachey, et al., n.d.). Moreover, all tasks require annotated corpora (including annotation guidelines) for training, development and testing the software. NEL output can be used for knowledge graph construction and visualisation (Hachey, Radford, & Curran, 2011).

Note that recent work on Named Entity Recognition and Linking goes beyond proper nouns as rigid designators (Kripke, 1972) and also includes concepts as long as they have a 'unique identification' and may be linked to a knowledge base or ontology (Marrero, Urbano, Sánchez-Cuadrado, Morato, & Gómez-Berbís, 2013).

Motivation The basic motivation of our work is to support researchers in their information seeking work, in particular by enhancing the research data's metadata. A major shortcoming of research data sets is that the metadata is generally not rich enough to describe the contents of the dataset properly[12]. Thus, detecting mentions of data sets in publication text along with associated *research methods, tools, measurements, etc.* would provide additional context information on how the dataset has been used and helps to improve search and retrieval of research data. This also applies when searching scholarly literature in digital repositories. Valuable information that might be beneficial for researches often is not contained in the metadata. In fact, there is a growing trend to systematically analyze and interlink all contents of publications and data repositories in order to 'produce a comprehensive graph capturing all elements related to scholarly communication' (e.g., *authors, publications, data, and projects*). This approach is promoted, e.g., by the OpenAIRE e-Infrastructure[13].

In a recent survey, Nasar et al. (2018) gives an overview of automatic information extraction methods from scientific articles that try to find the key entities from unstructured text and metadata. Extracting such information automatically, or annotating the key phrases or sentences in the text, would aid in a variety of applications, including automated literature review and trend analysis (Gupta & Manning, 2011; Augenstein, Das, Riedel, Vikraman, & McCallum, 2017).

Social Science Use Case The importance of extracting core metadata information from social science publications has been recently stressed by Eckle-Kohler (2013) who aim to automatically filter search results by a faceted search on the *research methods* field. They carry out experiments on SOLIS (Social Science Literature Information System)[14], using a fixed method list of 40 descriptors[15] (See Section 3.4.1.2).

Regarding the retrieval of research data in social science, e.g. survey datasets, publishers like SAGE Publications, Digital Science, and Project Jupyter have only recently started to build up a research infrastructure which aims to provide rich contextual metadata. They rely on machine learning algorithms for detecting research-related entities which are used either in publications or fields studies by researchers who use the datasets[16].

Inspired by their work, we define a list of basic entity types relevant for the social sciences in Table 3. Note that entity types might be related to each other. For instance, in order to realise a certain research objective, an experiment is instrumented where a specific combination of methods is applied to a data set that might be intellectual or software, thus achieving a specific performance and result in that context.
Example: P-values for *one-tail paired t-test* (method) on *Allbus* (dataset) and *ISSP* (dataset)

[12] See also https://coleridgeinitiative.org/richcontextcompetition

[13] https://www.openaire.eu/

[14] available from www.gesis.org

[15] https://www.gesis.org/angebot/recherchieren/tools-zur-recherche/methodenliste-fuer-die-datenbanken-sofis-und-solis/

[16] See also SAGE Rich Context Challenge https://coleridgeinitiative.org/richcontextcompetition

Table 3: Selected Semantic Entity Types

Entity	**Definition / *Example***
Research Method	Algorithm and approach carried out by author in the investigation. *Interaction effects* of *pre-testing*.
Research Theory	Theory used in order to address the problem. In *cross-cultural research*, results have been compared from various studies.
Research Tool	Tools used to perform the experiments. Literature review was carried out with *Google Scholar*.
Research Measurement	Evaluation measures to gauge performance. The *dependent variable* must be an *interval variable*.
Research Dataset	Datasets that are used in order to conduct experiments. Respondents of the *ISSP 2016 dataset* were accessed by age and gender.
Research Field	Domain addressed by the research study. The *health and well-being* of elderly people in Bavaria was investigated.

Formal problem definition Let E denote a set of entities. The Named Entity Recognition and Linking task consists of (i) identifying entity mentions m in a sentence and, (ii) linking them, when possible, to a reference knowledge base K (i.e, the SAGE Thesaurus[17]) and (iii) assigning a type to the entity, e.g., *research method*, selected from a set of given types. Given a textual named entity mention m along with the unstructured text in which it appears, the goal is to produce a mapping from the mention m to its referent real world entity e in K.

Challenges There are some major challenges that any named entity recognition, classification and linking system needs to handle. First, regarding NER, identifying the entities boundary is important, thus detecting the exact sequence span. Second, ambiguity errors might arise in classification. For instance,'range' might be a domain-specific term from the knowledge base or belong to the general domain vocabulary. This is a challenging task for which context information is required. In the literature, this relates to the problem of **domain adaptation** which includes fine-tuning to specific named entity classes[18]. With respect to entity linking, another challenge is detecting name variations, since entities can be referred to in many different ways. Semantically similar words, synonyms or related words, which might be lexically or syntactically different, are often not listed in the knowledge base (e.g. the lack of certain terms like 'questioning' but not 'questionnaire'). This problem of automatically detecting these relationships is generally known as **linking problem**. Note that part of this problem also results from PDF-to-text conversion which is error-prone. Dealing with incomplete knowledge bases, i.e. **handling of out of vocabulary (OOV) items** is also a major issue, since knowledge bases are often not exhaustive enough and do not cover specific terms or novel concepts from recent research. Last but not least, the combination of different semantic types gives a more coherent picture of a research article. We hypothesise that such information would be helpful and results in an insightful co-occurrence statistics, and provides additional detail that allows to resolve entity mentions jointly in a document, and finally helps to assess the **relevance of terms** by means of a score.

3.4.1.2 Related work

Dictionary-based approaches and rule-based approaches as advocated in Gate's ANNIE (Cunningham, 2002) are predominantly used in industry because of their higher interpretability and flexibility. Pros are that, e.g., domain experts can develop rules that take into account the relevance of some entities (Chiticariu, Li, & Reiss, 2013). Entity extraction using machine learning is common in academic research, where also a mixture of rule-based and machine learning techniques can be used (Florian, Ittycheriah, Jing, & Zhang, 2003).

Supervised machine learning methods for entity detection require fully annotated documents and a set of features to train models on large domain-specific corpora. The most effective machine learning approaches applied to the NER problem are conditional random field approaches (CRF), a sequential deterministic machine learning algorithm (Sutton & McCallum, 2006). The performance of CRF models rely heavily on the features, for example, orthographic, morphological, linguistic-based, conjunctions and dictionary-based.

Deep learning has also been applied to NER such as a fully connected neural network (Collobert et al., 2011) to effectively identify entities in a newswire corpus. The application of character and word embeddings in Bi-directional Long Short-Term Memory (LSTM) (Lample, Ballesteros, Subramanian, Kawakami, & Dyer,

[17] http://methods.sagepub.com/, approx. 600 research methods as key terminology provided by Sage publishing.

[18] apart from those used in traditional NER systems like *Person*, *Location*, or *Organisation* with abundant training data, as covered in the Stanford NER system(Finkel, Grenager, & Manning, 2005a)

2016; Ma & Hovy, 2016) achieved state-of-the-art performance in several sequence-to-sequence datasets, such as CoNLL03 (Tjong Kim Sang & De Meulder, 2003) for NER. Nevertheless, deep learning methods typically require a large amount of labeled data for supervised learning and take more time and computing resources to train than the classical machine learning methods. The effectiveness of neural networks can be attributed to their ability to learn effective features jointly with model parameters directly from the training dataset, instead of relying on handcrafted features developed from a specific dataset.

Interestingly, the traditional NER task conceived as a Sequence Labeling Task and based on CRFs allows for various extensions that make it state of the art for this task. For addressing the domain adaptation problem, supervised or semi-supervised variants can be used (Daumé III, 2009), depending on whether there exists no or some training data in the target domain. In supervised domain adaptation, there is a limited amount of target domain training data, but much more training data in a source domain. The semi-supervised case is similar, but instead of having a small annotated target corpus, we have a large but unannotated target corpus. The most common approaches to domain adaptation include: a) distributed word representations like Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), C&W (Collobert et al., 2011) or GloVe (Pennington, Socher, & Manning, 2014).[19], b) distributional semantic models like Brown clusters (Brown, Desouza, Mercer, Pietra, & Lai, 1992)or c) using combinations of different models like part of speech taggers (Passos, Kumar, & McCallum, 2014) or gazetteers (Florian et al., 2003). Regarding entity linking, popularity measures as well as joint inference are often used for mapping all mentions of various types in a text coherently (Hoffart et al., 2011). A summary of recent work is given in (Augenstein, Derczynski, & Bontcheva, 2017).

Named Entity Recognition and Linking with a focus on 'scientific entities' has been an evolving topic in recent years. Mesbah et al. (2018) propose a supervised learning approach that builds on distant supervision for extracting long-tail entities combined with a heuristic filtering approach. He considers conference papers drawn from 11 conferences and reports a performance of *dataset* (Precision:0.77, Recall:0.40, F1 Measure:0.53) and *method* (Precision:0.78, Recall:0.12, F1 Measure:0.21) using the Stanford NER tagger trained on a manually annotated small seed set of known 50 named entities instances and 2 entity types. Gupta et al. (2011) use the ACL anthology reference corpus (ACL ARC) (Bird et al., 2008) for extracting *focus, technique, and domain* from fulltext scholarly articles using a pattern-based bootstrapping approach, and achieve the following scores after 50 iterations: *technique* (Precision:0.30, Recall:0.47, F1 Measure:0.36) and *domain* (Precision:0.28, Recall:0.57, F1 Measure:0.37). Augenstein et al. (2017) report that for the task of identifying and classifying keyphrases, CRF-based methods with part-of-speech and orthographic features achieved a lower performance than recurrent neural networks in terms of F1 measure (Gupta & Manning, 2011), specifically CRF F1: 0.23-0.27 and RNN F1: 0.38-0.43. Apart from *process, material and task*, where the *task* label denotes a variety of entities, i.e. *application, end goal, problem, or task*.

In the experiments conducted by (Eckle-Kohler et al., 2013), the authors define a multilabel classifier to automatically assign scientific methods in Social Science papers based on abstracts and titles of publications, comparing results to the given metadata. Their main finding is that while some research methods are relatively easy to identify (i.e., with an F1 score of 0.67 for *empirical, quantitative empirical*), in general the abstract and metadata information are not sufficient for making a reliable prediction.

3.4.1.3 Method description

Our context-aware framework builds on Stanford's CoreNLP and Named Entity Recognition System[20]. The information extraction process follows the workflow depicted in Figure 7, using separate modules for preprocessing, classification, linking and term filtering, where the PDF-to-text conversion tool Cermine[21] has been applied to the original PDF files beforehand. We envision the task of finding entities in scientific publications as a sequence labeling problem, where each input word is classified as being of a dedicated semantic type or not. In order to handle entities related to our domain, we train a novel machine learning classifier with major semantic classes (see Table 3), using training material from the ACL RD-TEC 2.0 dataset (QasemiZadeh & Schumann, 2016). Apart from this, we follow a domain adaptation approach inspired by Agerri et al. (2016) and ingest semantic background knowledge extracted from external scientific corpora, in particular the ACL Anthology (Bird et al., 2008; Gildea, Kan, Madnani, Teichmann, & Villalba, 2018). We perform entity linking by means of a new gazetteer-based SAGE dictionary of Social Research Methods (Lewis-Beck, Bryman, & Liao, 2003), thus putting a special emphasis on the social sciences. The linking component addresses the synonymy problem and matches an entity despite name variations such as spelling variations. Finally, term

[19] See also (Kulkarni, Mehdad, & Chevalier, 2016; Guo, Che, Wang, & Liu, 2014; Seok, Song, Park, Kim, & Kim, 2016)
[20] https://nlp.stanford.edu/projects/project-ner.shtml
[21] https://github.com/CeON/CERMINE

filtering is carried out based on a termhood and unithood, while scoring is achieved by calculating a relevance score based on TF-IDF (see Section 3.4.1.4).

In order to conduct this study, we selected the repository for the Social Sciences SSOAR, split into train, develop and test data [22] as well as the train and test data of the SAGE Rich Context Challenge corpus [23]. Our work extends previous work in various ways. First, we do not limit our study to abstracts, but use the entire fulltext. Second, we focus on a broader range of semantic classes, i.e. *Research Method*, *Research Theory*, *Research Tool* and *Research Measurement*, tackling also the problem of identifying novel entities. We believe that these are the most important semantic types to be considered for the task[24].

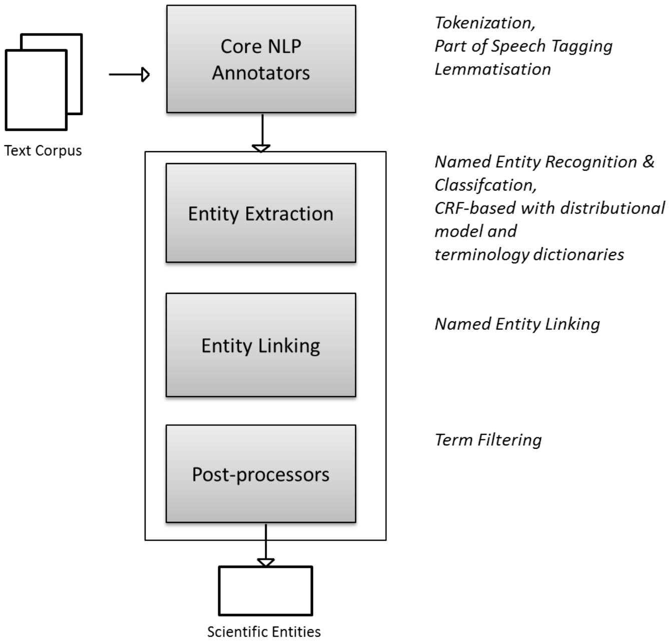


Figure 7: Overview of the entity extraction pipeline

Conditional Random Fields Linear-chain Conditional Random Fields (CRFs) are frequently used in Machine Learning for sequence labeling tasks, because as discriminative models they are constructed directly and don't make the false assumption that features need to be independent, as opposed to Hidden Markov Models. Moreover, when labeling Y_i, future observations can be taken into account, where Y is the hidden state and X is a sequence of observations.
According to Sutton et. al. (2006) we define:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right\} \tag{2}$$

Let $X = (x_1, x_2, ..., x_n)$ be a sequence of observations, and $Y = (y_1, y_2, ..., y_n)$ be a corresponding sequence of labels for each of the observations. In Equation 2, $f_k(y_t, y_{t-1}, x_t)$ is a feature function, defining a value given the current and previous labels. θ_k are the parameters for the distribution, and $Z(x)$ is a normalisation function to keep the total $p(y|x) = 1$. Both θ and Z must be computed, and consist of an exponential number of terms.

Distributed Semantic Models For domain adaptation, we integrate further background knowledge. We use vector embeddings of words trained on additional corpora and which serve as input features to the CRF model. Semantic representations of words are a successful extension of common features, resulting in higher NER performance (Turian, Ratinov, & Bengio, 2010) and can be trained offline.

The approach makes use of the proximity of distributed representations and builds on the hypothesis that word vectors belonging to the same name category, occur in close vicinity in the abstract vector space of the embedded words. Embedding features are real-valued word embeddings. Prominent word embeddings include probabilistic prediction approaches like the continuous bag of words (CBOW) model of word2vec (Mikolov et

[22] The SSOAR is available at `https://www.gesis.org/ssoar/home` with a total of 13,175 English documents
[23] `https://coleridgeinitiative.org/richcontextcompetition` with a total of 5,000 English documents
[24] We exclude *research dataset and field* which have been the focus of another study.

al., 2013) and C&W (Collobert et al., 2011) and reconstruction-based approaches like GloVe (Pennington et al., 2014).

In this work, the word vectors have been learned from the scientific ACL ARC[25] using GENSIM with the skip gram model (Mikolov et al., 2013) and a pre-clustering algorithm[26]. A summary of the size of the unlabeled English data used for training word embeddings can be found in Table 4.

Table 4: English data used for Training Word Embeddings

Corpus	Articles	Documents/Tokens
ACL Anthology Reference Corpus	22,878	806,791/2.5 GB

Features The features incorporated into the linear chain CRF are shown in the Table 5. The features depend mainly on the observations and on pairs of adjacent labels, using a log-linear combination. However, since simple token level training of CRFs leads to poor performance, more effective text features such as word shape, orthographic, gazetteer, Part-Of-Speech (POS) tags, along with word clustering (see Section 3.4.1.3) have been used.

Table 5: Features used for NER

Type	**Features**
Token unigrams	$w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, \ldots$
POS unigrams	p_i, p_{i-1}, p_{i-2}
Shapes	shape and capitalisation
NE-Tag	t_{i-1}, t_{i-2}
WordPair	(p_i, w_i, c_i)
WordTag	(w_i, c_i)
Gazetteer	SAGE gazetteer
Distributional Model	ACL Anthology model

3.4.1.4 Extension of the SAGE Thesaurus

We use the SAGE thesaurus[27] as a well-established knowledge resource in the Social Science domain. It includes well-defined concepts, an explicit taxonomic hierarchy between concepts as well as labels that specify synonyms of the same concept (see Figure 8 for an example). A portion of terms is unique to the social science domain (e. g., 'dependent interviewing'), while others are drawn from related disciplines such as statistics (e. g., 'conditional likelihood ratio test')[28]. However, since the thesaurus is not exhaustive and covers only the 600 top-level concepts related to social science methods, our aim was to extend it by automatically extracting further terms from domain-specific texts, in particular from the Social Science Open Access Repository. More concretely, we carried out the following steps to extend SAGE as an off-line step. For step 2 and 3, candidate terms have been recognised and classified by our pipeline for the entire SSOAR corpus.

1. Assignment of semantic types to concepts (manual)
2. Extracting terms variants such as abbreviations, synonyms, related terms from SSOAR (semi-automatic)
3. Computation of Term and Document Frequency Scores for SSOAR (automatic)

[25] https://acl-arc.comp.nus.edu.sg/

[26] Word embeddings are trained with a skip gram model using embedding size equal to 100, word window equal to 5, minimal occurrences of a word to be considered 10. Word embeddings are clustered using agglomerative clustering with a number of clusters set to 500,600,700 Ward linkage with euclidean distance is used to minimise the variance within the clusters.

[27] http://methods.sagepub.com/

[28] A glossary of statistical terms as provoded in https://www.statistics.com/resources/glossary/ has been added as well.

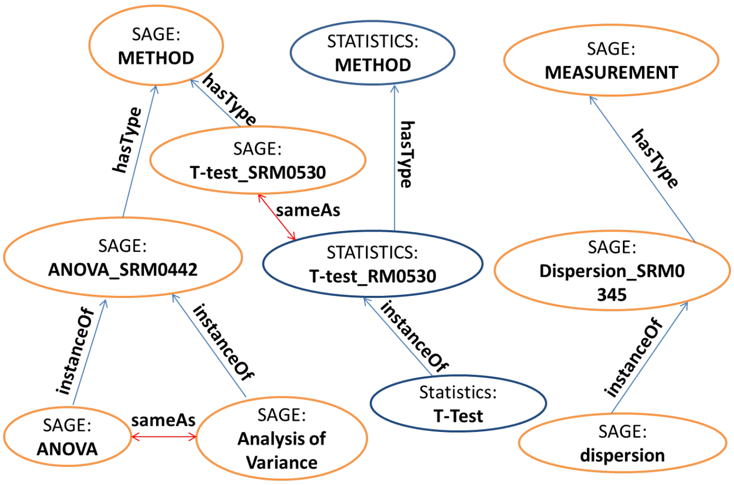


Figure 8: Integrating Domain-specific Knowledge

Extracting term variants such as abbreviations, synonyms, and related terms 26.082 candidate terms have been extracted by our pipeline and manually inspected to a) find synonyms and related words that could be linked to SAGE, and b) build a post-filter for incorrectly classified terms. Moreover, abbreviations have been extracted using the algorithm of Schwartz and Hearst (2003).

This way, a Named Entity gazetteer could be built and will be used at run-time. It comprises 1,111 terms from SAGE (i.e., 659 `skos:prefLabel` and 451 `skos:altLabel`), 447 terms from the Statistics glossary, and 54 previously unseen terms detected by the model-based classifier.

Computation of Term and Document Frequency Scores Term frequency statistics have been calculated off-line for the entire SSOAR corpus. The term frequency at corpus level will be used at run time to determine the term relevance at the document level by calculating the TF-IDF scores. The most relevant terms from SAGE are listed in Table 6.

Table 6: Most relevant terms from SAGE by Semantic Type

SAGE Term	TF-IDF Score	Semantic Class
Fuzzy logic	591,29	Research Method
arts-based research	547,21	Research Method
cognitive interviewing	521,13	Research Method
QCA	463,13	Research Method
oral history	399,68	Research Method
market research	345,37	Research Field
life events	186,61	Research Field
Realism	314,34	Research Theory
Marxism	206,77	Research Theory
ATLAS.ti	544,51	Research Tool
GIS	486,01	Research Tool
SPSS	136,52	Research Tool

Definition of a Relevance Score Relevance of terminology is often assessed using the notion of *unithood*, i.e. 'the degree of strength or stability of syntagmatic combinations of collections', and *termhood*, i.e. 'the degree that a linguistic unit is related to domain-specific concepts' (Kageura & Umino, 1996). Regarding *unithood*, the NER model implicitly contains heuristics about legal POS tag sequences for candidate terms, consisting of at least one noun (NN), preceeded or followed by modifiers such as adjectives (JJ), participles (VB*) or cardinal numbers (CD), complemented by wordshape features.

In order to find out if the candidate term also fulfills the *termhood* requirement, domain-specific term frequency statistics are required, which are set in contrast to the general domain vocabulary [29]. However, only

[29] based on COCA, i.e. a large, balanced Corpus of Contemporary American English, using *spoken, fiction, popular magazines and newspapers* rather than *academic journals*, cf. `https://corpus.byu.edu/coca/` with frequency infos at `https://www.wordfrequency.info`

a small portion of the social science terms is actually unique to the domain (e.g., 'dependent interviewing'), while others might be drawn from related disciplines such as statistics (e.g., 'conditional likelihood ratio test').

3.4.1.5 Experimental evaluation and comparison

The system evaluation is performed by comparing the extracted information with the respective gold standard dataset, i.e. ground truth data, adopting the evaluation metrics Precision, Recall and F-measure.

Evaluation metrics For evaluation, we used the NER evaluation script `conlleval`, which is frequently adopted for this task (Tjong Kim Sang & De Meulder, 2003).

The baseline performance for assigning named entity classes to word sequences is computed by means of the F1-score which considers both the precision and the recall and can be interpreted as a weighted average of the precision and recall. Standard evaluation is per entity, not per token.

Evaluation Measures

- Performance measure: $F = 2 * Precision * Recall / (Recall + Precision)$
- Precision: percentage of named entities found by the algorithm that are correct
- Recall: percentage of named entities defined in the corpus that were found by the program

Annotated scientific corpus We created a new annoted scientific corpus for the task, which includes training data from the ACL RTD corpus [30](QasemiZadeh & Schumann, 2016; Bird et al., 2008), comprising 1.500 sentences and 5,000 mentions of semantic classes which have been adapted to our scenario. Our major aims were to use the data for defining a new NER model that is able to a) use context information to discriminate ambiguous entities like 'range' which is a SAGE term and a term from the general domain vocabulary and b) extract novel entities with a proper assessment of ther *term unithood*.

Preliminary Results Our method has been developed on 5,000 English fulltext documents from the Rich Context Competition (RCC) [31]. For evaluation, a set of randomly selected 10 publications in a hold out corpus were tested by the RCC organisers. For evaluation, the judges reviewed these 10 publications of all 20 participating teams and generated qualitative scores for each document. In the first round of the competition, we received the best evaluation results for our named entity recogniser and linking tool.

3.4.2 Auditors' use case

In this section, we highlight the Named Entity Recognition (NER) extraction for the industrial use case. NER is a well-known problem and an important NLP task that automatically recognises entities in a text and classifies them in a set of pre-defined classes such as person, company names, or gene and protein names.

Since named entity recognition is a well-studied problem, a lot of state-of-the-art systems, both open source and commercial, are available. We choose to use the Stanford Named Entity Recognition software, part of the Stanford CoreNLP toolkit (Manning et al., 2014)[32], and based on linear chain Conditional Random Field (CRF) sequence models (Finkel et al., 2005a) along with a comprehensive set of discrete features that comes with the standard distribution. We rely on the Stanford NER tagger among other options, due to its state-of-the-art performance. Moreover, we have already used the Stanford's Part-Of-Speech tagger in other module of the MOVING platform, hence we were familiar to CoreNLP software and potential compatibility issues by using any other software could be eliminated. Finally, Stanford's CoreNLP provides the RegexNER[33], which is a pattern-based (i.e., rule-based) interface for doing Named Entity Recognition (NER). RegexNER can provide supplementary hand-crafted labels to the NER extraction procedure. This would be extremely beneficial for the industrial use case, since the user can provide manually new labels or new entities that are not included on the Stanford's NER system. In fact, auditors are often interested in specific known entities (such as companies) for which the specific name or abbreviation can be used to improve the annotation.

We used the latest version StanfordNER 3.9.1, and the `english.all.3class.distsim.crf.ser.gz` model with 3 classes, which is pre-trained and provides labels for locations (LOC), persons (PERS) and

[30] `https://github.com/languagerecipes/acl-rd-tec-2.0`

[31] https://coleridgeinitiative.org/richcontextcompetition

[32] Both Stanford CoreNLP and NER software are licensed under the GNU General Public License (v2 or later) and they can be used for academic research purposes and cannot be used for commercial usage.

[33] `https://stanfordnlp.github.io/CoreNLP/regexner.html`

organisations (ORG). This 3 class model was trained on the CoNLL (Tjong Kim Sang & De Meulder, 2003), MUC-6[34], MUC-7[35] and ACE[36] named entity corpora. Moreover, the RegexNER component of CoreNLP is used. RegexNER is implemented using TokensRegex[37] and annotates a textual part by using a simple text file with two tab-separated fields in every line. The first field has text to match and the second field has the entity class to assign. This class might be one of the three mentioned above classes or one of the new supplementary entity classes.

Our NER method is implemented in the Data Integration Service (DIS). For each indexed document, where there is an abstract or a full-text available, entities are recognised and stored in the corresponding entities field of the document (see Section 2). Each recognised entity is stored with a default identifier, in order to enable disambiguation at a later step. This procedure is analogue to the implementation of the author disambiguation (Blume et al., 2017; Vagliano et al., 2018).

3.5 Detecting and forecasting trending topics

Knowing what is increasing in popularity is important to researchers, news organisations, auditors, government entities and more. In particular, knowledge of trending topics provides us with information about what people are attracted to and what they think is noteworthy. Yet detecting trending topics from a set of texts is a difficult task, requiring detectors to learn trending patterns while simultaneously making predictions. We propose a deep learning model architecture for the challenging task of trend detection and forecasting. The model architecture aims to learn and attend to the trending values' patterns. Our preliminary results show that our model detects the trending topics with a high accuracy.

3.5.1 Problem statement

In the last decades, modeling of time series has received a lot of attention due to the growing need to have tools that facilitate decision making (Adhikari & Agrawal, 2013). Time series can exhibit not only regular patterns such as trends cycles, and seasonality, but also irregular patterns like structural changes, atypical sequences, effects of calendar days, etc (Harvey, 1990). The relationship between these patterns and time series are considered as an unknown independent variable. For this reason, the problem of detecting and forecasting time series values is challenging.

Various studies attempted to detect the anomalous behavior of time series(Goldberg & Shan, 2015), and analyze the popularity of certain topics. In the latter case, Cheng et. al. showed that national Twitter trends reflect the interests of the inhabitants (Chen, Spina, Croft, Sanderson, & Scholer, 2015). On the basis of this fact, Althoff et al. presented a novel approach to analyze and forecast the life cycle of trending topics (Althoff, Borth, Hees, & Dengel, 2013). They used three different datasets that are sampled from Wikipedia, Twitter, and Google news. Although the work of Althoff et al. showed that the trends' life cycle of semantically related topics is very similar, the authors' sequence matching and forecasting approach does not benefit from these similar patterns.

Motivated by this, we introduce a trend detection and forecasting approach, based on attention neural networks, which are able to learn the trending patterns of time series as well as forecasting the upcoming values. Our approach is composed of three main components. (1) The input text stream, e.g. Tweets, is passed to a sequence tagging model that extracts the main entities from the text associated with their frequency time series. (2) The frequency time series of these entities is transformed into volatility time series which measures the dispersion around the mean. (3) We utilise our forecasting model to learn the volatility time series, with a special attention units to memorise the patterns of the peaking values, and forecast trends.

We expect that the proposed trend detection method is applicable to any time series, despite focusing on Twitter time series. We focus on Twitter because Tweets generate a significant data flow, for which it is worthy to apply trend detection methods and because the MOVING platform crawls Twitter posts. This is particularly interesting for the auditors which use the platform, for example, an early detection of trends, such as the emerging hashtag `#dieselgate`, could be useful if Wolkswagen, one of its subsidiaries, or any other organisation involved is among the audited companies.

[34] https://catalog.ldc.upenn.edu/LDC2003T13
[35] https://catalog.ldc.upenn.edu/LDC2001T02
[36] https://catalog.ldc.upenn.edu/LDC2014T18
[37] http://nlp.stanford.edu/software/tokensregex.html

3.5.2 Related work

Goldberg and Shan presented a statistical approach for detecting anomalies in eBay's search results (Goldberg & Shan, 2015). For every search query, they collected 50 metric values such as the number of the returned search results and the median price of the returned items. They defined an anomaly to occur when any value of any of the 50 time series is greater than the median of that series of at least three times its standard deviation. Althoff et al. (Althoff et al., 2013) presented a novel model for detecting the life cycle of trending topics. They used three different datasets: Wikipedia Pageview statistics, Twitter, and Google news. In the case of Wikipedia, they used the number of page views over one year to infer the trending topics. They showed that the page views forecast of their approach is about 9-48k views closer to the actual viewing statistics. More specifically, their approach accumulated a mean average error of 45-19, for time period of up to 14 days, comparable to 20-90 % using the baseline line approaches. From a machine learning prospective, it is hard to detect data streams patterns due to the lack of the ground truth examples. On the other hand, the unsupervised models usually assume that infrequent patterns are anomalous. In the literature, there is a growing interest in forecasting time series values using artificial neural networks (ANN) (Adya & Collopy, 1998; Crone & Kourentzes, 2009; Hill, O'Connor, & Remus, 1996; G. Zhang, Patuwo, & Hu, 1998). Although many successful reports have been presented. Zhang et al. (G. Zhang et al., 1998) identified inconsistent results about the use of neural networks in time series forecasting. The authors argue that the inconsistencies can be due to different reasons such as: (1) the inadequate configuration of the network, (2) over-fitting, and (3) the lack of reliable and valid evaluations. In contrast with the previously mentioned ANN supervised models, we introduced an *unsupervised* prediction model that detects the trends using an attention deep learning techniques.

3.5.3 Method description

We design a preliminary attention model that is capable of detecting the trending topics in text streams. Attention models are encoder-decoder models that support the decoder to pick only the encoded inputs that are important for each step of the decoding process (Xu et al., 2015).

Assume that we have a sequence of n input time steps $Input = \{inp_1, ..., inp_{n-2}, inp_{n-1}, inp_n\}$ that contains an eBay Atlas trend (Goldberg & Shan, 2015) on inp_n, our model is required to forecast the trend as early as possible i.e. on inp_{n-3}, inp_{n-2} and detect the trending value inp_n. In the ebay Atlas trend, the trend occurs when one value of the time series is equal to or greater than 3σ of the time series values, where σ is the standard deviation. We define an output vector of two sets. The first set consists of the emerging trend values, while the second set represents the next time step of the time series that contains the ebay Atlas trend itself. For example, in the following time series $S = \{1,3,2,6,7,\mathbf{11},\mathbf{13}\}$, the expected input vector is $Input = \{1,3,2,6,7,\mathbf{11}\}$ while the output sequence vector will be $Output = \{\{0,0,0,6,7,\mathbf{11}\},\{\mathbf{13}\}\}$. In this case, we teach the network to attend to the trending values pattern (in bold), and forecast whether the trend will continue.

The model is composed of three main layers; namely encoder, attention and decoder. In the first layer, the encoder encodes the entire input sequence, for each time step, into a fixed length vector (Cho et al., 2014). The attention layer calculates the importance of the encoded vector e at decoding step j. Therefore, the attention model requires access to the output from the encoder for each input time step, and the internal hidden state of the decoder cell, h_{i-1}.

In order to calculate the attention probabilities $\alpha = (\alpha_0, , \alpha_n)$, a feed-forward neural network calculates the formalised importance of the encoded vector in step i in predicting j (Equation 3).

$$\beta_{ij} = V \cdot \tanh(W_1 \cdot e_i + W_2 \cdot h_j) \tag{3}$$

Then, the softmax operation (Jain, Mao, & Mohiuddin, 1996) normalises the probabilities of the encoded vectors by multiplying each encoded vector by its weight, in order to obtain a time dependent input encoding which is fed to each step of the decoder.

$$\alpha_{ij} = \frac{exp(\beta_i)}{\sum_{k=0}^{n} exp(\beta_i)}, c_j = \sum_{k=0}^{n} \alpha_{k,j} e_k \tag{4}$$

Afterwards, the model calculates the context vector. This vector summarises the importance of the different encoded values (Equation 4). The decoder steps through the output time series while reading from the context vector. Therefore, the attention model computes a weighted representation of the whole input sequence for each slide of the decoder, and the decoder can learn to pay attention to the trending values.

3.5.4 Experimental evaluation and comparison

From the sequence labeling approach, we obtain a time series of entities frequencies. This time series is considered as the input layer for our proposed attention model. For each entity time series, the model detects the trending values. To evaluate the time series prediction performance, we use the RooT Mean Squared Error (RMSE). RMSE measures the average squared difference between the real trending values and what is predicted.

Dataset. We use the *Tweets2011 dataset*[38]. The dataset is provided by the National Institute of Standards and Technology (NIST). It was published as a part of the micro-blog track in 2011. The dataset contains Twitter identifiers of approximately 16 million tweets. The tweets identifiers cover the period between January 23rd and February 8th, 2011.

Entities extraction. In order to extract entities from a set of texts, a dataset to train the named-entity recognition (NER) model is required. Such a dataset usually consists of texts and labels for each token of the text. We train the model with the W-NUT dataset (Strauss, Toma, Ritter, de Marneffe, & Xu, 2016). The dataset consists of 2,400 tweets and 34,000 entities with 10 different annotated types. We use the model that obtains the state-of-art performance in NER. The model uses a bi-LSTM and CRF with character embedding (Lample et al., 2016). We use cross validation to induce the optimal model training parameters. The trained model was then used to extract all entities from the Tweets2011 collection. From Tweets2011, we extracted $331,913$ entities.

Detecting trends. After generating the frequency time series of entities, we use the min-max normalisation (Patro & Sahu, 2015) to normalise the series values. Afterwards, we pass the normalised time series values for each entity to our trend detection model, attend2trend. As with every other deep learning model, the training process is computationally expensive due to the high number of parameters that should be tuned. In order to find the best set of parameters that reduces the mean squared testing error, our model has been executed with different parameters, each execution consists of 150 iterations (epocs). The initial set of parameters to choose from are:

- $Windowsize := [0, 150]$.
- $Activationfunction := \{Tanh, Sigmoid\}$.
- $Batchsize := [1, 10]$.
- $Numberofhiddenunits := [100, 200]$.

We use the adaptive gradient descent algorithm Adam for tuning the learning rates (Kingma & Ba, 2014).

After finding the best parameters for each dataset, we apply the holdout cross validation(Duda & Hart, 1973) to train and test the model. From the Tweets2011 dataset, only 280 entities (out of $331,913$) contain obvious eBay Atlas trends. The preliminary results shows that our encoder-decoder model, powered by the attention units, was able to detect the trends with an accuracy of 94.05% and RMSE of 1.98, comparable with the accuracy of 76.19% and RMSE of 7.80 of the classic LSTM encoder-decoder model. The integration of this model on the MOVING platform will depend on the final evaluation results, currently being computed as well as the impact on the platform's performance.

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

3.6.1 Problem statement

Semantic annotations are important for the MOVING platform as they enhance the search of scientific documents. Given the large amount of new publications, automatic annotation systems are a useful tool to classify the publications into categories from a (hierarchical) thesaurus. However, providing automated recommendations for subject indexing in such systems is a challenging task. This is partly due to the data from which recommendations may be generated. Often neither the full-text of a publication nor its abstract may be available. For instance, the MOVING platform contains a limited number of full-text due to legal barriers. Even when the content can be legally provided to the end users, copyright laws or regulations of the publishers may prevent text mining. Moreover, collecting and processing PDFs where possible, e. g., for some Open Access

[38] https://trec.nist.gov/data/tweets/

documents, adds high computational requirements to the library. Thus it is better for annotation methods to rely on data with better availability, such as the title. Previous work (Galke, Mai, Schelten, Brunsch, & Scherp, 2017), however, has shown that title-based methods considerably fall behind full-text methods in terms of performance when the number of samples for training is equal. If our classifier was a human expert, this would not be a surprising result. A full-text contains more information and therefore also more indication of the publication's topic. A human expert will always make better annotations based on the full-text. In fact, the gold-standard annotations for automated subject indexing experiments are created based on the full-text.

However, we argue that machine learning algorithms work differently than a human. In contrast to a human, they often require hundreds of thousands or even millions of training data to yield satisfactory models (Brain & Webb, 1999). These amounts of data are not always available in the real world. One common reason is that human expertise is required for creating a large enough gold standard, which is expensive. For semantic subject indexing, the availability issues mentioned above do not only come into play at prediction time, i.e., when a machine learning model is used in a productive system, but also during training. In effect, methods based on the full-texts have drastically less training data available than methods based on titles. This raises the question if title-based methods can potentially narrow the performance gap to full-text methods by fully incorporating all training data available.

Formally, subject indexing is framed as a multi-label classification problem, where a (commonly small) subset of labels has to be selected from a (relatively large) set of labels. From two digital libraries of scientific literature, PubMed[39] and EconBiz[40], we have compiled an English full-text dataset and an English title dataset. Our compiled datasets are quite different with respect to their size.

From PubMed, we extracted 12.83 million titles, for 5% of which a full-text is available (646k). From EconBiz, we extracted 1.06 million titles, of which approximately 7% have a full-text (71k). In order to fully utilise these large amounts of data, we develop and compare three different classifiers that have emerged from the deep learning community in recent years. Deep learning has advanced the state-of-the-art in many fields, such as vision, speech, and text (LeCun, Bengio, & Hinton, 2015). These techniques are known to shine when a lot of training data is available. For text classification in particular, recent work (X. Zhang, Zhao, & LeCun, 2015) suggests that deep learning starts to outperform strong traditional models when 650k or more training samples are available. The number of full-text in PubMed is right at the edge of this number, whereas EconBiz has far less full-texts, making these datasets an interesting and revealing choice.

In natural language processing, different types of neural networks have been successfully employed on different tasks, but it is an open question whether convolutional neural networks (CNNs), recurrent neural networks (RNNs), or multi-layer-perceptrons (MLPs) are superior for text classification tasks. Therefore, we employ a representative of each type in our study. We compare them against another strong MLP baseline (*Base-MLP*), which has previously been shown to also outperform traditional bag-of-words classifiers such as SVMs, Naive Bayes, and kNN (Galke et al., 2017).

Since the label space in our datasets is very large, our study can be understood as *eXtreme Multi-Label Classification* (XMLC). Here, only few studies have leveraged deep learning techniques to tackle the considerably harder problem when the label space is large (W. Zhang, Wang, Yan, Wang, & Zha, 2017; Liu, Chang, Wu, & Yang, 2017). Hence, with our study, we contribute to the knowledge in this field, as well.

The results of our study indicate that title-based methods can match or even outperform the full-text performance when enough training data is available. On EconBiz, the best title classifier (MLP) performs on par with the best full-text classifier (MLP) when only 8× as many titles are used for training than for full-text. When all available titles are used (approximately 15× more than full-texts), the title-based MLP outperforms its full-text counterpart by 9.4%. On the PubMed dataset, the best title method is the RNN, and it almost reaches the best full-text performance produced by an MLP. The gap is only as small as 2.9%. When using the same number of titles as full-texts are available, the gap in classification performance is 10.7%, indicating a considerable benefit from leveraging all title data.

Generally, the MLP performs well, outperforming RNN and CNN in all but one case. It also consistently outperforms the baseline when all titles or full-texts are used. Moreover, our analysis suggests that our proposed classifiers are well-chosen for our study, because they benefit from increasing amounts of training data better than the baseline.

In summary, in this section, we investigate whether title-based methods can reach the performance of full-text-based methods by exploiting the surplus of available training data, and we demonstrate that title-based methods are on par or even outperform full-text methods when the number of training samples is sufficiently large.

[39] https://www.ncbi.nlm.nih.gov/pubmed/

[40] https://www.econbiz.de/

3.6.2 Method description

Training Procedure The semantic annotation task is formally a *multi-labeling problem*, where instead of belonging to exactly one class, each publication is assigned a set of labels. This is an important difference to most of the previous literature on text classification with large datasets. *Binary Relevance* is a common technique to adapt a classifier for a multi-labeling problem. However, this is a costly technique when the number of labels is high because it requires to train as many classifiers as there are labels. Neural networks have a more natural way to deal with multi-label classification, which is also used by Galke et al. (Galke et al., 2017). For multi-class text classification, the softmax activation function is used at the output layer to obtain a probability distribution over the classes. For multi-label classification, however, the sigmoid activation can be employed to determine a probability p_l for each label l whether it should be assigned or not. The difference is that softmax regards all labels at once, while sigmoid makes an independent decision for each label. Finally, the binary decision whether label l is assigned is made by checking whether p_l exceeds a threshold θ.

Using Adam (Kingma & Ba, 2014), the networks are trained as to minimise the sum of all binary cross-entropy losses over all labels. This has shown to be superior to a ranking-based loss on multi-label text classification (Nam, Kim, Loza Mencía, Gurevych, & Fürnkranz, 2014; Liu et al., 2017). Training is executed in mini-batches of size 256. We employ early stopping for regularisation and as criterion to terminate training.

Since the output of the sigmoid activation function can be interpreted as the probability whether a label should be assigned, a typical choice for the threshold is $\theta = 0.5$. However, depending on the evaluation metric, dataset, or model that generates the assignment probabilities, this value does not necessarily yield optimal results. Unfortunately, finding a good value for θ can be computationally expensive, especially when the datasets are very large. Therefore, we use a heuristic that continuously adjusts the threshold *during* training. To this end, the evaluation on the validation set used for early stopping is also used to optimise θ.

Formally, we initially set $\theta_0 := 0.2$, which is a better threshold value than 0.5 (Galke et al., 2017). After each validation step i, where the classifier predicts a probability for each of the $|L|$ labels and each of the n samples in the validation set, accumulated in $P_i \in (0,1)^{n\times|L|}$, we set

$$\theta_i := \operatorname*{arg\,max}_{\tilde{\theta}\in\{-k\cdot\alpha+\theta_{i-1},\dots,k\cdot\alpha+\theta_{i-1}\}} F_1(P_i;\tilde{\theta})$$

where $\alpha > 0$ is the step size and k controls the number of threshold values to check. This heuristic is motivated by the observation that the optimal choice for θ_i is in most cases in close to the optimal choice of the previous evaluation step, θ_{i-1}. Since computing the F_1 score can be costly, we set $k = 3$ and $\alpha = 0.01$ to trade off granularity with speed.

In our preliminary experiments, this way of optimizing the threshold consistently yields good results, sometimes even better than manual tuning.

Multi-Layer-Perceptron Our baseline is a multi-layer-perceptron (MLP) described by Galke et al. (Galke et al., 2017). It has one hidden layer with 1,000 units and rectifier activation and it takes a TF-IDF (Salton & Buckley, 1988) bag-of-unigrams as input. The bag-of-unigrams only contains the 25,000 most common unigrams, which we determined to be sufficient for the multi-labeling task. For regularisation, dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is applied after the hidden layer with a keep probability of 0.5. We will refer to this baseline as *Base-MLP*.

We extend the MLP from Galke et al. (Galke et al., 2017) by incorporating some techniques inspired from recent deep learning literature. The MLP architecture introduced in this study can be viewed as a non-linear adaptation of fastText (Grave, Mikolov, Joulin, & Bojanowski, 2017) for multi-label classification. FastText is a popular text classification library that excels at training speed while still attaining classification performance close to state-of-the-art. Fast training is obtained through two major design decisions. First, the costly softmax operation was approximated through hierarchical softmax. Since in this study we deal with a multi-labeling problem and thus employ sigmoid, this is not a concern for our models. Secondly, they employed an efficient linear bags-of-words (BoW) model that they enhance with feature sharing and local word-order information. The feature sharing component is introduced through a hidden layer with identity activation function (in order to retain a linear classifier). The outputs at the hidden layer are then latent features shared among all classes. This architecture of fastText is already similar to our model, except that we employ a non-linear activation function (rectifier). However, for our study, computational speed is not of essence. Moreover, we found experimentally that omitting this non-linearity in fact hurts the classification performance considerably. In fastText, local word-order information is provided in the form of bi-grams. For our models, we integrate this information by adding the 25,000 most common bi-grams in addition to the 25,000 most common unigrams.

In the introduction, we mentioned that deep neural networks excel when the number of training samples is large. This is because the representational power of neural networks increases as the number of parameters increases. The representational power, i. e., capacity, increases by using more layers or more hidden units per layer. Additionally, deeper networks may be able to learn hierarchical representations of the input, as can be observed in the vision domain (He, Zhang, Ren, & Sun, 2016). The problem with deep neural networks is that they are generally harder to train due to the vanishing and exploding gradient problems (Bengio, Simard, & Frasconi, 1994). In order to alleviate this problem, a lot of techniques have been proposed that attempt to normalise the output of a layer, so the respective gradient is well-behaved. Among those techniques, we picked the well-established Batch Normalisation (Ioffe & Szegedy, 2015) to our deep MLPs.

In our experimental evaluation, we found wider networks to do better than deeper networks in most cases. The only exception is the PubMed title dataset, where a two-layer neural network with Batch Normalisation performs best.

In summary, MLP differs from Base-MLP in that it incorporates bi-grams and uses multiple layers and Batch Normalisation where appropriate.

Convolutional Neural Network We present a CNN architecture whose core was introduced by Kim (Kim, 2014) for sentence classification and has since been repeatedly adopted and enhanced upon. We adopt and combine some of these enhancements for our model.

The CNN operates on word embeddings, which are initialised with a pretrained model but fine tuned during training. As in Kim's model, our CNN applies a 1D-convolution by sliding a window over the text in order to extract features at each position. These outputs are then transformed by a non-linear activation function (the detector). Commonly, the most salient position is selected by applying max-pooling after the detector stage. However, Liu et al. (Liu et al., 2017) instead split the output of the convolution into p nearly equal chunks, and perform max-pooling on each chunk. Afterwards, the outputs of the pooling stages are concatenated. For $p = 1$, this is identical to Kim's architecture. In our experiments, using $p > 1$ improved performance only on full-text. This is comprehensible, because titles are already quite short, so they do not need to be split in chunks.

Commonly, this process is repeated for multiple window sizes. The outputs of these processes are then concatenated before passing them to the next layer. For example, Kim's CNN uses window sizes 3, 4, and 5, while Liu et al. use 2, 4, and 8. We experimentally determined that using 2, 3, 4, 5, and 8 yields to even better results.

In Kim's model, the concatenated output of the pooling stages is directly propagated to the output layer. Liu et al. argue, however, that it is better to have an additional fully-connected layer, called the *bottleneck layer*, with n_b units, where n_b is smaller than the output of the convolutional layer, injected before the output layer. The reasoning is twofold. First, a more narrow layer after the pooling actually reduces the number of parameters in the model if the number of output labels is large. Secondly, it adds more representational power to the network through the increased depth. For our model, we also found that a bottleneck layer is beneficial.

Considering the complexity of our datasets and the number of training samples available, the question of increased capacity arises with CNNs. Similar to MLPs, an increase in capacity can be achieved either through wider convolutions (larger feature-map) or additional stacked layers of convolutions. Since a recent study by Le et al. (Le, Cerisara, & Denis, 2018) has shown that depth does not yield improvement over shallow nets, we only consider the former approach. Previously, a feature map size of around 100 was a common choice (Kim, 2014; Le et al., 2018; Liu et al., 2017). On our datasets, we often found a size of up to 400 to considerably improve the results, even on the full-texts of EconBiz, which has relatively few samples.

Recurrent Neural Network The Recurrent Neural Network (RNN) is a family of neural networks that was specifically designed for sequential input data (Goodfellow, Bengio, & Courville, 2016). By maintaining a hidden state, the network is able to keep track of previous inputs. However, the vanilla RNN has difficulties keeping track of inputs that are far in the past (Bengio et al., 1994). Some solutions to this problem have been proposed, most prominently the LSTM (Hochreiter & Schmidhuber, 1997) and the Gated Recurrent Unit (GRU) (Chung, Gulcehre, Cho, & Bengio, 2014). Both architectures base on the principle that they explicitly model the control over whether the current hidden state is discarded, updated, or kept. Both the LSTM and GRU oftentimes yield comparable performance. For this study, initially we use an LSTM that has already achieved good results for text classification in a study by Zhang et al. (X. Zhang et al., 2015). This LSTM is the "vanilla" version described in (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017). Our final model, however, incorporates two techniques which have proven useful for NLP tasks in recent years, attention and bidirectionality (Young, Hazarika, Poria, & Cambria, 2017).

Since any RNN produces an output at every time step, the outputs have to be aggregated after processing the entire sequence, such the higher layer receives a vector of fixed size to the next layer. We experimented with choosing the last output, computing the sum, computing the average, and computing a weighted average where the weights are determined by an attention mechanism as used by Yang et al. (Yang et al., 2016). While the benefit over the other aggregation methods is not large for titles, the attention mechanism consistently performs best. On full-texts, however, the difference is considerable. This is intuitive, because there is less need to focus on specific parts of the input if the input is as short as a title.

A standard RNN reads the sequence from left to right. That means that at any time step i the RNN's state does not encode information that follows step i, even though those may be critical for understanding the current step of the sequence. Bidirectional RNNs are a simple yet effective way to address this problem. Here, one standard RNN reads the sequence from left to right, and another RNN reads it from right to left. The outputs of both RNNs at each time step are then concatenated. Bidirectional RNNs are a common technique to boost the performance over standard RNNs. For text classification in particular, Yang et al. (Yang et al., 2016) have employed a bidirectional GRU. In our study, we use a bidirectional LSTM, since we found it to improve the performance considerably.

Again, we made some effort to investigate an increase in the capacity of the LSTM to account for the large number of training samples in our datasets. In the past, both increasing the memory cell size (the width) and stacking LSTMs on top of each other has been successful in some NLP tasks. For text classification, this has not been the case. The results of our experiments support this, where wider LSTMs are superior to stacked LSTMs, even with variational dropout (Gal & Ghahramani, 2016). The single-layer LSTM used by Zhang et al. (X. Zhang et al., 2015) uses a cell size of 512. For our final experiments, we found cell sizes up to 1,536 work better. Beyond that, the training time becomes unacceptable.

3.6.3 Experimental evaluation and comparison

Datasets We built English datasets from two digital libraries of scientific publications: EconBiz[41] and PubMed[42]. EconBiz is a search portal for economics and business studies. It contains 2,485,000 English publications, out of which 615k are open access. Domain experts have annotated the publications with a variable number of subject headings taken from a controlled vocabulary, the "Thesaurus for Economics" (STW)[43]. After deleting duplicates, 1,064,634 publications remain. Out of the 615k open access publications, the number of publications that have annotations and whose full-text can be downloaded and processed reduces to 70,619, which is 6.63% of all publications.

PubMed is a search engine for biomedical and life science literature provided by the US National Library of Medicine. The publications found on PubMed are annotated by human curators with another controlled vocabulary, namely "Medical Subject Headings" (MeSH)[44]. We obtained a dataset consisting of millions of publication metadata, including title and MeSH annotations, from the training set of the subject indexing task of the BioASQ challenge 2017 (Tsatsaronis et al., 2015), which are all in English language. PubMed Central[45] is an archive of full-texts of biomedical and life science literature provided by the US National Library of Medicine. It comprises 4.3 million publications, which can be accessed freely and which are mostly English. However, only 1.5 million are open access and therefore allow text mining. From this dataset, we computed the intersection with the publications obtained from the BioASQ challenge. After removing duplicates, 12,834,026 titles and 646,513 full-texts with respective annotations remain. Hence, 5.04% of the samples have a full-text.

Table 7 lists some characteristics of the two datasets, EconBiz and PubMed. In terms of combinatorial complexity, the PubMed dataset is a harder problem because the number of labels out of which to pick the annotations for a publication is much higher. Yet, due to the relatively large number of labels and small number of samples per label on average, both datasets can be considered as XMLC problems[46]. The titles in PubMed contain on average more words than the publications' titles in EconBiz. However, this fact is put into perspective considering that the titles in PubMed have on average more labels to be predicted than there are words in the title. Regarding the full-texts, the ratio of words/labels is approximately the same in both datasets. Another fact worth noting is that the titles corpora have on average one label less than the full-texts. This suggests that the label distributions in the title dataset and full-text dataset are quite different.

Please note that on both datasets, the set of titles is a superset of the set of full-texts.

[41] https://www.econbiz.de/

[42] https://www.ncbi.nlm.nih.gov/pubmed/

[43] http://zbw.eu/stw/version/latest/about

[44] https://www.nlm.nih.gov/mesh/meshhome.html

[45] https://www.ncbi.nlm.nih.gov/pmc/

[46] An overview of datasets commonly considered XMLC can be found at http://manikvarma.org/downloads/XC/XMLRepository.html.

Table 7: Characteristics of EconBiz and PubMed datasets. $|D|$ denotes the sample size, $|L|$ denotes the number of labels used in the dataset, d/l is the average number of publications a label is assigned to. l/d is the average number of labels assigned to a publication. $|V|$ is the size of the vocabulary and w/d denotes the average number of words per document.

	EconBiz (STW)		PubMed (MeSH)	
	Title	Full-Text	Title	Full-Text
$\|D\|$	1,064,634	70,619	12,834,026	646,513
Size	78.8MB	6.27GB	1.32GB	20.06GB
$\|L\|$	5661	4849	27773	26276
d/l	819.1	75.8	5852.3	331.0
l/d	4.4	5.3	12.6	13.5
$\|V\|$	91,505	1,502,336	660,180	6,774,130
w/d	6.88	6694.4	9.6	2533.4

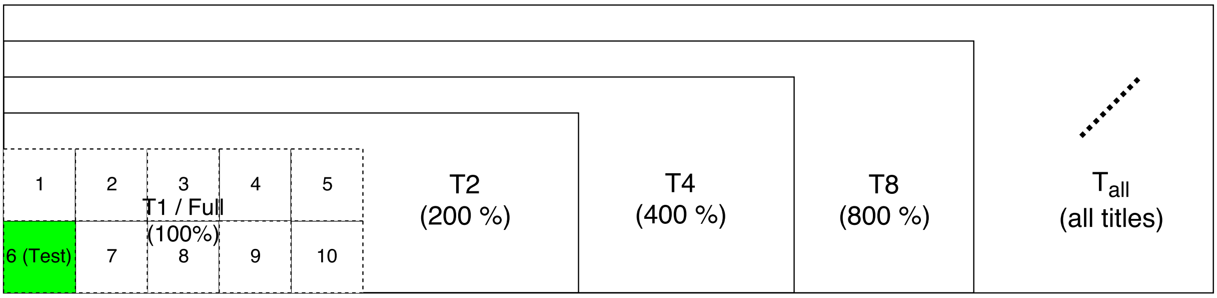


Figure 9: We organise our dataset in several sub-datasets to perform an iterative evaluation. T1 and Full comprise of the same set of publications (all samples where a full-text is available), and is split into 10 folds in order to perform the same 10-fold cross-validation with titles and full-texts for a fair comparison. For T2, T4, T8, increasingly more titles that do not have a full-text are added for training, but are in each cross-validation step evaluated on the same test samples (highlighted in green for exemplification). T_{all} includes all title samples.

Preliminary Experiments In order to assess how the title-based methods behave as more and more titles are considered for training, we create sub-datasets of the title datasets by iteratively adding more data. This is illustrated in Figure 9.

We perform a number of preliminary experiments to determine good hyperparameter values that we can fix for subsequent experiments. Unless specified otherwise, in the subsequent experiments, MLP, CNN, and LSTM refer to the base models, Base-MLP, Base-CNN, and Base-LSTM. CNNs and LSTMs use an embedding size of 300 without pretraining. Following Galke et al (Galke et al., 2017), we set the learning rate to $\alpha = 0.01$ and the keep probability to $p = 0.5$. For the validation frequency $\tilde{b}$ we choose one epoch. The binary decision whether or not to assign a label is made with the threshold method, where the threshold is adjusted during training. For MLPs, the input is a TF-IDF vector with the 50,000 most common words in the vocabulary.

On each (sub)-dataset, we would like to find a small vocabulary for BoW-based classifiers (i.e., MLPs), such that little to no information relevant to the prediction task is lost. We achieve this by performing preliminary experiments where we show that the classification performance does not suffer from restricting the vocabulary to the m words that appear most frequently in the training corpus. In particular, we assess the impact of the choice of m on the classification with the *kNN* algorithm $(k = 1)$. At test time, the algorithm assigns the same labels that are assigned to the closest sample in the training set, where distance is measured in terms of cosine similarity. By choosing kNN as our classifier, we adopt the reasoning by Galke et al. (Galke et al., 2017) that the kNN algorithm is suitable for assessing the quality of the features. For kNN in particular, a good feature representation is critical to achieve a good classification performance. We compute a TF-IDF representation of the text, but we only retain the m most common terms. We run one experiment per dataset on each of T1, T_{all}, and Full. This is necessary because these sub-datasets have very different vocabularies due to different domains and sample sizes. We run the 1NN algorithm on a 90:10 training-test split. On the EconBiz dataset, the differences between scores on the same sub-dataset are 0.003 at maximum across all vocabulary sizes. On the PubMed dataset, the differences are larger, but in inconsistent directions. While the performance on titles decreases by up to 0.013 when going from unlimited vocabulary size to 25,000, the performance on full-texts increases by 0.005 when going from unlimited to 25,000.

The effects of performing early stopping on a validation set as opposed to not doing that vary a lot across the datasets. On full-texts, the performance drops considerably on all datasets but Reuters, where we can observe a marginal increase of 0.003. Most notably, the difference for NYT is as much as 0.083. On titles, however, early stopping helps for Reuters and NYT, and the decrease on the economics and political science datasets are smaller than on full-texts. Second, we can observe that additionally optimizing the threshold on the validation set always helps. While the difference is rather small on Reuters (0.004), learning the threshold boosts the performance by up to 0.017 on the other datasets. Nonetheless, in 4 out of 8 cases, the performance with early stopping and threshold learning is considerably worse than when no early stopping is used.

When running our neural networks with a validation frequency $\tilde{b}$ of one epoch (a commonly chosen standard value), we noticed that on PubMed the performance of the titles declined when adding more training data. In our experiments, T1, T2, and T4 achieved an F_1 score of 0.462, 0.452, and 0.440, respectively. A reason could be that due to the size of the PubMed dataset, the classifier reaches its optimal parameter values in terms of validation set performance somewhere in the middle of an epoch, but has already overfit the dataset at the end of that same epoch. As the validation set performance is only assessed at the end of each epoch, it uses an overfitted model at test time and thus yields bad performance. In order to verify that this is indeed what causes the drop in performance when more training samples are added, we conducted experiments where we determine the validation set performance after a set number of batches, i.e., weight updates.

The larger the dataset, the more likely is the above scenario of overfitting to happen. Therefore, we only looked at our largest datasets, i.e., T_{all} from EconBiz and PubMed, respectively. We assess the validation performance after 2,000, 1,000, and 500 batches, respectively, and compare to when the validation is performed after an entire epoch. Since the illustrated problem could occur irrespectively of the type of neural network, we only run experiments with the MLP.

On EconBiz, the differences are barely noticeable. On PubMed, there too is no difference between validation after 500 batches, 1,000 batches, and 2,000 batches, respectively. However, when validating after a whole epoch (after seeing $25\times$ as many training samples before validation compared to 500 batches), the performance drops drastically by 0.094 in F_1-score.

We experiment with GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013), which have shown to be the most successful on intrinsic tasks (Schnabel, Labutov, Mimno, & Joachims, 2015) and are commonly used. Additionally, we experiment with embeddings that incorporate subword information (called *fastText* here), which have recently been shown to improve the quality of embeddings further (Bojanowski, Grave, Joulin, & Mikolov, 2017).

Due to differences in the vocabularies they were trained on, different pretrained embedding models may perform differently depending on the dataset. We therefore evaluate them on PubMed and EconBiz separately. In contrast, the quality of embeddings may be assumed to benefit both CNNs and LSTMs by the same relative amount. Therefore, for a relative comparison of the embeddings, we run experiments only with LSTMs and will assume that CNNs will perform accordingly. Moreover, we assess whether finetuning the pretrained word embeddings helps the classification performance or not.

The effect of finetuning is very inconsistent across the datasets. While it benefits fastText on PubMed by 0.024 in terms of F_1-measure, it degrades the performance on EconBiz by 0.043. GloVe shows slight improvement on both datasets with finetuning, while word2vec is always weakened. It is noteworthy that even with no pretrained embeddings the model can perform better than with some pretrained embeddings that are finetuned. Overall, finetuned fastText wins on PubMed, but is closely followed by finetuned GloVe. On EconBiz, GloVe is the winner.

In order to enable training in batches, we need to limit the length of the sequence. The titles datasets have a maximum sequence length of 60 (PubMed) and 84 (EconBiz), respectively, which is not problematic. Therefore, we conduct experiments only on the full-text datasets. Moreover, since CNNs and LSTMs react very differently to the length of the sequence, we need to assess its impact on both classifiers. Finally, the effect of the sequence length might also differ depending on the dataset. This can be seen when considering that full-texts in PubMed are on average much shorter than full-texts on EconBiz (see Table 7). In our experiments, we evaluate sequence lengths 100, 250, 500, and 1,000, respectively.

The results show that the top performance on both datasets and with both models is reached at a sequence length of 250. However, the CNN is rather invariant with respect to the lengths, which results in relatively small differences in F_1-score of 0.025 at maximum. The LSTM, on the other hand, is very sensitive to the length of the sequence. On EconBiz, it benefits a lot from seeing more words up to 250, increasing the performance by 0.066 compared to a sequence length of 100. The performance is stable when increasing the length to 500, but collapses completely when increasing the length to 1,000.

Experiments For a fair comparison of title and full-text performance, the trained model must be evaluated on the same data. To this end, we split the set of publications where a full-text is available into ten folds, and perform a 10-fold cross-validation (Amatriain, Jaimes, Oliver, & Pujol, 2011). In each iteration, nine folds are selected for training, and one is selected for testing. From this training set, we randomly select 20% for the validation set for early stopping and adjusting the threshold, as described in Section 3.6.2. These comprise the data used for our experiments on full-text, and will be abbreviated as EconBiz Full and PubMed Full, respectively.

For the experiments on titles, the same publications from the 10-fold cross-validation are used for testing. However, the training set is iteratively extended with more samples from the titles dataset, so that the total number of training samples is always a power of two of the number of samples in the full-text experiment. In total, we conduct experiments on five title sub-datasets per domain: T1, T2, T4, T8, and T$_{all}$. Here, Tx means that x times as many title samples are used for training as there are full-text samples in the dataset. Lastly, T$_{all}$ contains all title samples from the dataset.

We run each of the four classifiers Base-MLP, MLP, CNN, and LSTM on all of the sub-datasets. In total, we run 48 cross-validations. Following Galke et al. (Galke et al., 2017) who argue that the sample-based F_1-metric best reflects how subject indexers work, we use this metric to report the results. We also use this metric for early stopping and threshold adjustment on the validation set.

Choice of Hyperparameters and Training Since there are a lot of tunable hyperparameters involved in deep learning, tuning multiple hyperparameters at the same time can be very expensive, especially when the datasets are very large. On the other hand, fixing hyperparameters across all datasets and models would not be a fair approach in our study because the datasets and model architectures are very different and therefore may require very different hyperparameter settings. As a compromise, we decided to tune the hyperparameters for full-texts and titles separately in an incremental fashion on one fold. Here, we tuned one hyperparameter at a time and selected the locally best solution for full-text and titles, respectively. It is important to note that the parameters for titles were determined based on the performance on T$_{all}$, and were adopted for all other title sub-datasets. On the one hand, this alleviates a lot of the computational cost and allows to compare the performance between title sub-datasets. On the other hand, especially the performance of smaller sub-datasets might be suboptimal due to overfitting. This must be kept in mind for analysis.

In our experiments involving the MLP, we use a one-layer MLP with 2,000 units and dropout with a keep probability of 0.5 after the hidden layer for all experiments. Only for the experiments on the PubMed titles, we use a two-layer MLP with 1,000 units each, and apply no dropout. Instead, we use Batch Normalisation after each hidden layer. In all cases, the initial learning rate for Adam is set to 0.001.

In the CNN experiments, we use $p = 3$ chunks and $n_b = 1,000$ units at the bottleneck layer (Liu et al., 2017) for both full-text experiments. On titles, we do not perform chunking ($p = 1$), and use a bottleneck layer size of $n_b = 500$. The size of the feature map is set to 400 in all experiments except for PubMed Full, where we use 100. The keep probability is set to 0.75 in all cases, and the initial learning rate is 0.001.

We use a single-layer LSTM for all experiments. For both datasets, we determined 1,536 to be the best size for the memory cell when using titles. 1,024 units and 512 are used for PubMed Full and EconBiz Full, respectively. The keep probability is set to 0.75 in all experiments except for PubMed on titles, where we set it to 0.5. The initial learning rate is 0.01 for EconBiz Full and 0.001 in all other cases. Training is done with backpropagation through time by unrolling the LSTM until the end of the sequence.

We adopt the preprocessing and tokenisation procedure of Galke et. al (Galke et al., 2017). For the LSTM and CNN, we use 300-dimensional pretrained word embeddings obtained from training GloVe (Pennington et al., 2014) on Common Crawl with 840 billion tokens[47]. Out-of-vocabulary words are discarded. The maximum sequence length is limited to the first 250 words. Longer sequences were harmful in preliminary experiments.

For implementation of our neural network models, we used the deep learning library TensorFlow[48] and integrated them within the multi-label classification framework "Quadflor"[49]. We conduct all experiments either on an NVIDIA TITAN or on a TITAN Xp GPU which both have 12GB of RAM.

Results The results of our experiments are shown in Table 9. In addition, we plot the performance of each method as a function of the number of samples used for training the title model. These are shown in Figure 10.

On the EconBiz dataset, the best results on both titles and full-texts are obtained by MLP. The title-based method is on par with the full-text method when eight times as many titles as full-texts are used. When all

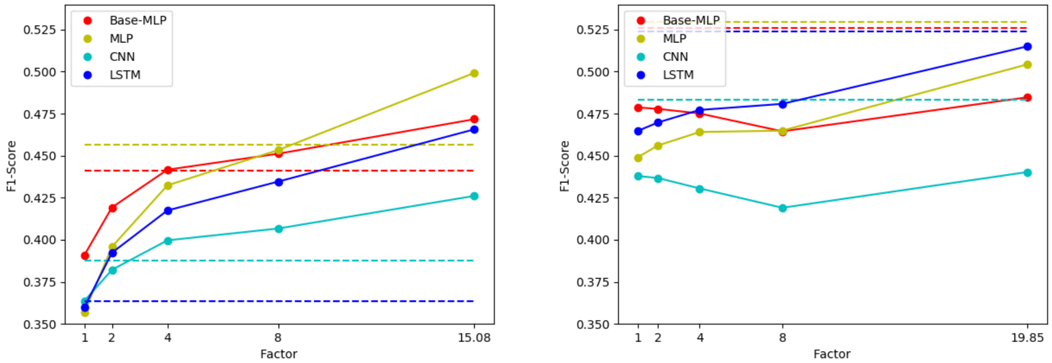
[47] This pretrained model can be downloaded at https://nlp.stanford.edu/projects/glove/.

[48] https://www.tensorflow.org/

[49] To increase the reproducibility of our study, we made the source code and the title datasets available at https://github.com/florianmai/Quadflor.

Table 8: Results of experiments in terms of sample-based F_1-measure. The best performing method on each sub-dataset is printed in bold font.

Data	Full-Text	T1	T2	T4	T8	T$_{all}$
			EconBiz F_1 scores			
Base-MLP	0.441	**0.391**	**0.419**	**0.442**	0.451	0.472
MLP	**0.457**	0.357	0.396	0.432	**0.453**	**0.500**
CNN	0.387	0.364	0.382	0.400	0.407	0.426
LSTM	0.363	0.360	0.392	0.417	0.435	0.466
			PubMed F_1 scores			
Base-MLP	0.526	**0.479**	**0.478**	0.475	0.465	0.485
MLP	**0.530**	0.449	0.456	0.464	0.465	0.504
CNN	0.483	0.438	0.437	0.431	0.419	0.440
LSTM	0.524	0.465	0.470	**0.477**	**0.481**	**0.515**



(a) Sample-averaged F1-scores with respect to varying sample size on the EconBiz dataset.

(b) Sample-averaged F1-scores with respect to varying sample size on the PubMed dataset.

Figure 10: The figures show the performance of each classifier on titles as a function of the sample size relative to the number of full-texts as a solid line on EconBiz (a) and PubMed (b). The dashed horizontal lines represent the respective classifier's performance on the full-text.

titles are used, the title-MLP outperforms its full-text counterpart by 9.4%, achieving an F_1-score of 0.500. In contrast, when the same number of samples for full-text and titles is used, the gap between the best title method (Base-MLP) and the best full-text method (MLP) is 16.9% in favor of the full-text.

The MLP seems to benefit the most from additional titles. Initially, when as many titles as full-texts are used for training, all our proposed methods, i. e., MLP, CNN, and LSTM, perform within 0.007 points in F_1-score from each other, but MLP performs worst. However, the relation flips as more titles are added. In Figure 10 (a), we can observe that MLP has the steepest curve of improvement out of all methods, in particular when considering the improvement from T1 to T2. With twice as many titles as full-texts, MLP is already the best performing classifier out of the ones we have proposed. The gap to the other methods only gets wider as more data is added for training. Overall, the MLPs performance on T$_{all}$ improves over the performance on T1 by 40.1%. The other methods also improve continuously as more training data is added. However, the CNN and LSTM improve by only 17% and 29.4% with respect to T1, respectively. Still, this is enough to surpass their full-texts counterparts by 10.1% and 28.4%, respectively.

When using as many titles as full-texts, Base-MLP outperforms our proposed methods clearly by 0.027 points in F_1-score. However, Base-MLP does not benefit as much from additional training data. Its overall improvement is only 20.7%, so when all titles are used for training, it is outperformed by MLP by a margin of 5.9%. On the full-text, MLP has an advantage of 3.6% over the baseline. Yet, the baseline still has a large advantage over LSTM and CNN, both on titles and full-texts.

On PubMed, MLP shows the best full-text performance, whereas LSTM yields the best results on titles. However, even when all titles are used for training, that is almost 20 times as many titles as full-texts, the LSTM still shows a F_1-score 2.9% lower than the full-text MLP. However, this is a considerably smaller gap than when the same number of samples are used. Base-MLP, the best performing method on PubMed T1, achieves 10.7% lower scores than the best method on PubMed Full, which is MLP.

The MLP and the LSTM show similar behavior when more training data is added. As shown in Figure 10 (b), the plots of the MLP and LSTM are almost parallel. However, the overall improvement from T1 to T$_{all}$ is slightly higher for the MLP than for the LSTM. The former improves by 12.3%, whereas the latter improves by 10.8%. The CNN does not seem to benefit from more data at all. Initially, on T1, the CNN performs close to the other methods, scoring just 5.7% less than the LSTM. With more training data, the CNN demonstrates a worse classification performance than with fewer training samples. Only when all available titles are used, the CNN barely outperforms itself on T1 by a slim margin of 0.5%. Consequently, the CNN has the largest difference to its full-text counterpart out of all proposed methods. It scores 9.8% lower on T$_{all}$ than on PubMed Full, whereas the gap is 5.2% for the MLP and 1.8% for the LSTM.

By a considerable margin, the baseline is the best method on T1, where few samples are used for training. Despite using 20 times as many training samples, the performance on T$_{all}$ is only 1.3% better, which is a difference to the MLP and LSTM of 3.9% and 6.2%, respectively. As it is the case on the EconBiz dataset, the full-text performance of BaseMLP is the second best and gets as close to the MLP as 0.8%.

Discussion The problem addressed in this section is to which extent title-based methods can perform similarly to full-text-based methods by increasing the amount of title training data. On EconBiz, the best title-based method outperforms the best full-text method by 9.4% when all title training data is used. Considering that the difference is 16.9% in favor of the full-text when the sample sizes are equal, this is an impressive improvement. On PubMed, the improvement is less astounding. However, the title-based method is close to the full-texts, as the difference in score is small (less than 3%). Considering that the gap is much larger for equal sample sizes (10.7%), we must acknowledge that current machine learning techniques in combination with large quantities of data are able to obtain just as good classification performance by merely using the titles.

However, in order to utilise title-based methods in a particular application, it is important to understand why there is such a large difference between the EconBiz and PubMed datasets regarding the benefit of employing title-based methods with large amounts of data. A possible explanation for that difference lies in the absolute numbers of full-texts available for training. As we have pointed out, previous literature suggests that deep learning models require around 650,000 samples to outperform more traditional approaches. On EconBiz, this number of full-texts is far from being reached. Due to this lack of enough training data, our deep learning models may not do so well with full-texts, in absolute numbers. The models based on titles on the other hand may be able to achieve their impressive results because there is just enough data to unleash the power of deep learning models. In fact, our MLP, which was optimised for large sample sizes, starts to outperform the baseline when eight times as many titles as full-texts are used, which nets to approximately 560,000 training samples. On the PubMed dataset, there are almost 650,000 full-text samples available. Here, the deep learning models can already work well on full-text. This may explain why the LSTM performs so much better on PubMed's full-texts than on EconBiz's full-texts. These findings support the claim from previous literature (X. Zhang et al., 2015) that deep learning models work well for text classification only when the sample size is several hundred thousands. Furthermore, since our datasets have large label spaces, we can state that this observation extents to XMLC, which is arguably a harder task than single-label classification.

In order to push the limits of text classification based on titles, our strategy was to develop and employ methods that can make use of the vast amount of data available for training. Our results suggest that this strategy was largely successful. On both datasets, some of our methods surpass the performance of the baseline as they are given more and more data for training. BaseMLP on the other hand cannot make such good use of the additional training data. This becomes particularly clear on the PubMed dataset, where it improves by only 1.3% even when it has 20 times more training data. Our methods on the other hand improve considerably the more data is used for training. To be fair, part of the much larger gain compared to the baseline is due to overfitting on the small title datasets such as T1, given that our methods are optimised towards their performance on T$_{all}$. This design decision was necessary to fairly assess the development of the performance as the sample size increases. We observed that the capacity of the resulting models is likely too large for smaller datasets, which results in overfitting. This can be seen by the fact that BaseMLP, which has considerably lower capacity, outperforms our proposed methods on both PubMed and EconBiz. Yet, on T$_{all}$, MLP outperforms Base-MLP by a wide margin on both datasets. LSTM is close to Base-MLP on EconBiz and outperforms it drastically on PubMed. Again, this indicates the success of our strategy. The only exception is the CNN, which does not benefit as much from additional data as our other proposed methods, although its capacity is large compared to other architectures recently proposed in the literature. It is particularly interesting that the performance of CNN (and Base-MLP, too) actually drop on PubMed as more title samples are used for training. We explain this by the fact that in T2 to T8 more than 1,300 new labels that do not occur in

T1 are introduced. Hence, the models have to account for these new labels even though they never appear in the test set, reducing their capacity to learn to classify the labels relevant to the test set.

Considering the amount of attention CNNs have received in recent years for their performance in text classification, the results of our CNN is rather underwhelming. This is true for both full-text, and titles. In neither case is this due to overfitting. Our preliminary experiments showed that the CNN benefits from increasing the feature map size a lot. Yet, CNNs seem to not benefit as much from more training data as LSTMs and MLPs do. There is no observable benefit on the PubMed dataset. On EconBiz, the rate of improvement is comparable to MLP and LSTM up to a factor of four times the number of full-texts. After that, the improvement is marginal.

Although, we do not focus on achieving results beyond the state-of-the-art performance on, e. g., full-texts, our proposed models do enhance other proposed approaches for XMLC. As described in Section 3.6.2, the MLP is at its core a non-linear version of the popular fastText. The employed CNN is based on recent advances from the domain of text-based XMLC. Its performance improves by integrating more fine-grained window sizes and larger feature maps. We present a strong bidirectional LSTM with attention over the outputs that does not assume a hierarchical structure of the document and, thus, also works for short text snippets in contrast to previous work by Yang et al. (Yang et al., 2016). Therefore, our methods are good candidates for researchers to adopt also for single-label text classification.

A common problem in machine learning, and in deep learning in particular, is that models are very sensitive to the choice of hyperparameters. However, examining the whole hyperparameter space is very difficult due to its combinatorial complexity. Recently, this has called the validity of deep learning results into question, for example in language-modeling (Melis, Dyer, & Blunsom, 2017) or even text classification (Le et al., 2018). This problem persists in our study as well. However, instead of simply assuming values for our parameters or manually tuning them in a somewhat arbitrary fashion, we took an incremental tuning approach that in the end led to an improvement over initial base models from the literature in all cases. This gives us reason to believe that our results are largely reliable.

Although we have compared three deep learning methods, we did not compare against linear models such as logistic regression, and we did not compare against other non-linear approaches such as kNN or SVMs. However, traditional linear methods are inferior to non-linear ones when the training data is large. More importantly, we compared against a non-linear baseline by Galke et al. (Galke et al., 2017) that was shown to outperform not only linear models, but also other non-linear, non-parametric models like kNN and SVMs on a diverse set of datasets and by a wide margin on both titles and full-texts.

For comparability, our proposed models were chosen such that they can be employed to titles and full-texts uniformly. This prevents us to fully exploit certain strengths of the full-text. For instance, full-text models might benefit greatly from a hierarchical model as proposed by Yang et al. (Yang et al., 2016). On the other hand, we tried our best to tap the full potential of full-texts. For instance, in our CNN we employ dynamic max-pooling with $p = 3$ for full-texts although this does not have any beneficial effect on titles.

We have examined two datasets from digital libraries of scientific content. We expect that these results generalise to other datasets of scientific publications as well. Consistently to previous text classification research, we found our deep learning methods to require about 550,000 samples to outperform the previous baseline (EconBiz T8). While this number of titles can certainly be reached in domains other than economics and biomedicine, not many scientific domains will reach this number of full-texts. This can be seen by considering the fact that the availability of full-texts is generally tied to their open access rate. The rate of open access journals in academia is approximately 7% as reported by Teplitskiy et al. (Teplitskiy, Lu, & Duede, 2017). This number closely matches the rate of 5 to 6.5% of available full-texts in our datasets. Moreover, Teplitskiy et al.'s study also suggests that the corpus of publications from the medical domain are among the largest. Therefore, we think it is likely that in other domains at least the relatively small gap of less than 3% between titles and full-texts can be achieved as well. However, in many other domains where only few full-texts are available, training models on titles may actually be much better, as we demonstrated for the economics domain.

In the MOVING platform the titles are always available, while the full-text is accessible for a limited number of documents because of legal barriers. Algorithms that produce good annotation by using the title as textual input instead of the full-text is desirable. This is because the title is always present in bibliographic data and free to use in text mining applications, whereas the full-text and even the abstract are often either not available or automatic processing is legally restricted. For instance, in EconBiz, which is included in the MOVING platform, few documents have an abstract (15%) or full-text (7%) available. Furthermore, the task of downloading and processing the full-text is cumbersome, where one has tens of gigabytes of data. In contrast, as Table 7 shows, titles get along with a couple of gigabytes of space. Thus, the full-text data is by an order of magnitude larger.

In conclusion, we demonstrated that in a realistic scenario deep learning algorithms are able to fulfill the demand for sufficiently strong title-based classification methods. Thus, it is possible shifting the focus from full-text-based classification to title-based classification to maximise the applicability of fully automated subject indexing systems.

3.7 Video processing

In this section we describe two different video analysis methods: a temporal fragmentation and annotation method for lecture videos using their transcripts (Section 3.7.1), applied to VideoLectures.NET[50] videos, and a new machine learning technique for dimensionality reduction and concept annotation with complex concept labels for large-scale datasets (Section 3.7.2).

3.7.1 Transcript-based lecture video fragmentation

3.7.1.1 Problem statement

In deliverable D3.2 (Vagliano et al., 2018) a transcript-based lecture video fragmentation method was briefly described. In this deliverable we describe the developed method in more detail and we extend it by (i) integrating a keyword extraction procedure and (ii) introducing a new large-scale dataset of artificially-generated lectures. Moreover, we report the experimental results of the developed method in this dataset.

3.7.1.2 Method description

As described in deliverable D3.2 (Vagliano et al., 2018), the lecture video fragmentation method utilises the textual information derived from a transcript, to extract meaningful textual cues, which are phrases or terms that the original text contains. These cues are characteristic of the original text; they capture very concisely the essence and the meaning of that text. The transcript text is used as input to our method, which outputs a set of time boundaries of the video fragments. The textual parts of transcripts are processed in order to extract meaningful textual cues. Two different methods for cue extraction are developed. The first method is a state-of-the-art work in video lecture segmentation that uses Noun Phrases as cues (Shah et al., 2015). The second one is based on the textual analysis and textual decomposition component of an ad-hoc video search system (Markatopoulou, Galanopoulos, Mezaris, & Patras, 2017). Then, these cues are vectorised in a way that each textual part is represented as a single vector. Again, two different approaches are developed for transforming the extracted cues in a vector space. Finally, to fragment each lecture video, a sliding-window-based method is used in order to detect time boundaries. These boundaries define the final set of temporal video fragments.

Text processing and cue extraction A transcript is a sequence of text parts, each one being followed by the start and end time of the corresponding spoken content in the video's audio track. Standard Natural Language Processing (NLP) techniques are used in order to process the transcripts text. Text cleaning methods, such as stop-word removal, punctuation and tag cleaning are applied, followed by text lowercase conversion, in order to reduce vocabulary size. Consequently, the Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003) is used for part of speech tag extraction and the Stanford Named Entity Recogniser (Finkel, Grenager, & Manning, 2005b) for named entity extraction (e.g. names, organisations, etc.). These tags are used to find cue phases and words that can encapsulate the information of a text part. Two different approaches are examined to this end. The first approach is the method of (Shah et al., 2015), based on which Noun Phrases (NP) are extracted from the available text. A "noun phrase" is basically a noun, plus all the words that surround and modify the noun, such as adjectives, relative clauses and prepositional phrases. The motivation behind choosing to examine this method is that in (Lin, Chau, Cao, & Nunamaker Jr, 2005) the performance of several different textual features was examined, and it was shown that NP performance is better than the one of other textual feature extraction methods. The second approach we examine is inspired from the query analysis and decomposition method of an ad-hoc video search (AVS) system (Markatopoulou et al., 2017). Specifically, in (Markatopoulou et al., 2017) task-specific NLP rules are used in order to extract textual cues from a text part. For example, "Noun-Verb-Noun" sequences are searched for in the text. Such a triad can encapsulate more information than one word by itself. Both the above approaches produce a set of words or phrases $C = [c_1, c_2, \dots, c_t]$, where t is the number of extracted cues in a textual part, which characterises this particular part.

[50] http://videolectures.net/

Cue representation The extracted cues are represented in a vector space, and for that two different representations are adopted.

First, a Bag-of-words approach with an N-gram language model, which uses the extracted cues as sequences of the model. For a specific part of text, the TF-IDF weighting of the cues C extracted form this part of text is calculated, to produce a vector $\mathbf{V}^C_{BoW} = [v_{c_1}, v_{c_2}, \dots, v_{c_d}] \in \mathbb{R}^d$, where d is the total number of distinct cues in the whole transcript, i.e. the dictionary of the language model.

As a cue representation alternative, Word2Vec (Mikolov et al., 2013) is utilised, a state-of-the-art neural-network-based word embedding method that transforms words into a semantic vector space. Word2Vec represents every word w_i of a phrase or other piece of text as a continuous vector $\mathbf{V}^{w_i}_{word2vec} = [v_1, v_2, \dots, v_n]$ in a low dimensional space $\mathbb{R}^n$, which captures lexical and semantic properties of words. As global representation of a text part, $\mathbf{V}^C_{word2vec}$, the *average word vector* approach is followed, which averages the vectors of each word of each cue that has been extracted from this text part.

Each one of the aforementioned approaches results in a vector that represents a specific part of text, making the comparison of text parts easy.

Video fragmentation A sliding window (W_i) of N words moves across the entire text of a transcript with a certain step of $N/6$ words. On each step the similarity between two neighboring windows (W_i, W_{i+1}) is calculated. For each sliding window we follow the cue extraction process which is described above and each window is represented as a set of cues, C_i and C_{i+1} respectively. For each window a vector $\mathbf{V}^C$ is calculated using one of the two approaches described in the cue representation subsection above. Finally, the cosine similarity (Shah et al., 2015) is utilised to calculate the similarity between two neighbor windows. The result of the similarity calculation across the entire transcript is an one-dimensional signal $y = f(x)$ is produced, where x represents time and the y represents two neighboring windows similarity score. We determined the valleys and the peaks of this graph and the deepest valleys are assigned as candidates for segment boundary. The depth of a valley $depth_{val}$ is calculated based on the distances from the peaks on both sides of the valley. Then, a fixed number of k valleys with the largest $depth_{val}$ can be selected as the boundaries of the fragments.

Keyword Extraction Once the lecture video fragments are calculated, our method extracts a set of related keywords for every fragment. For this purpose, two different methods are utilised. Firstly, a simple Bag of Word (BoW) method is used; each fragment is considered as an independent document and the Noun Phrases (NP) text processing previously described is followed to build the dictionary from the calculated cue phrases. Then, for each cue phrase the TF-IDF weight is calculated and the 10 cue phrases with the higher score in TF-IDF weighting terms, are selected as the most relevant, to the fragment, keywords.

Moreover, we developed an updated version of the keyword extraction procedure, in order to eliminate a number of issues that are presented in the above approach. More specifically, we introduced several rules to handle Automatic Speech Recognition (ASR) system failures (due to the noisy environment for example) and transcripts tags (e.g. ”[SILENCE]“ ”[HESITATE]“). For example, when the ASR system outputs multiple and consecutive instances of the same word (e.g. ”same same same same“), then these words are ignored. Furthermore, in order to find the most meaningful keywords in a fragment, we choose to ignore cue phrases that appear in the majority of the lectures. We find the 200 most frequent words in a corpus of 17k lecture videos, and we built a new stop-word list from them, in order to ignore these words in the keyword extraction procedure. For example, words like ”number“, ”work“, ”good“ or ”idea“ occur several times in almost every lecture, and they can not be considered as meaningful keywords. Finally, we added a lemmatisation procedure in order to avoid the appearance of the same word, in several inflected forms, in the extracted keywords.

In Figure 11, we show an example of a video lecture with a subset of the calculated fragments and the extracted keywords derived from the two different procedures.

3.7.1.3 Experimental evaluation and comparison

Dataset An important problem in the development and evaluation of video lecture fragmentation methods is the lack of annotated datasets, due to the difficultly and the time-consuming nature of manual annotation. Moreover, constructing such datasets is a difficult task due to the subjectiveness of defining fragment boundaries. In most of the cases it is not clear where exactly a fragment boundary exists, even to the lecturer. Thus, in a 1-2 hour lecture, where the transcripts of free continuous speech of a speaker are available, the fragmentation results will be quite arbitrary even if coming from a human expert.

To overcome this problem, we choose to follow an approach well-known in the document segmentation field. Following (Brants, Chen, & Tsochantaridis, 2002), in which document fragments of various lengths

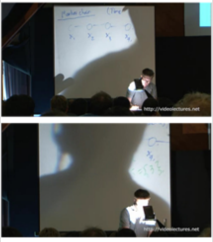
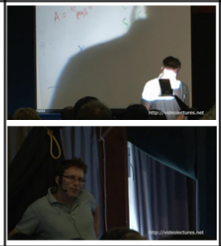
Lecture Video Title : “**Graphical Models and message-passing algorithms**”				
Fragments	**Keyframes**	**ASR generated transcripts**	**Extracted Keywords (old method)**	**Extracted Keywords (updated method)**
Fragment # 01 **00:00:00,000 -> 00:09:49,240**		[…] so i'll be talking about graphical models, i guess some aspects of graphical models appear up there are things like, graphical models trees, bounds reweighted kelsey, we'll see, algorithm will see many of these things in these tutorial talks, so the plan is the, first talk today is gonna be somewhat higher level just giving the basics of graphical, models is given you feel what they can be used for, all start drilling down a little bit towards the end of the lecture today, and then tomorrow we get more in detail and message passing and also what are, known as variational methods […]	• graphical models • clique • vertices • graphs • subset • lecture • history • random variables • structure • graph	• graphical model • subset • vertex • clique • history • structure • edge • graph • lecture • dependency
Fragment # 02 **00:09:49,240 -> 00:16:37,340**		[…] that's one way of defining markov chains right you say that a process that if, you know the president then everything else in the past tells you nothing more about, the future, right but this is a simple instance of this more general property that i'm about, to introduce because this is the cuts that this cuts split the graph into two, pieces the past and the future and what you have is the conditional independence property, the, and the notation we use for this we say that x of a that's the, random variables in the past is conditionally independent of x of, i think i messed up my notation i was using […]	• graph • markov • vertex • cutset • variables • chain • notation • random variables • pieces • parts	• graph • property • chain • notation • markov • vertex • piece • theoretic property • speed • position
Fragment # 03 **00:16:37,340 -> 00:18:37,730**		[…] the maximal cliques for this guy, the edges of the maximal cliques there's nothing else, so what this would say for this markov chain, he would say that you've got a distribution over four variables, and i would say that it should factorize this easy that i'm writing this just, normalisation termites to make the probability distribution normalize would say this should factorizes in this way, right so go to clique on one two, and you've got to clique on three four, right so it's saying that you can sort of choose three functions, here i done one for every maximal clique, and the functions depend only on two variables, the whole distribution depends on […]	• markov • speech • chain • recognition • distribution • function • variables • maximal cliques • factorizations • cliques	• function • number • maximal clique • distribution • termite • pairwise term • maximal clique guy • creek • binary variable parent • chain

Figure 11: Example of a lecture video where video frames, the corresponding speech transcripts, the calculated fragments and the extracted keywords are illustrated.

were concatenated and formed new documents, we have created a new dataset[51] of artificially-generated lectures. We used 1,498 transcript files from the world's biggest academic online video repository, the VideoLectures.NET. These transcripts correspond to lectures from various fields of science, such as Computer science, Mathematics, Medicine, Politics, etc. We split all transcripts in random fragments, the duration of which ranges between 4 and 8 minutes. A synthetic lecture is then created by combining exactly 20 randomly-selected parts. The first 300 such artificially-generated lectures were chosen for assembling our test dataset. Each such lecture file has a mean duration of about 120 minutes, and the overall dataset contains about 600 hours of artificially-generated lectures. Every pair of consecutive fragments in these lectures originally comes from different videos, consequently the point in time where such two fragments are joined is a known ground-truth fragment boundary. All these boundaries form the dataset's ground truth. We should stress that we do not generate the corresponding video files for the artificially-generated lectures (only the transcripts) and we do not use in any way the visual modality for finding the fragments.

Results In Table 9, the experimental results are presented. We evaluate the combination of the two cue extraction methods i) NP and ii) AVS, with the two different representations i) BoW and ii) Word2vec embeddings. We evaluate the performance of our method using a fixed number of 19 calculated fragment boundaries per video, which means we produce exactly 20 fragments for every artificially-generated video lecture. We also measure the performance of our system while the window size N changes.

Our method is compared with two competitive works. The first one is the transcript based lecture video fragmentation of (Shah et al., 2015), which is actually identical to the BoW-NP combination of our experiments setup, with a fixed window size of 120 words. Moreover, we compare with the supervised text segmentation method of (Koshorek et al., 2018).

[51] Large-scale video lecture dataset and ground truth fragmentation available at `https://github.com/bmezaris/lecture_video_fragmentation`

Table 9: Experimental results (Precision, Recall and F-Score) of the three variations of the proposed approach, and comparison with (Shah et al., 2015) using different text window sizes.

Method	Measure	Window size (N)								
		120	240	360	480	600	720	840	960	1080
BoW NP(Shah et al., 2015)	Precision	0.287	0.228	0.204	0.262	0.349	0.415	**0.426**	0.408	0.391
	Recall	0.315	0.251	0.224	0.288	0.383	0.455	**0.459**	0.422	0.378
	F-Score	0.3	0.239	0.213	0.274	0.365	0.434	**0.442**	0.414	0.383
	Avg Num_of_Cues	20.78	42.03	63.15	84.28	105.30	126.18	146.98	167.68	188.25
	Fragment duration mean	330.5	330.5	330.5	330.5	330.7	332.0	338.6	354.0	380.1
	Fragment duration std	16.4	16.4	16.4	16.4	16.6	17.0	22.6	32.5	40.7
BoW AVS	Precision	0.27	0.281	0.287	0.315	0.365	0.398	**0.415**	0.386	0.377
	Recall	0.297	0.309	0.316	0.346	0.401	0.437	**0.455**	0.416	0.383
	F-Score	0.283	0.294	0.301	0.33	0.382	0.416	**0.434**	0.4	0.379
	Avg Num_of_Cues	17.31	34.93	52.49	70.13	87.54	104.95	122.22	139.42	156.57
	Fragment duration mean	330.5	330.5	330.5	330.5	330.5	330.6	332.1	338.6	361.0
	Fragment duration std	16.4	16.4	16.4	16.4	16.4	16.4	16.8	22.7	34.0
Word2Vec NP	Precision	0.335	0.248	0.252	0.29	0.373	0.427	**0.465**	0.448	0.427
	Recall	0.368	0.273	0.278	0.319	0.411	0.466	**0.491**	0.437	0.377
	F-Score	0.351	0.26	0.264	0.304	0.391	0.446	**0.477**	0.441	0.398
	Avg Num_of_Cues	20.77	42.07	63.20	84.28	105.30	126.18	146.98	167.68	188.25
	Fragment duration mean	330.5	330.5	330.5	330.5	330.6	333.0	345.3	375.3	417.1
	Fragment duration std	16.4	16.4	16.4	16.4	16.5	17.6	30.7	44.3	54.2
Word2Vec AVS	Precision	0.268	0.289	0.299	0.321	0.373	0.412	**0.425**	0.417	0.402
	Recall	0.295	0.318	0.329	0.353	0.41	0.453	**0.46**	0.423	0.377
	F-Score	0.281	0.303	0.313	0.336	0.391	0.431	**0.441**	0.419	0.388
	Avg Num_of_Cues	17.35	34.99	52.60	70.13	87.54	104.95	122.22	139.42	156.57
	Fragment duration mean	330.5	330.5	330.5	330.5	330.5	330.9	336.4	359.8	392.7
	Fragment duration std	16.4	16.4	16.4	16.4	16.4	17.0	22.1	36.8	52.2

Moreover, in Table 10 the developed methods, using the best-performing window size from Table 9, are compared with (Shah et al., 2015) and (Koshorek et al., 2018). As shown in Table 10, the best overall performance was achieved by the combination of the text analysis using Noun Phrases and Word2Vec representation.

Table 10: Experimental comparison of the three variations of our lecture video fragmentation method, for the most appropriate window size (Table 9), with (Shah et al., 2015) and the supervised segmentation method of (Koshorek et al., 2018).

Measure	Method				
	BoW AVS	Word2Vec NP	Word2Vec AVS	BoW AVS(Shah et al., 2015)	Supervised segmentation(Koshorek et al., 2018)
Precision	0.415	**0.465**	0.425	0.426	0.237
Recall	0.455	**0.491**	0.46	0.459	0.393
F-Score	0.434	**0.477**	0.441	0.434	0.293

3.7.1.4 Implementation, APIs and integration

We developed the lecture video fragmentation REST service, which inputs the transcript file of a lecture video (in the `dfxp` file format), calculates the fragments boundaries and the corresponding keywords and finally outputs the results of the procedure in the JSON format. The communication between the user and the service is done via HTTP GET calls. This service is an external component of the MOVING platform and it is accessed via its REST API. To call the lecture video fragmentation REST service a GET request is issued: `GET http://160.40.51.36:8000/LTF?file=<dfpx_file_URL>`, where the parameter <dfpx_file_URL>is the URL link where the `dfxp` file is stored.

The output of the service is a JSON file containing the following fields:

Slug The slug name of the lecture

Video The video number of the lecture

Duration The duration of the lecture video

Number of Fragments The number of the calculated fragments

Fragments An array containing the list of the calculated fragments; each fragment contains 4 fields:

id The identifier of the fragment

Start time The time the fragment begins

End time The time the fragment ends

Keywords An array containing the top-10 extracted keywords of each fragment in descanting order; each keyword contains 2 fields:

Keyword The keyword itself

Relevance The relevance score between the keyword and the corresponding fragment

Lecture video keywords An array that contains the top-10 extracted keywords of the entire lecture video in descending order; each keyword contains 2 fields:

Keyword The keyword itself

Relevance The relevance score between the keyword and the entire lecture

An example is illustrated in Listing 3.

Listing 3: An example of an output from the lecture video fragmentation REST service

```
{
"Video": 1,
"Number_of_Fragments": 2,
"Fragments": [
        {"Keywords": [
                {       "keyword": "movie",
                        "relevance": 0.21},
                {       "keyword": "rating",
                        "relevance": 0.1902},
                {       "keyword": "datum",
                        "relevance": 0.1808},
                                .
                                .
                                .
                                ],
        "EndTime":              "00:10:03,620",
        "StartTime":    "00:00:00,000",
        "id": 1
        },
        {"Keywords": [
        {       "keyword": "exponential family",
                "relevance": 0.1377},
        {       "keyword": "artist",
                "relevance": 0.1377},
        {       "keyword": "case",
                "relevance": 0.1296},
                        .
                        .
                        .
                        ],
        "EndTime":      "00:17:32,330",
        "StartTime":    "00:10:03,620",
        "id": 2
        }
],

"Duration": "00:17:32,330",
"LectureVideoKeywords": [
        {       "keyword": "datum",
                "relevance": 0.1212},
        {       "keyword": "movie",
                "relevance": 0.0808},
        {       "keyword": "rating",
                "relevance": 0.0707},
                        .
                        .
                        .
                        ],
"Slug": "lms08_williamson_pmdc"
}
```

3.7.2 Concept Detection

3.7.2.1 Problem statement

In recent years, the amount of annotated video data is growing rapidly, e.g., in social media repositories such as ϒouTube, Facebook and Instagram. To this end, dimensionality reduction (DR) techniques are currently getting increasing attention, not only because they can significantly reduce the size of high-dimensional observations, which is beneficial to applications with specified requirements, e.g., for energy consumption, memory size, latency and network bandwidth, but also due to the fact that when used in combination with traditional classifiers, these approaches can effectively deal with the curse of dimensionality problem (Hou, Nie, Yi, & Tao, 2015). Linear discriminant analysis (LDA) is one of the most powerful supervised learning methods for DR (Belhumeur, Hespanha, & Kriegman, 1997; Howland & Park, 2006). It solves an eigenvalue optimisation problem (Golub & Loan, 2013) to identify a linear transformation that preserves class separability. The computed transformation matrix is then used to project the observation data from the input space to a much lower-dimensional subspace. However, LDA, similarly to most of the traditional learning approaches, appear formidable challenges to deal successfully with large-scale datasets. More specifically, when the size of the training data matrix exceeds the memory size of the computing system it is infeasible to apply LDA. Furthermore, due to this limitation, LDA cannot benefit from the abundance of the existing large training corpora in order to further improve its performance as, for example, is shown in (Banko & Brill, 2001) for traditional learning approaches. Despite the importance of this research direction, the extension of LDA for large-scale applications is almost an unexplored topic. An exception to this is the spectral regression discriminant analysis (SRDA) presented in (Cai, He, & Han, 2008). This method, uses spectral graph analysis in order to cast the eigenproblem (EP) of LDA into a least squares (LS) regression framework, thus, facilitating the computation of the transformation matrix iteratively. However, SRDA requires centred data, which may not be the optimal normalisation for the data distributions involved in certain applications. Additionally, the data centring required during both the training and testing phases of SRDA increases the computational cost and round-off errors, which have a negative effect in efficiency and classification accuracy. Most importantly, although LS-based learning models have shown quite good performance in regression analysis, they appear severe drawbacks in classification problems, such as the lack of robustness to outliers and the low classification performance when combined with gradient-based optimisation (Bishop, 2006). To alleviate the drawbacks described above, we propose a new DR method called logistic regression discriminant analysis (LRDA). LRDA utilises a novel factorisation framework of the scatter matrices involved in LDA in order to effectively transform the EP to an equivalent linear system of equations, and subsequently employs logistic regression and the stochastic gradient descent algorithm to obtain the required linear transformation. Experimental results in the YouTube-8M benchmark (Abu-El-Haija et al., 2016), the largest public multi-label video dataset, confirm the excellent performance of the proposed method.

3.7.2.2 Method description

Let $\mathbf{X}$, $\mathbf{Y}$ be the training and associated class-indicator matrix of $c = 2$ classes, denoted as ω_1, ω_2, and $n = \sum_{i=1}^{C} n_i$ observations $\mathbf{x}_j$ in the input space $\mathbb{R}^f$,

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{f \times n}, \tag{5}$$

$$\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{c \times n}, \tag{6}$$

where, n_i is the number of observations of class i, f is the dimensionality of the input space, and $\mathbf{y}_j = [y_{1,j}, y_{2,j}]^T$ is the indicator vector associated with $\mathbf{x}_j$, i.e., $y_{i,j} = 1$ if $\mathbf{x}_j \in \omega_i$ and $y_{i,j} = 0$ otherwise. LDA seeks the linear transformation $\psi \in \mathbb{R}^f$, satisfying the following EP

$$\mathbf{D}\psi = \lambda \mathbf{S}\psi, \tag{7}$$

where, $\mathbf{D}$, $\mathbf{S}$ are the between- and total-scatter matrix,

$$\begin{aligned} \mathbf{D} &= \sum_{i=1}^{C} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T, && (8) \\ \mathbf{S} &= \sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{m})(\mathbf{x}_j - \mathbf{m})^T, && (9) \end{aligned}$$

$\mathbf{m}_i = \frac{1}{n_i}\sum_{j\in\omega_i}\mathbf{x}_j$, $\mathbf{m} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j$ are the estimated sample mean of class i and total sample mean, respectively, and λ is the eigenvalue of the EP. This optimisation problem can be solved by employing appropriate methods for symmetric EPs, such as the symmetric QR algorithm, or SVD-based approaches in case that the matrix $\mathbf{S}$ is singular (Belhumeur et al., 1997; Howland & Park, 2006; Golub & Loan, 2013). However, for large scale datasets most of the current LDA approaches are not applicable due to their very high memory requirements. To this end, as we explain in the following, we reformulate the EP in Equation 7 to an equivalent logistic regression optimisation problem and employ stochastic gradient descent to solve it. The scatter matrices in Equations 8 and 9 can be factorised as follows

$$\begin{aligned} \mathbf{D} &= \sum_{i=1}^{c} n_i \mathbf{m}_i \mathbf{m}_i^T - \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{c} n_i n_j \mathbf{m}_i \mathbf{m}_j^T \\ &= \mathbf{M}(\mathbf{N} - \frac{1}{n}\tilde{\mathbf{N}})\mathbf{M}^T \;= \mathbf{X}\mathbf{C}\mathbf{X}^T, && (10) \\ \mathbf{S} &= \sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^T - \frac{1}{n}\sum_{j=1}^{n}\sum_{\kappa=1}^{n} \mathbf{x}_j \mathbf{x}_\kappa^T \\ &= \mathbf{X}\mathbf{X}^T - \frac{1}{n}\mathbf{X}\mathbf{J}\mathbf{X}^T \;= \mathbf{X}\mathbf{Q}\mathbf{X}^T, && (11) \end{aligned}$$

where,

$$\begin{aligned} \mathbf{M} &= [\mathbf{m}_1, \mathbf{m}_2], && (12) \\ \mathbf{C} &= \mathbf{Y}^T\dot{\mathbf{N}}^{-1}\mathbf{O}\dot{\mathbf{N}}^{-1}\mathbf{Y}, && (13) \\ \mathbf{Q} &= \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T, && (14) \\ \mathbf{O} &= \mathbf{I} - \check{\mathbf{N}}, && (15) \\ \mathbf{n} &= [n_1, n_2]^T, && (16) \\ \mathbf{N} &= \mathrm{diag}(n_1, n_2), && (17) \\ \tilde{\mathbf{N}} &= \mathbf{n}\mathbf{n}^T, && (18) \\ \dot{\mathbf{n}} &= [\sqrt{n_1}, \sqrt{n_2}]^T, && (19) \\ \dot{\mathbf{N}} &= \mathrm{diag}(\sqrt{n_1}, \sqrt{n_2}), && (20) \\ \check{\mathbf{N}} &= n^{-1}\dot{\mathbf{n}}\dot{\mathbf{n}}^T, && (21) \end{aligned}$$

and $\mathbf{1}$ is the all-one vector. Let (λ, γ) be the nonzero eigenpairs (NZEP) of $\mathbf{O}$ (Equation 15) and ψ be the solution of the linear system

$$\mathbf{X}^T\psi = \theta, \qquad (22)$$

where,

$$\theta = [\overbrace{\theta_1, \ldots, \theta_1}^{n_1}, \overbrace{\theta_2, \ldots, \theta_2}^{n_2}]^T, \qquad (23)$$

and $\theta_1 = \sqrt{n_2/(n_1 n)}$, $\theta_2 = -\sqrt{n_1/(n_2 n)}$. Then, as we show in (Gkalelis & Mezaris, 2018), assuming that the linear system in Equation 22 is consistent, (λ, ψ) is the NZEP of the EP (Equation 7). For large-scale trainings sets, where the training data matrix $\mathbf{X}$ is too large to fit in the memory, we need to resort to iterative solutions in order to solve Equation 22. To this end, we reformulate Equation 22 into an unconstrained empirical loss minimisation problem, as shown in Equation 24,

$$\arg\min_{\psi} J, \; J = \sum_{j=1}^{n} l_j + \frac{\alpha}{2}\|\psi\|^2, \qquad (24)$$

where α is the penalty regularisation parameter and l_j is the instant logistic loss for $\mathbf{x}_j$, and the minibatch stochastic gradient descent algorithm is then applied to estimate ψ.

3.7.2.3 Experimental evaluation and comparison

The proposed LRDA is compared against the state-of-the-art SRDA in the YouTube-8M benchmark (Abu-El-Haija et al., 2016). This dataset contains more than 8 milion videos annotated with complex concept labels. It is currently the largest public multi-label video benchmark for content-based learning approaches. In order to permit their comparison, both the DR methods are further combined with a binary linear support vector machine (Vapnik, 1998) operating in the DR subspace. The evaluation is performed by directly utilizing the training/validation divisions as provided in (Abu-El-Haija et al., 2016), and the performance of each method is measured using the mean average precision (MAP) (Robertson, 2016). In average, we have observed a more than 5 % MAP improvement of LRDA over SRDA. In Table 11, the results for 15 concepts are depicted. The identification numbers and associated names of these concepts are shown in Table 12.

Table 11: Evaluation results on YouTube-8M

Method								**ID**							
	0	*91*	*133*	*630*	*1000*	*1312*	*1975*	*2080*	*2106*	*2217*	*2316*	*2373*	*2608*	*3088*	*3188*
LRDA	94%	81%	50%	68%	27%	21%	62%	24%	65%	36%	36%	13%	34%	18%	7%
SRDA	94%	66%	38%	67%	25%	7%	61%	7%	57%	17%	22%	12%	28%	14%	5%

Table 12: Identification numbers and names for 15 concepts

ID	**Name**
0	Game
91	Highlight film
133	Comedy (drama)
630	Sitcom
1000	Brake
1312	Haruhi Suzumiya
1975	Rock fishing
2080	Tow truck
2106	Gorilla
2217	Killer Instinct
2316	Sprite (computer graphics)
2373	Indoor cycling
2608	Rocket launch
3088	Wine tasting
3188	Tinker Bell

For most concepts, LRDA outperforms SRDA for more than 10 % in MAP rate (e.g., see concept IDs 91, 133, 1312, 2080, 2106, 2217, 2316). It seems that the above concepts are quite complex to separate in the input space, and possible include a significant portion of outliers, and for this reason, SRDA fails to derive an adequate discriminant subspace. On the other hand, for some concepts the two methods perform equivalently. In these experiments, we assume that the associated concepts are either well separated from the rest of the concepts in the input space (IDs 0, 630, 1000, 1975), or in contrary, the feature vector observations carry little discriminant information for any of the two methods to achieve a satisfying solution (ID 3188).

4 User logging and data analysis dashboard

As described in deliverable D3.2 (Vagliano et al., 2018), the MOVING platform is currently instrumented to capture interaction data from users, and the WevQuery interaction dashboard allows stakeholders to create custom queries to consider users' interaction in their modules. Newly implemented modules relying on interaction data sometimes require additional information about users' behaviour, which can only be provided through the creation of new interaction events or more comprehensive interaction context for the captured information (see Section 4.1). As previously mentioned in D3.2, WevQuery has been extended with pattern mining capabilities. However, the perceived usefulness of adding further analysis functionalities remained to be evaluated. Section 4.2 presents a user study where twenty participants performed typical analyses tasks with WevQuery. The pattern mining extension has also been used to evaluate the Adaptive Training Support (ATS) module in the MOVING platform. These results will be presented in the upcoming deliverable D1.4: *Final implementation of user studies and evaluation*.

4.1 Interaction events

4.1.1 Problem statement

The ever-evolving nature of the project has posed two particular problems regarding user interaction capture. The first one is ensuring all new interfaces and changes comply with the necessary restrictions, so the captured interaction data contains relevant information and can be queried effectively. Information from the interaction events needs to support the needs of other modules, as well as the later analyses of user interaction, making user evaluations, like the one planned for deliverable D1.4, possible. Even when the interface is appropriately annotated for effective interaction capture, there can still be a need for behaviour information that requires the creation of new events. For example, interaction events tailored to the MOVING interface can help store particular information shown in the screen, so it can be used as context of the interaction taking place.

4.1.2 Interaction documentation

In order to allow UCIVIT (Apaolaza, Harper, & Jay, 2013) to capture interaction events effectively, all relevant interface elements need to have certain HTML attributes annotated. The most relevant attributes are the `ID` and the `class`. The `ID` is used as a unique identifier that allows the extraction of all interaction with a particular interface element. The `class` is used for elements that are not unique, but are still relevant, so extracting their interaction from the database is still possible.

The previous deliverable D3.2 (Vagliano et al., 2018) showed a table documenting the values set for the various interface elements from the MOVING platform. The purpose of this document is to make the extraction of events targeting different interface elements possible. Certain elements, such as the search bar text input, can be easily identified via a unique HTML `ID` attribute. For example, events describing a click on the search button can be retrieved by extracting all `mousedown` events on `search-button`.

Other elements are not unique, and are instead identified via their HTML `class` attribute. This is the case for the list of result items shown after carrying out a search. As planned in D3.2 (Vagliano et al., 2018), these elements now include the `ID` of the document shown as the result, so this information is used as context for the interaction. This information was initially planned to be used for the recommender system, as a way to determine which single documents are consumed by different users. However, the number of times each source is consumed by the users has been found useful to reduce the workload of Web crawlers. This way, if particular sources are not used often, the frequency of the tracker can be decreased.

New interfaces, such as the ones created for the MOOC and the learning environment also needed to be tested, and the corresponding annotations added. New modules, such as the recommender system, and the curriculum widget were annotated by their respective developers. These new annotations have been documented and will be included in the final version of the table in D1.4 as the final documentation for the interaction capture of the MOVING platform.

4.1.3 Additional events

Some modules required access to information specific to new changes made in MOVING. An important requirement for the *search history* module is being able to show the users how many documents were returned for each particular search. Also, as showing to the user all the search iterations carried out for each query would not be useful, a way of highlighting the last iteration of a series of searches was necessary. For example, a user could be looking for “machine learning”, but then add the filters “language” and “date”. Instead of showing to the user all the three searches, only the last one, along with its filter options and the number of results, would be shown. In order to be able to retrieve this information efficiently, a new event was created storing the value of the search options of the latest search carried out for particular keywords. As the interaction capture and the MOVING platform work independently (i.e. the interaction capture only has access to the information shown to the client), this event also extracts the information about the number of resulting documents from the interface, so it can be stored as additional context for the event in the interaction database.

New events have also been created to describe the interaction with the visualisations. As the visualisations shown in the MOVING platform are highly interactive, the events designed to describe interactions with a Web page were found not to be descriptive enough. Each visualisation triggers its own set of events describing interactions specific to them, such as selecting a set of nodes, or moving a node in the "concept graph" visualisation.

4.2 WevQuery

4.2.1 Problem statement

WevQuery (Apaolaza & Vigo, 2017) has been used in the MOVING platform to provide other members of the consortium the means to create their own REST queries to get access to the interaction data. These customised requests have already been used for modules such as the ATS to retrieve the number of times each search functionality have been used, and by the recommender system to retrieve all the search queries for individual users.

As described in D3.2, WevQuery was also extended to include pattern mining into the supported analyses workflows. A user study has been carried out to evaluate the trade-off between the usefulness and the complexity of this new feature. We found that, despite being more difficult to learn, assisted pattern mining was perceived as more useful and helps user researchers to acquire more actionable insights. Additionally, the new analyses capabilities of WevQuery have been used to support a remote evaluation of the ATS widget in the MOVING platform.

4.2.2 Pattern mining evaluation

In order to evaluate the trade-off between complexity and usefulness of the pattern mining functionality of WevQuery, we have designed two different analyses workflows of increasing complexity: assisted, and assisted++.

The **assisted** workflow allows user researchers to guide the execution of pattern mining algorithms by customising the event set to be used as input for the algorithms and iteratively add/remove custom events to refine the results, choosing the appropriate granularity of the events as they reformulate their hypotheses.

The **assisted++** workflow allows for the definition and testing of custom hypothesis of Web use in order to discover frequent and outlying behaviours. These analysis workflows support user researchers in formulating hypotheses that might be considered weak or could even be mere expectations. Nevertheless, these hypotheses serve as a starting point that inform the initial exploration of data. Then researchers can iterate from expectations to consolidated hypotheses, which can be tested in experiments and A/B tests.

Study Twenty participants (10 female, median age 29.5, SD=4.82, fifteen computer scientist, two psychologists, one business school student, one social scientist, and one telecommunications engineer) took part in a user study to evaluate the trade-off between the complexity of assisted pattern mining workflows and the knowledge acquired through their use. We asked participants to report their confidence about various topics on a range from 1 (unconfident) to 4 (confident). Participants' confidence of user experience (UX, median = 3, SD = 0.72, ▁▃█▆) and Web markup languages (median = 3, SD = 0.88, ▁▂▆█) was high, while their confidence of pattern mining techniques (median = 2, SD = 1.14, █▆█▆) was lower. Our sample represented individuals who are experienced in Web technologies and are knowledgeable about human factors on the Web but lack the skills to use pattern mining tools to conduct sophisticated analyses on the data. The participants performed the tasks in Table 13 using the functionalities described in the previous section. Following a think-aloud procedure (Ericsson & Simon, 1980), a researcher took notes of the feedback given by participants as well as any insight they reported while carrying out the tasks. Participants played the role of a user researcher who was willing to use pattern mining algorithms to get further insights into the usage of the user interface on a large website, but lacked the necessary data processing and pattern mining skills. Two months of interaction data from the School Of Computer Science website from University of Manchester [52] were used as stimuli of the

[52] http://www.cs.manchester.ac.uk

Table 13: Tasks given to the participants in the study

Workflow	Task	Prompt
Non-assisted	Guided	Work with a predefined set of events. Simulate the situation where an expert in data processing has extracted a series of sequences of events for you to analyse.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Run the Frequent Pattern Mining algorithm again, with the same input, but this time limit the minimum Support threshold to 40%.
Assisted	Guided	Generate your own dataset to use as input for pattern mining. Choose which events will be included (`mousedown`, `mouseup`, `mouseover`, and `mouseout`), as well as the user interface element upon which such events were triggered.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Run the Frequent Pattern Mining algorithm, but this time try to answer the following question. How many times do users click on an image AND a link during the same episode?
Assisted++	Guided	Test Hypothesis 1: Users hover over the same element for longer than 10 seconds, either because they are triggering an interactive event (such as disclosing dropdown dialogues) or as part of exploring the interface.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Test Hypothesis 2: Within 10 seconds from loading the page, users start the action of clicking on the search bar.

study, accounting for a total of 5.7m low-level events generated by 2,445 unique users. This website follows modern website standards, as is the home page of a school from a university that attracts thousands of visitors every month. The screenshot shown in Figure 12 was given to the participants so they would understand the structure of the page and the location of the various interface elements.

The tasks given were conducted on three different workflows. We used the two workflows mentioned above, and added a **non-assisted** workflow as the baseline whereby participants had to apply pattern mining techniques to a predefined event set which could not be further modified. This way we simulated the scenario where user researchers are provided with a pre-formatted input for the available interaction data. This workflow would be comparable to using a set of independent tools including the SPMF (Sequential Pattern Mining Framework) (Fournier-Viger et al., 2016) library. We integrated SPMF into our tool-supported workflows to simplify its use and keep consistency across the three options and enable comparisons between the non-assisted and assisted workflows.

Results Median completion times in directed tasks were 60 seconds (SD = 36) on the non-assisted workflow, 143 seconds on the assisted workflow (SD = 118) and 580 on the assisted++ workflow (SD = 235). Exploration tasks took a median time of 281 (SD = 149) seconds on the non-assisted, whereas accomplishing the exploratory tasks took users 290 (SD = 121) and 371 seconds (SD = 100) for the assisted and assisted++ workflow respectively. Longer completion times are observed in the exploratory tasks and the assisted workflows, which is confirmed by a one-way repeated-measures ANOVA, showing an effect of task on completion times F(5,95) = 44.09, $p < 0.0001$. A post-hoc Tukey cite10.2307/3001913 test indicates significant differences[53] on the directed tasks between the non-assisted and assisted ($p < 0.03$), non-assisted and assisted++ ($p < 0.0001$), and assisted and assisted++ ($p < 0.0001$).

When we compare the baseline and the two assisted workflows (assisted and assisted++) the USE (Usefulness, Satisfaction, and Ease of use) (Lund, 2001) questionnaire yields medians of 3.7 (SD = 0.66) and 3.2 (SD = 0.67) for ease of use on the non-assisted and the assisted workflows respectively, 4.1 (SD = 0.57) and 3.75 (SD = 0.85) for ease of leaning, 3.6 (SD = 0.56) and 3.7 (SD = 0.46) for satisfaction and 3.6 (SD = 0.63) and 3.8 (SD = 0.40) for usefulness. Paired t-tests (McDonald, 2009) on these usability qualities yields significant differences for usefulness (t(19) = -2.20, $p < 0.05$), ease of use (t(19) = 2.14, $p < 0.05$) and highly significant differences for ease of learning (t(19) = 3.84, $p < 0.01$) —see the distribution of values in Figure 13.

In directed tasks, the CBUQ (Component-based Usability Questionnaire) (Brinkman, Haakma, & Bouwhuis, 2009) for ease of use yields medians of 4.3 (SD = 0.56) on non-assisted tasks, 4.5 (SD = 0.56) on assisted tasks and 3.75 (SD = 0.80) on assisted++. As far as exploratory tasks are concerned, non-assisted tasks yield medians of 3.8 (SD = 0.56) and 4 (SD = 0.50) for assisted tasks and 3.5 (SD = 0.72) for assisted++. The boxplots in Figure 14 display the distribution of the values. A one-way repeated-measures ANOVA found a significant effect of type of task on ease of use, F(5,95) = 8.22, $p < 0.001$. Post-hoc Tukey tests show significant differences between the assisted and assisted++ workflows on exploratory ($p < 0.01$) and directed

[53] We do not report interactions between exploratory and directed tasks as they are of a different nature.

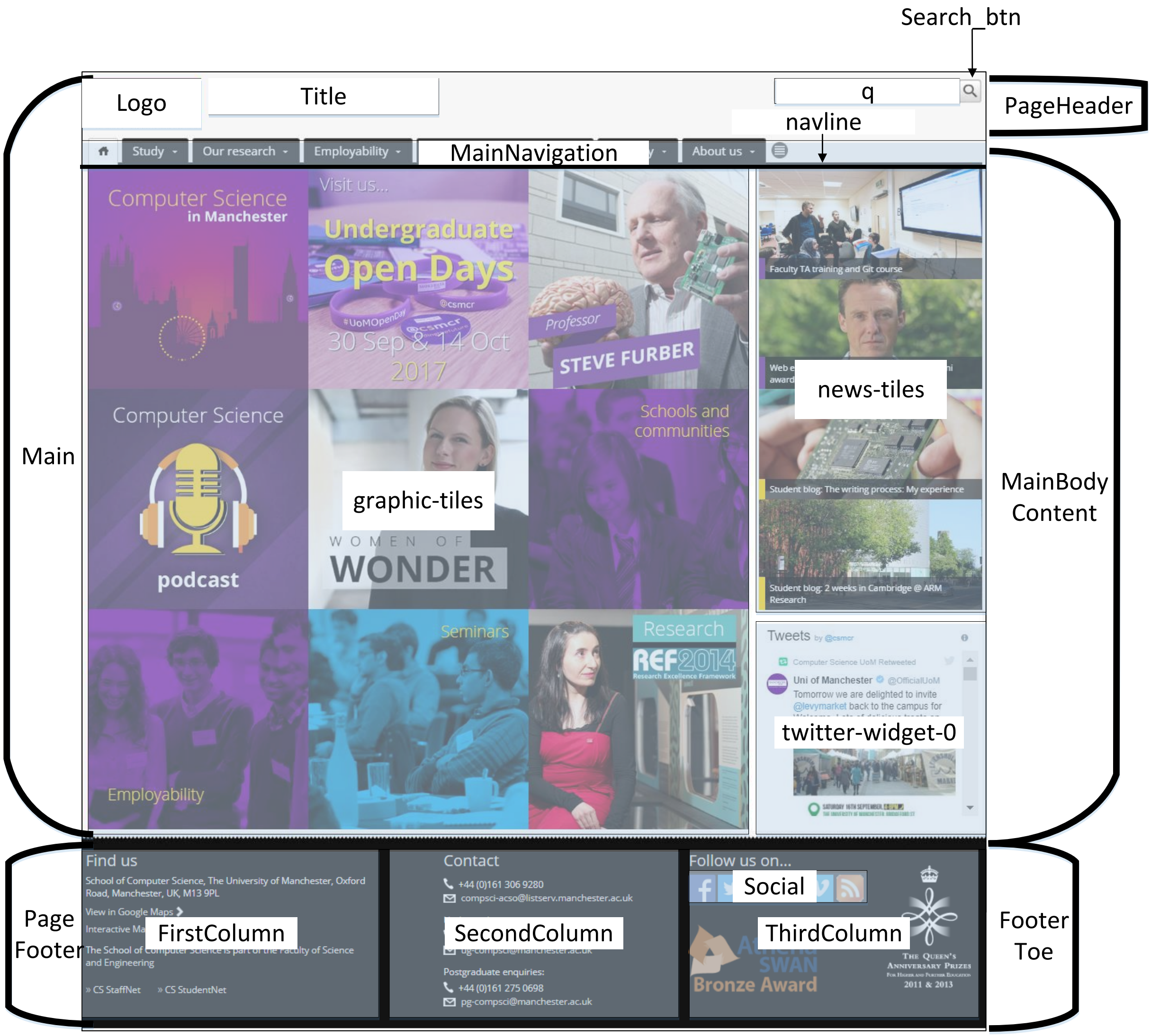


Figure 12: Screenshot of the Web page used for the study, annotated for the participans

tasks ($p < 0.001$). On directed tasks differences are significant between the non-assisted and assisted++ workflow.

All tasks get a median of 4 (i.e. *easy*) for perceived difficulty as measured with the PDAQ (Perceived Difficulty Assessment Questionnaire) (Ribeiro & Yarnal, 2010) except for those tasks executed in the assisted++ workflow, which yield a median of 3 (i.e. *fair*). There is again an effect of task on difficulty, as indicated by a one-way repeated-measures ANOVA, $F(5,95) = 8.96$, $p < 0.0001$. A post-hoc Tukey test indicates significant differences between the assisted++ and assisted workflow, and assisted++ and non-assisted workflow on directed tasks ($p < 0.0001$). Marginally significant differences are found ($p = 0.08$) between the two assisted workflows on exploratory tasks.

Discoveries Table 14 shows the discoveries made by the participants grouped by the workflow and the type of discovery: whether it was descriptive knowledge, inferred knowledge or the participant refined the current hypothesis. The types of discoveries map approximately to the learning objectives in Bloom's taxonomy for learning (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956): comprehension, analysis and synthesis. While we acknowledge other approaches to classify discoveries (Livingston, Rosenberg, & Buchanan, n.d.), our classification contemplates the formulation of new hypotheses.

Descriptive discoveries indicate a basic level of understanding of the output of the pattern mining algorithm, and users being able to distinguish the relevance of a pattern based on its frequency. Discoveries by *inference* suggest that the participants established links between the output of the pattern mining algorithms and particular behaviours exhibited on the Web page: e.g. clicks on the search text field might indicate that users are intending to use search functionalities. *Prospective hypotheses* were formulated when participants

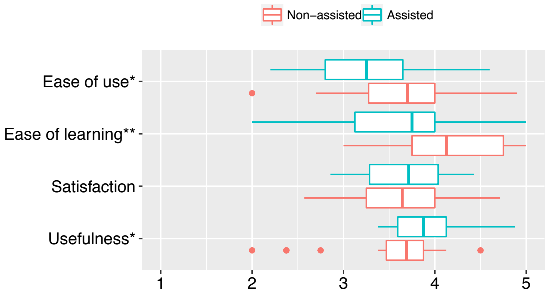


Figure 13: USE questionnaire: ease of use, ease of learning, satisfaction and usefulness of the workflows. Significance leves at *: $p < 0.05$, ** $p < 0.01$

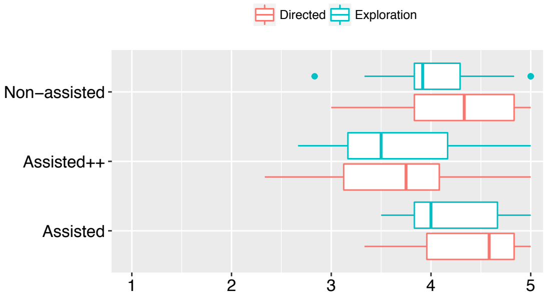


Figure 14: CBUQ questionnaire: ease of use of directed and exploratory tasks on the non-assisted and assisted workflows

gave possible explanations for the discovered behaviours, which could be used to guide the creation of new hypotheses to be then reintroduced into the analysis workflows. Participants came up with 100 instances of discoveries that corresponded to the descriptive category, 65 instances belonging to inference and 20 prospective hypotheses were formulated.

In the **non-assisted** workflow, the explored event set included the occurrence of all the available combinations of events and contexts. Participants were able to understand the output from the pattern mining but struggled to infer meaning from it. A total of nine participants were able to recognise the top of the page as the main point of interest for users after identifying the interface elements that got most interactions, and one participant realised small interface elements triggered greater number of hover events. Another participant realised that due to the nature of the page (a homepage providing access to other parts of the website) a mouse click would ordinarily indicate the end of the interaction, leading the user to a different Web page. This participant inferred the existence of intense mouse hovering activities would commonly take place before that click. Another participant hypothesised that users were trying to access navigation menus, while a last one assumed users were just exploring the Web page.

In the **assisted** workflow, the event set was filtered by selecting mouse click events on interface elements with a known ID, which drastically reduced the size of the output. Nineteen participants immediately noticed the high frequency of interactions with the element that had `q` as an ID. This element was a text input field to type the keyword to conduct searches on the website. Based on this information, half of the participants were able to link the mouse interaction on the `q` input field with the `search button` element (a button next to the mentioned text input field, that submits the query and triggers the search) and determined that users were using the "search" feature that was available on the Web page. Noticeable but less frequent interactions with other interface elements were also identified, such as the `title` and the `footer` elements of Web the page. The interpretation of the role of the remaining user interface elements was generally speculative (e.g. users clicking on `title` to go to the homepage). Eight participants realised that only a subset of clicks on the `q` text input field took place together with `search button`, and formed prospective hypotheses. Six participants suggested adding keyboard interactions to the analysis (e.g. *"maybe they just press "enter" after writing something in q"*), and two participants suggested that users were intending to search, but changed their mind afterwards. Finally, two participants considered the use of the search function as an indicator of bad design of the homepage: *"Users are not finding what they are looking for straight away. It is not visible or easy to find"*.

In the **assisted++** workflow, participants incorporated `hypothesis1`, as defined in Table 13, into the pattern mining analysis workflow. The analysed Web page contains interactive elements that disclose further information when hovered. For example, the main navigation bar contains a series of drop-down elements that when hovered, disclose a list of up to 45 links at once (see an example in Figure 15). This hypothesis is shaped by a hypothetical user researcher's expectations and prior knowledge about the Web page under study: some users hover those interactive elements for longer than 10 seconds, which is an abnormal behaviour worth exploring further. From the nineteen participants who explored the occurrences of `hypothesis1`, thirteen of them learned the relationship between `hypothesis1` and hyperlinks (defined as `link` elements in our system). Six participants also noticed multiple occurrences of `hypothesis1` within the same session and proposed possible explanations such as users reading and the existence of potential usability problems. Four participants took into account the nature of the analysed page and suggested that users could have been exploring the interactive elements, to then click on a link. Three participants pointed out repeated hovering activities before clicking on a link, and one of them suggested that the multiple occurrences of `hypothesis1` could also indicate

Table 14: Table of discovered knowledge. Participants made 100 discoveries belonging to the "Descriptive" category, 65 to the "Inference" category, and 20 to the "Prospective hypothesis" category

	Descriptive	**Inference**	**Prospective hypothesis**
Non-assisted	▪ Hovering `navline` is frequent (14) ▪ Hovering `pageHeader` is frequent (10) ▪ `mouseover` is the most frequent event (8) ▪ Hovering `image` is frequent (3) ▪ Hovering `graphic_tiles` is frequent (2) ▪ Hovering `title` is frequent (1)	▪ Most of the interaction is at the top (4) ▪ Menus at the top are used more (3) ▪ The header of the page is used more (2) ▪ Small items trigger more hover events (1)	▪ People exhibit hovering action before clicking (1) ▪ Users are trying to access navigation menu (1) ▪ Users are exploring the page (1)
Assisted	▪ Interaction with `q` is relevant (19) ▪ Clicks on `title` are frequent (8) ▪ Clicks on `footer` are frequent (4) ▪ Clicks on `google_maps` are frequent (3) ▪ Clicks on `contact` are not frequent (1)	▪ `q` element is relevant, users are searching (16), and as `q` and `search_btn` are connected, they are using the search dialog (10) ▪ Users click on `title` to go to homepage (4) ▪ Users look for the school's location (2) ▪ Users look for contact information (1) ▪ Users might copy text from `footer` (1) ▪ Users do not look for the email address (1)	▪ `q` and `search_btn` are only connected sometimes: users might be using the "enter" key (6), or it might indicate that users are not actually carrying out the search (2) ▪ People searching might indicate that they cannot find what they want at first, does this indicate bad design? (2)
Assisted++	▪ Notice frequency of `hypothesis1` (19) ▪ Clicks on `link` are frequent (7) ▪ Clicks on `image` are frequent (1)	▪ Noticed the relation between `hypothesis1` and clicking on `link` (13), or other element (1) ▪ Noticed several `hypothesis1` taking place in the same episode: users might be reading something (2), users might be waiting for the page to do something (2), there might be particularly slow users (1), or the interface might not be intuitive enough (1)	▪ Users might hover an element to disclose information to then click on a disclosed `link` (4) ▪ Repeated hovering behaviour (`hypothesis1`) followed by a click on `link` (3). It might Indicate users have been exploring several menus till they found the information they were looking for (1)

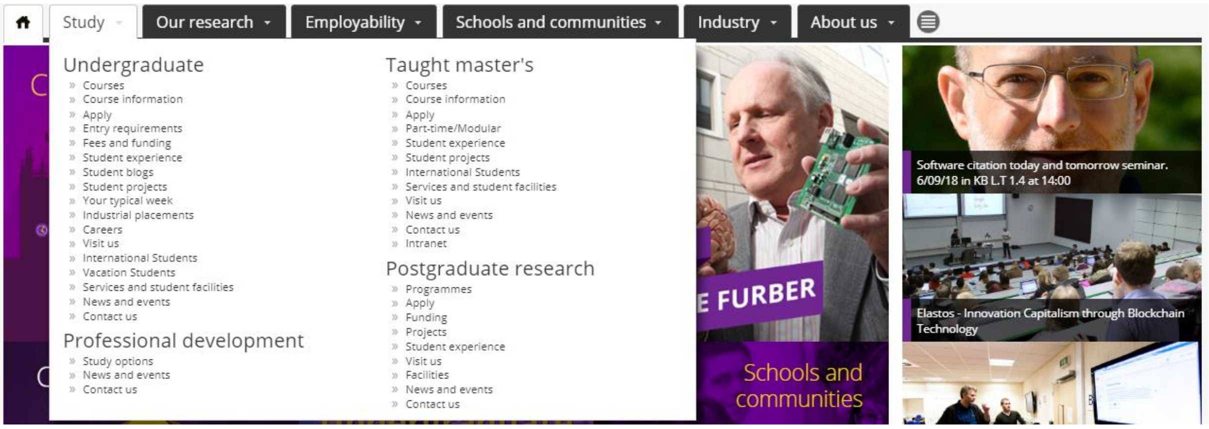


Figure 15: Example of interaction when hovering one of the menus in the Web page.

that users had to traverse more than one menu (in a hierarchical menu) before finding the information they were looking for.

Conclusions We found that user researchers can discover actionable knowledge from low-level Web log data provided that functionalities for data wrangling and data mining remove the complexity around these tasks. Our study suggests that while a baseline system (non-assisted workflow) does not prevent this from happening, tool support (assisted workflows) facilitates higher order knowledge discoveries. The perceived difficulty of the assisted workflows is counterbalanced by both the perceived usefulness and the higher number of actionable knowledge discoveries.

4.2.3 Longitudinal evaluation

WevQuery has been employed to carry out a remote study of the ATS module. The pattern mining functionality enabled the discovery of emerging behaviours from the participants. This study will be followed by a more extensive remote evaluation of the MOVING platform, carried out using UCIVIT and WevQuery. The results of these studies will be presented in the deliverable D1.4.

5 Visualisation technologies

The MOVING platform's search engine provides users with a list of search results. It is a known fact that "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 approaches are commonly used to support finding valuable information in search results, as "the human visual system has enormous power to perceive information from visualised data" (Ware, 2012).

Currently, the following visualisations are included in the MOVING Platform:

1. *Concept Graph* - for the discovery and exploration of relationships between documents and their properties.
2. *uRank* - a tool for the interest-driven exploration of search results.
3. Tag Cloud - a visualisation for the analysis of keyword frequency in the retrieved documents.
4. Top Properties - a bar chart displaying aggregated information about the properties of the retrieved documents.

The visualisations in the MOVING platform can be accessed over the same tab-menu, as the result list. This can be seen in Figure 16, with the *Concept Graph* being currently selected. In the rest of this section, we describe these visualisations with a focus on new features included since the previous deliverable D.3.2 (Vagliano et al., 2018). The section concludes with a user evaluation of the two major visualisations, *Concept Graph* and uRank, applied to the two MOVING use cases: young researcher and auditors. These use cases are described in the previous deliverable D1.1 (Bienia et al., 2017).

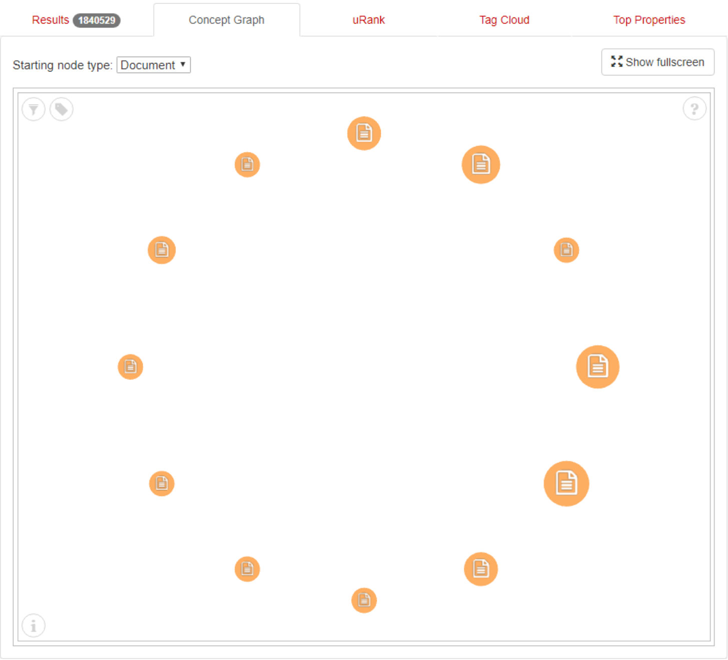


Figure 16: A preview of the initial display of nodes in the *Concept Graph*. The user can switch to other visualisations of the retrieved data (including the result list) by clicking on their respective tabs.

5.1 Concept Graph: an interactive network visualisation

5.1.1 Problem statement

GVF (Graph Visualisation Framework) is a web-based framework designed to support interactive analysis of large, complex networks consisting of documents, topical concepts, authors, venues, locations and other named entities, which are connected by relationships arising from co-occurrences, hierarchies, discourses, reading orders etc. To ensure scalability and provide smooth animated transitions, GVF is implemented using WebGL[54]-based rendering.

[54] https://www.khronos.org/registry/webgl/specs/latest/, last accessed: 12/12/2018

Node-link diagrams are commonly 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. computed by a force-directed placement algorithm (Fruchterman & Reingold, 1991), and the geometry of the links, which are typically straight lines, but may also be optimised by methods such as edge bundling (Holten & van Wijk, 2009). Different types of entities and relations as well as metadata can be visualised through different visual channels of nodes and links (e.g. color, shape, size/thickness etc.).

The basic rendering engine of the GVF was developed as part of the AFEL project[55] 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. For this purpose, the framework had to be specifically tailored to the common data model provided by MOVING's search engine (see Section 2). Those adaptations included explicitly defining and implementing the design of nodes and edges. Furthermore, statistical information and various interactions supporting the explorative analysis of the graph were added, resulting in the *Concept Graph* visualisation.

We focus on visual representation of metadata and 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.

5.1.2 User interface

Figure 17 shows the previous version of the *Concept Graph* from D.3.2 (Vagliano et al., 2018). In addition to the displayed document nodes, the old version only had an indicator for the number of displayed nodes (i) and a button which opened a dialog with a help text (ii).

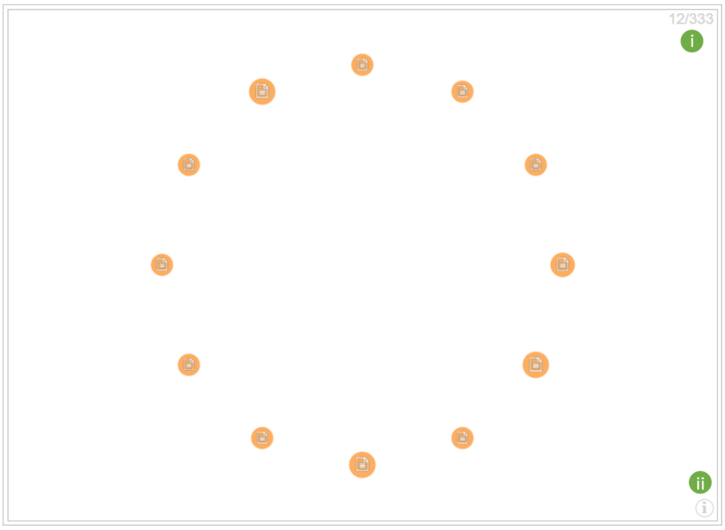


Figure 17: Concept Graph version from D3.2 (Vagliano et al., 2018): i) the progress indicator, which shows the number of displayed nodes out of all possible nodes for a fully expanded graph, ii) help text button, clicking on it opens a dialog with instructions.

Figure 18 shows the current version of the *Concept Graph*. The initial layout of the search results, represented as document nodes placed around a circle, remained the same as in the previous version. The default node type is *Document*, as before. The new version made improvements in the user interface and added new functionality to simplify the exploration process. In the drop-down list on top of the visualisation (A), the starting node type can be changed to any of the other available node types, such as *Author*, *Publication Year*, *Affiliation*, *Keyword* and *Concept*. Inside the visualisation container it is now possible to apply various filters to the graph (B), turn the labels on and off for all visible nodes (C) or inspect the properties of the currently displayed graph (D). While the number of displayed nodes from the previous version moved inside the properties dialog, which will be described later, the help text button remained but was moved to the top right (E). The label for the help text button was also changed from `i` to a question mark, since the `i` label was more fitting to be used for the graph properties button.

[55] `http://afel-project.eu`

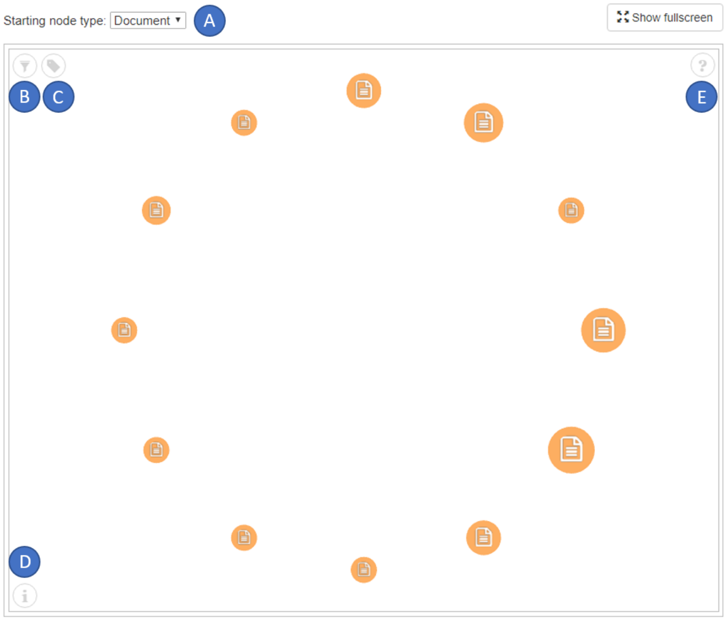


Figure 18: New user interface elements in the *Concept Graph*: A) starting node type drop-down list, B) node filter menu button, C) node labels toggle button, D) graph properties button, E) help text button.

Starting node type By default, the retrieved information will initially be displayed as *Document* nodes in the *Concept Graph*. The order in which the nodes are displayed is determined by the relevance score given by the search engine. The graph can then be expanded by clicking on the visible nodes and revealing the connected properties. However, sometimes it might prove more useful to start from other node types that represent the properties of the documents (authors, publication years, etc.). For this purpose, it is possible to change the starting node type in the pull-down list placed on top-left, just above the visualisation container, i. e. (A) in Figure 18. Since the same properties occur in multiple different retrieved documents, the relevance score of a single document cannot be used to infer the relevance of a property node. Therefore, the number of connections a property node is used as its relevance. This can be easily seen in Figure 19 where the largest node (with most connections) is on top, while the size gets smaller counter-clockwise.

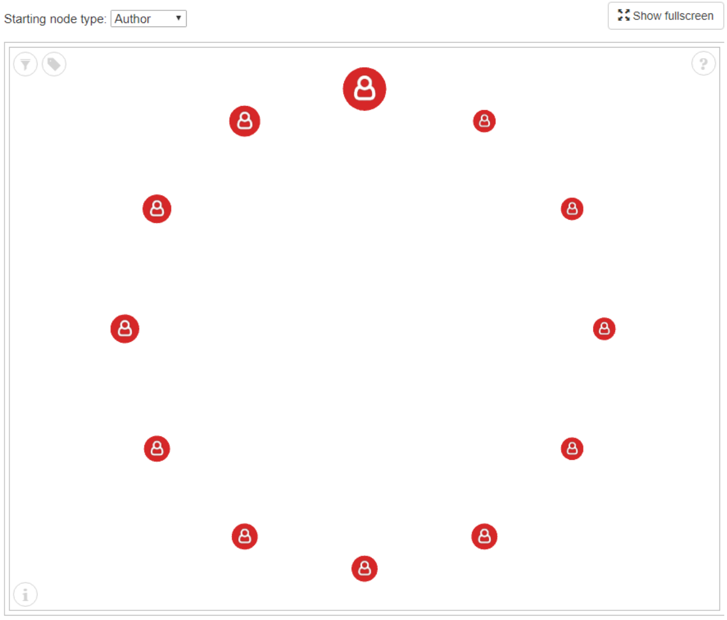
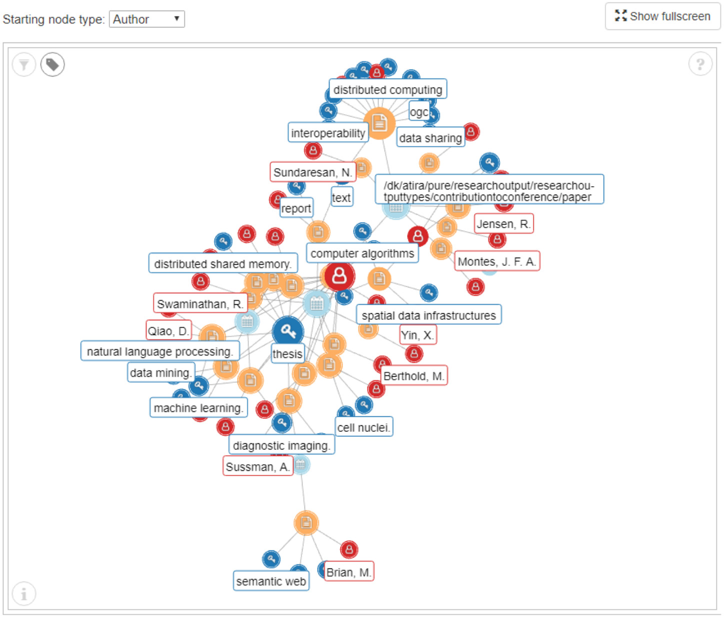


Figure 19: Initial node layout in the *Concept Graph* with *Author* as the starting node type.

Label toggle In the previous version of the *Concept Graph*, the user had to hover with the mouse over each individual node to get additional information. While the structure of the graph remains clear this way, this is obviously an inefficient and tedious way of exploration when inspecting multiple nodes. Therefore, it is now possible to show or hide all the node labels by clicking on the toggle label button (see C in Figure 18). Since the graph can be very dense, only the names of the nodes are displayed when the label toggle is enabled. Hence, hovering over a node still has the benefit of displaying additional information, e.g. the type of a document, or what other nodes are connected to this node. To avoid clutter caused by too many overlapping labels, only a limited number of labels are displayed on the screen at once. The order in which the labels are displayed is prioritised by node type and by node relevance. At first, the labels of the nodes containing topical information are shown. Those are the nodes of the type *Concept* and *Keyword*. Following that, the labels of the *Affiliation*, *Author* and *Publication Year* nodes are presented. The labels for the *Document* nodes are depicted at last, considering they might use up a lot of space in the visualisation container, due to the possibly long document titles. Most importantly, if the next label to be displayed overlaps with an already visible one, then it will not be drawn at all as this would lead to both labels being unreadable. To reveal such a hidden label, the user can either hover with the cursor over the node to temporarily reveal it, zoom in to provide a greater free area on the screen, or drag the node to an area with more space available. Figure 20 depicts the activated labels and the label prioritisation for the sub-graph that was created by further expanding the largest author node of Figure 19. Note that, in the current version, label positioning is not yet optimised to minimise overlap with the nodes.



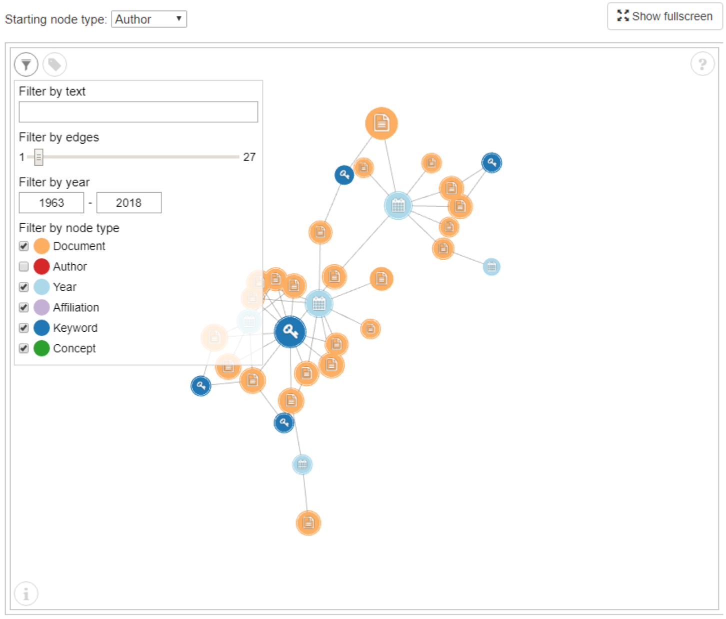
Starting node type: Author

Figure 20: Activated labels and label prioritisation for a sub-graph

Graph filters Another new functionality in the *Concept Graph* is the option to filter the visible nodes. Clicking on the filter button (see B in Figure 18) in the visualisation container reveals the node filter dialog. The nodes of the graph can be filtered in the following four ways:

1. *Filter by text* - only the nodes containing the entered text remain visible
2. *Filter by edges* - only the nodes which have at least the requested number of edges remain visible
3. *Filter by year* - only the nodes associated with a year in the specified range remain visible (this applies only to nodes of the type *Document* and *Publication year*)
4. *Filter by node type* - only the selected node types are visible

In the case that multiple filters are set, only the nodes satisfying all selected filters will be visible. Figure 21 shows the filtered sub-graph of Figure 20. In this case, visible nodes have at least two edges and cannot be of type *Author*.


Starting node type: Author

Filter by text

Filter by edges: 1 – 27

Filter by year: 1963 - 2018

Filter by node type
- ☑ ● Document
- □ ● Author
- ☑ ● Year
- ☑ ● Affiliation
- ☑ ● Keyword
- ☑ ● Concept

Figure 21: Activated filter for a sub-graph: nodes must have at least two edges and cannot be of type *Author*

Graph properties The last new addition to the user interface of the *Concept Graph* are the graph properties. Clicking on the graph properties button (see D in Figure 18) opens a small dialog that reveals the following information about the currently displayed graph(Figure 22):

1. *Total number of nodes* - it shows the total number of nodes that the completely expanded graph would have.

2. *Displayed number of nodes* - it shows the number of currently visible nodes.

3. *Total number of edges* - it shows the total number of edges that the completely expanded graph would have.

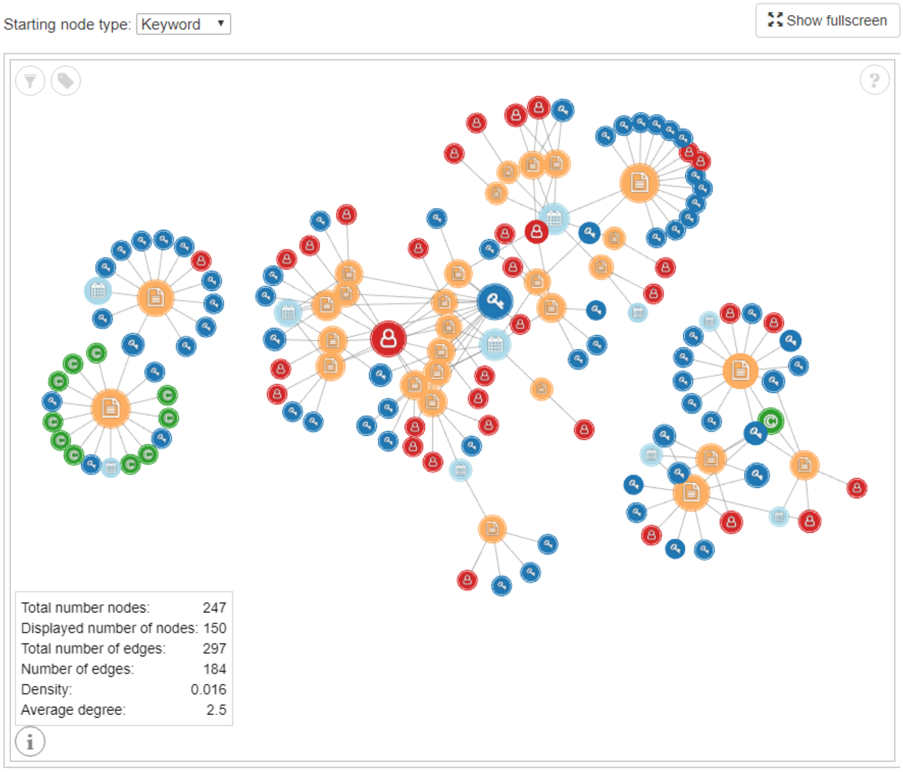
4. *Displayed number of edges* - it shows the number of currently visible edges.

5. *Density* - it represents the number of existing edges over all possible edges in a graph (Coleman & Moré, 1983). The higher the number is, the more interconnected the graph is. If this number is 1, the graph is fully connected. In case of Figure 22, the graph is not very densely connected.

6. *Average degree* - The average number of edges each node has (Feige, 2006).

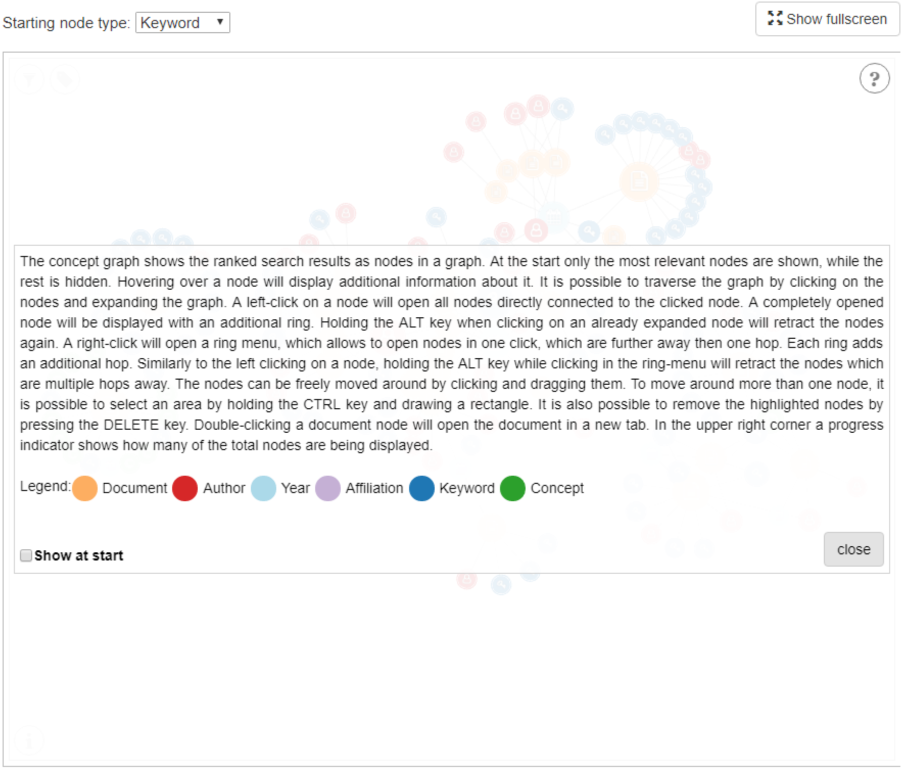
Help text The button for accessing the help text dialog has been moved from the bottom right (Figure 17, ii) to the top right (Figure 18, E) to be more visible. Moreover, the icon was changed to a question mark, indicating help. When the user accesses the *Concept Graph* for the first time, the help text dialog will be displayed automatically. The dialog now spans the whole visualisation container and, in addition to the text describing the interactions, includes a legend showing colors assigned to different node types (Figure 23).

Additional interactions A prolonged exploration process in the *Concept Graph* will sooner or later result in a great number of nodes that are not relevant anymore to the user. This is especially pronounced when expanding a whole neighborhood of a node (i.e. nodes connected over multiple hops) using the nodes ring menu, as shown in Figure 24. Therefore, interactions for reducing the number of displayed nodes were added. The first interaction, allows the user to select the nodes which are of no interest anymore using the area selection tool, and then to remove them from the graph by pressing the `Delete` key (Figure 25). In this way, the user can explicitly pick out the nodes which should be removed. The second interaction allows the user to 'collapse' a node. Holding the `Alt` key pressed and clicking on an already expanded node collapses all the directly connected nodes and remove them from the visualisation (Figure 26). This node collapsing functionality can also be used with the ring-menu to collapse a whole sub-graph (i.e. nodes connected over multiple hops).



Total number nodes:	247
Displayed number of nodes:	150
Total number of edges:	297
Number of edges:	184
Density:	0.016
Average degree:	2.5

Figure 22: The properties dialog reveals additional information about the currently displayed graph.



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 completely opened node will be displayed with an additional ring. Holding the ALT key when clicking on an already expanded node will retract the nodes again. 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. Similarly to the left clicking on a node, holding the ALT key while clicking in the ring-menu will retract the nodes which are multiple hops away. 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. It is also possible to remove the highlighted nodes by pressing the DELETE key. 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.

Legend: ● Document ● Author ● Year ● Affiliation ● Keyword ● Concept

☐ **Show at start**

Figure 23: The help text contains a description of the graph, the possible interactions and a legend showing colors assigned to different node types.

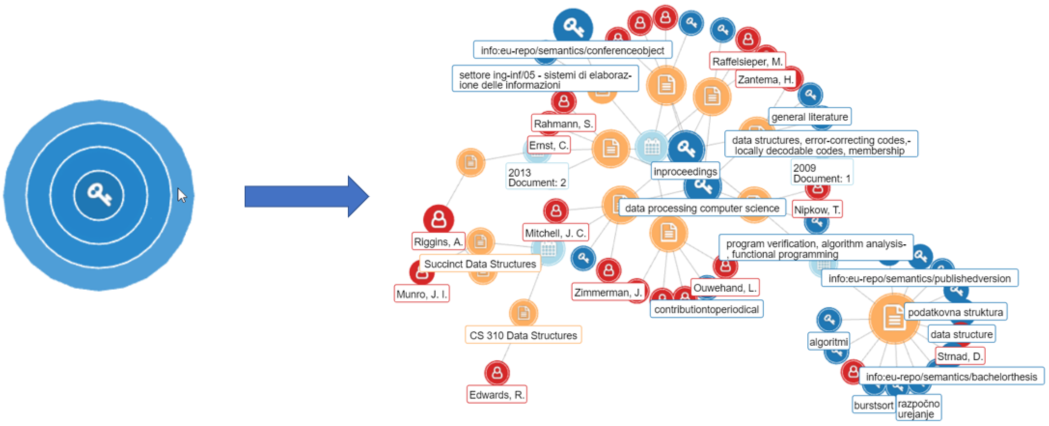

info:eu-repo/semantics/conferenceobject
settore ing-inf/05 - sistemi di elaborazione delle informazioni
Raffelsieper, M.
Zantema, H.
general literature
Rahmann, S.
Ernst, C.
data structures, error-correcting codes,- locally decodable codes, membership
inproceedings
2013 Document: 2
2009 Document: 1
data processing computer science
Nipkow, T.
Riggins, A.
Mitchell, J. C.
program verification, algorithm analysis- , functional programming
Succinct Data Structures
info:eu-repo/semantics/publishedversion
Munro, J. I.
Zimmerman, J.
Ouwehand, L.
contributiontoperiodical
podatkovna struktura
data structure
algoritmi
CS 310 Data Structures
Strnad, D.
info:eu-repo/semantics/bachelorthesis
Edwards, R.
burstsort
razpočno urejanje

Figure 24: Clicking on the third ring in the ring-menu of a node will create a sub-graph by showing all the nodes, which are connected to this node over 4 hops

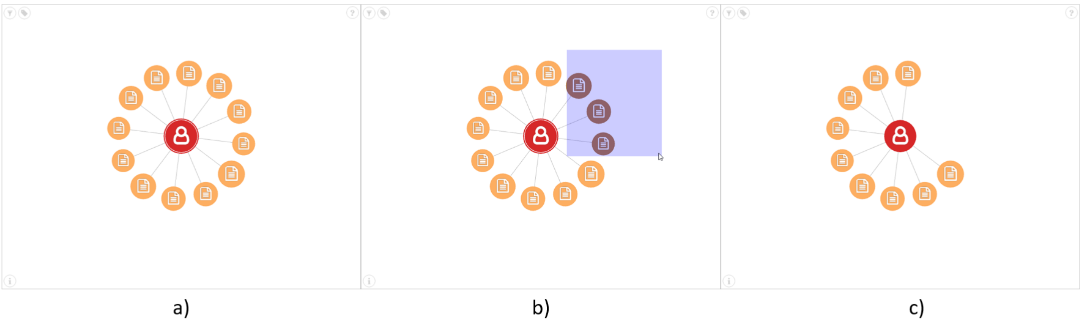


Figure 25: Removing nodes with area selection: a) initial layout of an expanded node, b) selection of nodes which should be removed, c) pressing the Delete key removes the selected nodes

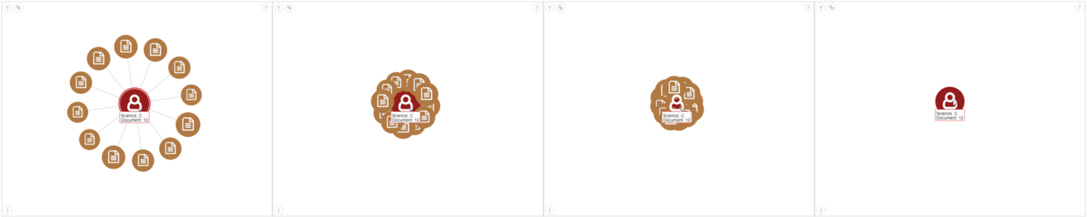


Figure 26: Clicking on a node while holding the Alt key pressed will collapse all the directly connected nodes and remove them from the visualisation. The 'collapse' animation, being the inverse of the 'expand' animation, will move the collapsed nodes towards the originating node, where they will finally disappear.

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 taken into account. 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 user interface displays a list of documents along with the extracted collection-keywords shown in a tag-cloud-like view.
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 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 visualizing document scores related to each keyword.
7. The users can select a single document to access more detailed information about it.

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

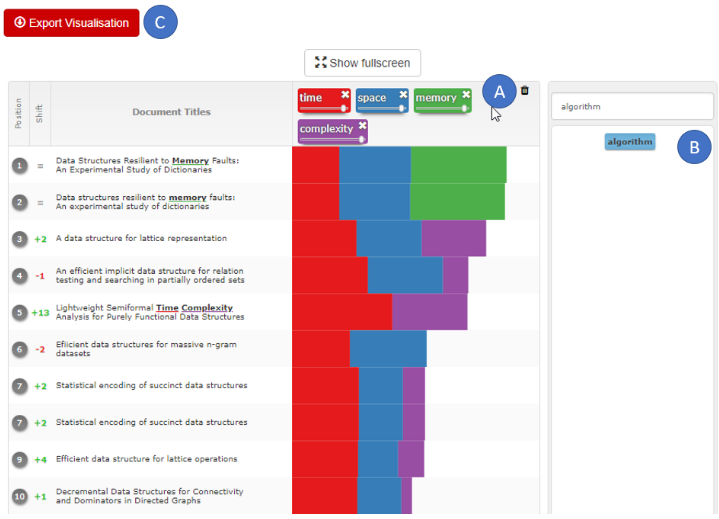
Note that *uRank* was first developed as part of the EEXCESS[56] project. The concept and the implementation were adapted for and integrated into the MOVING platform. Currently, uRank's keyword extraction performs Natural Language Processing (NLP) on the retrieved results in the web browser. This NLP processing includes the stemming and the removal of stop words for the English language. Optionally, *uRank* can also rely on server-side keyword extraction. uRank adaptations and improvements for MOVING include changes needed to accommodate the common data model provided by the MOVING's search engine (see Section 2), the initial handling of German language (mainly stop word elimination), adaptions of the look & feel of *uRank* to fit the platform, and numerous adjustments to the user interface to deal with the limited screen space and to incorporate the user feedback previously collected.

5.2.2 User Interface

The *uRank* user interface was further improved since the version in D3.2. (Vagliano et al., 2018). In addition to the improvements of the user interface, an export functionality was added, which captures the current search configuration and generates an image of the visualisation. The current version of *uRank* can be seen in Figure 27. Due to the limited screen space, the position and size of the document titles and other information shown in the list were made responsive to better utilise the available width. Additionally, the selected keywords in the tag cloud can now be removed all at once by clicking on the bin in the tag container (see Figure 27, A). The tag filter (see Figure 27, B) was reworked in order to allow users to easier search for available keywords by removing all those that do not match the entered text. Also, keyword selection via drag-and-drop was replaced by simple multiple selection in the tag cloud view.

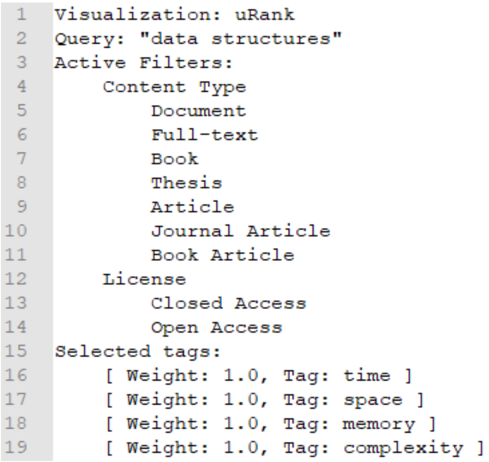
Export The state of the *uRank* visual interface with the currently set search configuration can be exported simply by clicking on the `Export Visualisation` button (Figure 27, C) on top of *uRank*. Clicking on the export button automatically initialises the download of a zip file. This zip file contains an image of the visualisation in its current state and a report file, which describes the settings that were applied to generate the visualisation. Figure 28 shows the content of a report file generated from the configuration shown in Figure 27. The name of the visualisation can be seen in the first line, followed by the search query in the second line. After that, all of the facets applied in the MOVING platform (`Active Filters`) are listed in a hierarchical manner. Those first three parts describe the current state of the page and are the same for the other visualisations that implement the export functionality. The last part (Figure 28, line 15) is visualisation specific and will contain additional information or actions performed inside the visualisation. In the case of *uRank*, a list of selected tags and applied weights is provided.

[56] `http://eexcess.eu/`



Position	Shift	Document Titles
1	=	Data Structures Resilient to Memory Faults: An Experimental Study of Dictionaries
2	=	Data structures resilient to memory faults: An experimental study of dictionaries
3	+2	A data structure for lattice representation
4	-1	An efficient implicit data structure for relation testing and searching in partially ordered sets
5	+13	Lightweight Semiformal Time Complexity Analysis for Purely Functional Data Structures
6	-2	Efiicient data structures for massive n-gram datasets
7	+2	Statistical encoding of succinct data structures
7	+2	Statistical encoding of succinct data structures
9	+4	Efficient data structure for lattice operations
10	+1	Decremental Data Structures for Connectivity and Dominators in Directed Graphs

Figure 27: Current version of *uRank* with highlighted improvements with respect previous one: A) the bin removes all the selected tags at once, B) the filter field reduces the number of displayed tags while entering the text, C) export the search configuration and the current state of the visualisation.



```
Visualization: uRank
Query: "data structures"
Active Filters:
    Content Type
        Document
        Full-text
        Book
        Thesis
        Article
        Journal Article
        Book Article
    License
        Closed Access
        Open Access
Selected tags:
    [ Weight: 1.0, Tag: time ]
    [ Weight: 1.0, Tag: space ]
    [ Weight: 1.0, Tag: memory ]
    [ Weight: 1.0, Tag: complexity ]
```

Figure 28: An example of a report created by exporting the *uRank* visualisation shown in Figure 27.

5.3 Content and statistics-based result filtering

In addition to the *Concept Graph* and *uRank*, the content and statistics-based visualisations were also improved. Moreover, the two separate bar chart visualisations, *Top Concepts* and *Top Sources*, presented in D3.2 (Vagliano et al., 2018) were bundled under a single tab called *Top Properties*, which now supports any number of different properties.

5.3.1 Top Properties

Figure 29 shows the current *Top Properties* bar chart visualisation. On top of the visualisation a drop-down menu (highlighted in green) can be used to select a property in the retrieved results set, for which the distribution of values should be displayed. It is currently possible to select between the following properties: Sources, Concepts, Keywords, Authors and Publication Year.

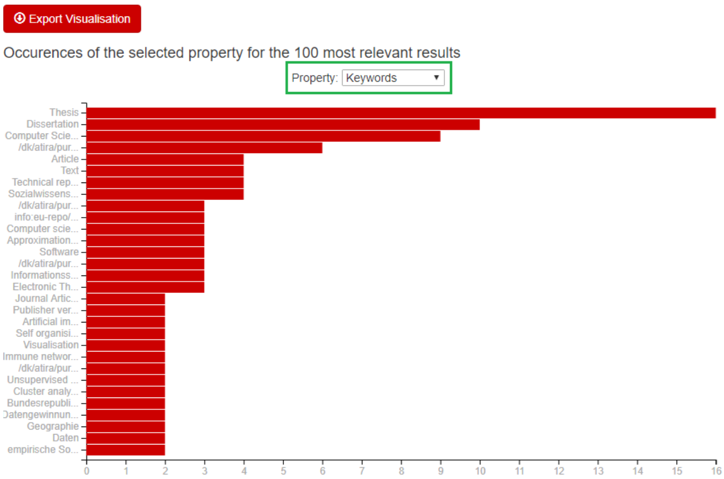
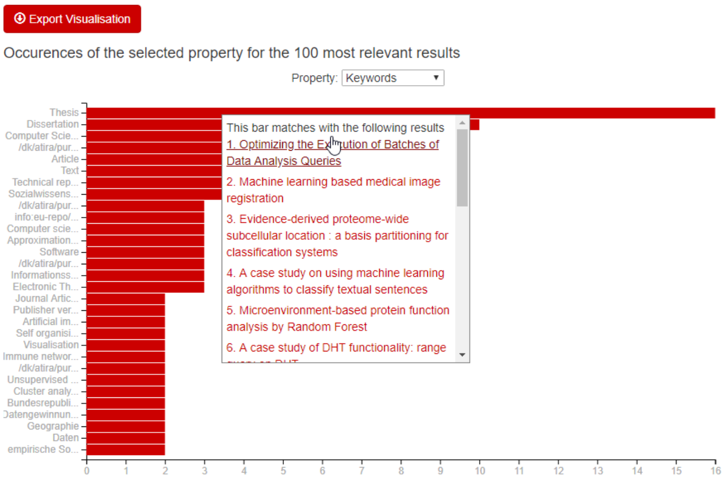


Figure 29: *Top Properties* visualisation - It is possible to select a property for which the bar chart should be created (highlighted in green).

Clicking on one of the bars reveals a small dialog (Figure 30) that lists all the documents from the retrieved result set with this property. The ranking of the documents in this document selection dialog is based on the initial ranking provided by the search engine. Clicking on one of the document names opens the associated URL in a new tab.



This bar matches with the following results

1. Optimizing the Execution of Batches of Data Analysis Queries

2. Machine learning based medical image registration

3. Evidence-derived proteome-wide subcellular location : a basis partitioning for classification systems

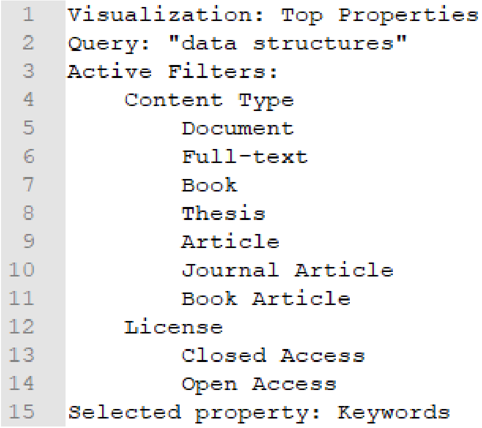
4. A case study on using machine learning algorithms to classify textual sentences

5. Microenvironment-based protein function analysis by Random Forest

6. A case study of DHT functionality: range

Figure 30: *Top Properties* visualisation - Clicking on a bar will reveal a document list associated with this property

Like *uRank*, the *Top Properties* visualisation also has an export functionality. Similarly to *uRank*, the bar chart visualisation export creates an image of the visualisation and generates a report with the name of the current visualisation, the search query and the applied facets. Additionally, the selected property, for which the bar chart was created, is also included in the export. This is depicted in Figure 31.



```
Visualization: Top Properties
Query: "data structures"
Active Filters:
    Content Type
        Document
        Full-text
        Book
        Thesis
        Article
        Journal Article
        Book Article
    License
        Closed Access
        Open Access
Selected property: Keywords
```

Figure 31: *Top Properties* - An example of a report created by exporting the visualisation illustrated in Figure 29.

5.3.2 Tag Cloud

Since the version described in deliverable D3.2 (Vagliano et al., 2018), the *Tag Cloud* visualisation received a visual overhaul, additional sorting and filtering mechanisms, inspection of documents related to a tag and, like *uRank* and the *Top Properties* visualisation, an export functionality. Figure 32 shows the current version of the *Tag Cloud*. Currently, the *Tag Cloud* can be sorted either alphabetically or according to the number of the keyword occurrences in the retrieved result. By default, the keywords are sorted from the most to the least frequently occurring (see Figure 32, A). Various filters can be applied to limit the number of the displayed keywords. It is possible to directly search for a certain keyword (Figure 32, B), display only keywords with occurrences in a certain range (Figure 32, C), or show keywords from documents that were published in a certain time range (Figure 32, D).

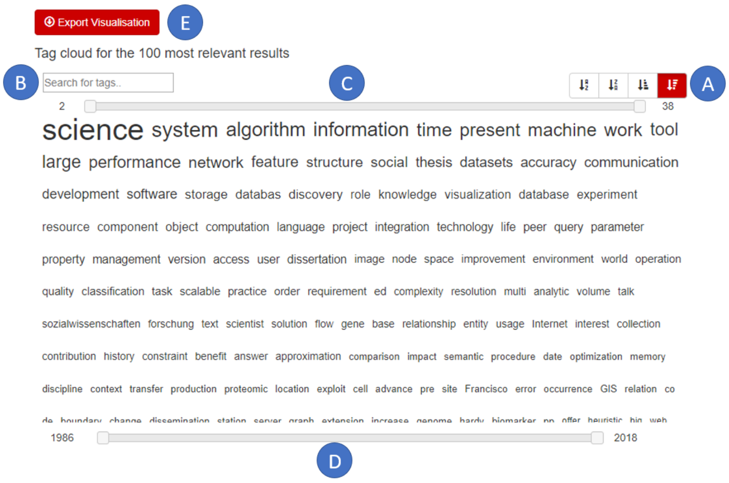
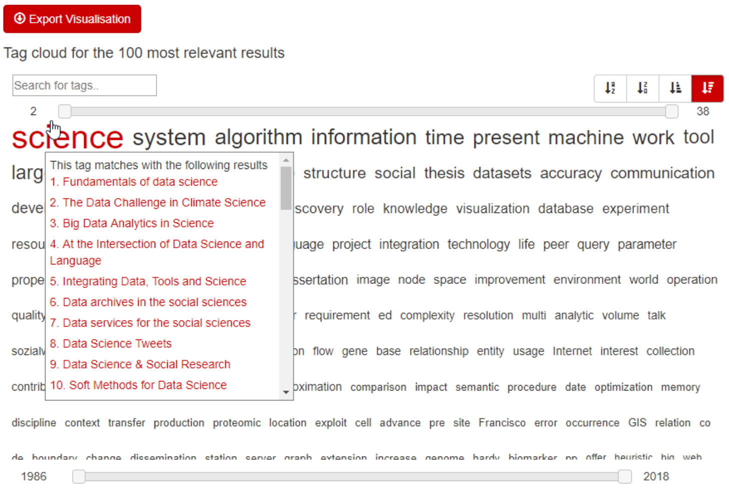

Export Visualisation
E
Tag cloud for the 100 most relevant results
B
Search for tags..
C
A
2
38
science system algorithm information time present machine work tool
large performance network feature structure social thesis datasets accuracy communication
development software storage databas discovery role knowledge visualization database experiment
resource component object computation language project integration technology life peer query parameter
property management version access user dissertation image node space improvement environment world operation
quality classification task scalable practice order requirement ed complexity resolution multi analytic volume talk
sozialwissenschaften forschung text scientist solution flow gene base relationship entity usage Internet interest collection
contribution history constraint benefit answer approximation comparison impact semantic procedure date optimization memory
discipline context transfer production proteomic location exploit cell advance pre site Francisco error occurrence GIS relation co
de boundary change dissemination station server graph extension increase genome hardy biomarker pp offer heuristic big web
1986
2018
D

Figure 32: *Tag Cloud* - A) sort buttons (default is by occurrence descending), B) keyword text filter, C) keyword occurrence range selector, D) document publishing year range selection, E) Tag Cloud export button.

Like the *Top Properties* visualisation, in the *Tag Cloud* it is also possible to inspect the documents associated with the keywords by clicking on a keyword, which will open a dialog listing the corresponding documents (Figure 33). The ranking of the documents in this document selection dialog, as in the *Top Properties*, is based on the initial ranking provided by the search engine. Furthermore, like *uRank* and the *Top Properties* visualisation, the *Tag Cloud* supports an export functionality. The export creates an image of the current *Tag Cloud* and a report containing the name of the visualisation, the current search configuration, the sort order and all the filters applied to the visualisation (Figure 34).


Export Visualisation
Search for tags..
2
38
science system algorithm information time present machine work tool
larg structure social thesis datasets accuracy communication
deve scovery role knowledge visualization database experiment
resou uage project integration technology life peer query parameter
prope ssertation image node space improvement environment world operation
quality r requirement ed complexity resolution multi analytic volume talk
sozialv on flow gene base relationship entity usage Internet interest collection
contrib oximation comparison impact semantic procedure date optimization memory
discipline context transfer production proteomic location exploit cell advance pre site Francisco error occurrence GIS relation co
de boundary change dissemination station server graph extension increase genome hardy biomarker pp offer heuristic big web
1986
2018

Tag cloud for the 100 most relevant results

This tag matches with the following results

1. Fundamentals of data science
2. The Data Challenge in Climate Science
3. Big Data Analytics in Science
4. At the Intersection of Data Science and Language
5. Integrating Data, Tools and Science
6. Data archives in the social sciences
7. Data services for the social sciences
8. Data Science Tweets
9. Data Science & Social Research
10. Soft Methods for Data Science

Figure 33: *Tag Cloud* - Clicking on a keyword opens the document selection dialog.

```
Visualization: Tag Cloud
Query: data science
Active Filters:
    Subject Area
        education
        business
        economics
        science
Minimum frequency: 2
Maximum frequency: 38
Minimum year: 1986
Maximum year: 2018
Sorted by: Frequency, Descending
```

Figure 34: *Tag Cloud*: an example of a report created by exporting the visualisation from Figure 32.

5.4 Experimental evaluation and comparison

Following the completion of the required features for the visualisations, a user study has been conducted. The user study was designed as a comparative study analysing the benefits and drawbacks of using the *Concept Graph* and *uRank* for information retrieval, compared to the classical approach of using the result list and the text-based faceted search interface. The objective of the user study was to measure the effectiveness and the usability of the visualisations, as well as to gather user feedback for further improving them.

5.4.1 Experimental setup

Two target groups were addressed in the study, which were determined from the two use cases specified in deliverable D1.1: *User requirements and specification of the use cases* (Bienia et al., 2017). Therefore, **researches** were the target group for the **first use case** and **auditors** the target group for the **second use case**.

In addition to the two target groups, the study aimed to compare the visual interfaces (graph visualisation and uRank) to the result list with the faceted search interface. Thus, the participants were divided into 4 groups. Two groups were created based on the different target groups, and then two sub-groups were generated depending on whether the corresponding participants used the visualisations or the text-based faceted interface. Figure 35 shows how the participants were divided into the different groups.

The participants in group *Use Case 1F* were assuming the role of researchers and used the faceted search interface, while the group *Use Case 1V* used the visual interfaces. Similarly, the group *Use Case 2F* was for participants in the role of auditors that used the faceted search interface, and group *Use Case 2V* for the ones using the visual interfaces.

Participants The participants consisted of Master students from the Graz University of Technology and of employees from Ernst & Young in Essen, Germany. All of the employees of Ernst & Young were assigned

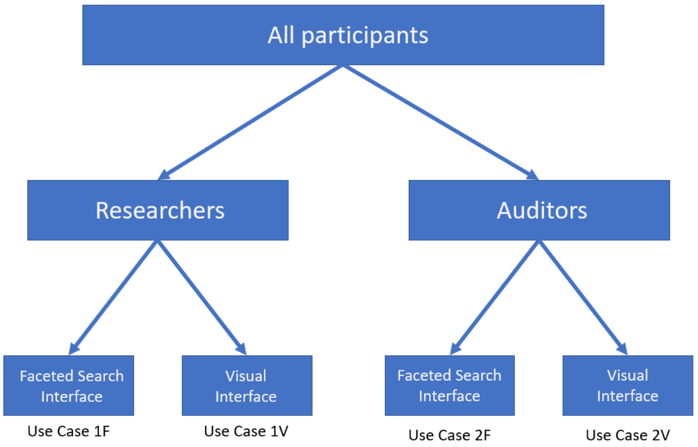


Figure 35: Division of participants in the user study into different groups.

to the Auditors use case, i.e. to groups *Use Case 2F* and *Use Case 2V*. The Master students of the Graz University of Technology consisted of students doing their master either in *Computer Science* or in *Software Engineering and Management*. Therefore, the *Software Engineering and Management* students, who have a strong background in economy, were also assigned to the Auditor group. *Computer Science* students, who have pronounced IT research interests, were assigned into the Researcher group. Due to the high number of participants ($n = 77$) and the short period in which the study should have been conducted, it was decided that the participants will take part in the study independently and unsupervised.

Procedure While the data (query and search results) used for the four groups differed from each other, the procedure of the user study and the executed user tasks were the same. Also, the data was selected in such a way that the execution of the tasks was of comparable complexity over the different groups. Since the study was conducted unsupervised, forms had to be created for each use case, that guided the participants step-by-step through the study. Google Forms[57] was used to create those forms, which contained all the necessary information to complete the study. To begin with the study, every participant had to open a link in a browser directing them to the form associated with their use case.

The study started by first giving the users a short overview of what the MOVING Platform is about and the procedure of the study (Appendix 8). After reading the participant information sheet according to the General Data Protection Regulation (Appendix 11), the participants had to confirm to the terms of the study before continuing.

Then, the participants were asked to answer some demographic questions, questions regarding their preferences, and questions regarding their skill set (Appendix 9). Thereafter, the participants were walked through the process of registration on the MOVING Platform (Appendix 10). After a successful registration and account activation, they were asked to read a document that explained all the relevant features that are used in the study (Appendix 12 and Appendix 13). Before starting the tasks, the participants were requested to familiarise with the MOVING Platform by logging in and exploring on their own for 5 minutes.

Task structure The complete user study contained in total three tasks. The first two tasks were designed to compare the visual interface *uRank* and the *Concept Graph* with the faceted search interface, while the third task asked the participants to interpret a *Learning-How-To-Search* widget. Since the focus here was on the visual interfaces, only the first and second task were described in this deliverable. The analysis and explanation of the third task will be covered separately.

Even though the use cases for the two target groups differ from each other, for purposes of comparison, the tasks in the user study were designed in such a way that a comparison between the different target groups was possible. For example, in one of the tasks that required using *uRank*, the participants from both target groups were asked to identify the most important keywords occurring in *uRank*'s tag cloud. This task could have been solved the same way independent of the researched topic. Furthermore, the queries and the retrieved data were picked in such a way that they had a comparable difficulty level. The same can be said for the tasks used to evaluate different user interfaces (visual vs. text-based) within the same use case. The tasks cover the same workflow and, although the data they operate on are not the same, the resulting level of difficulty

[57] https://www.google.com/forms/about/, last accessed: 07.01.2019

is very close or equal. In one task, performed with *uRank*, the participants were asked to identify the most important keywords in the *uRank*'s tag cloud, while in the matching task using the faceted search interface, the participants were asked to identify the most important subject areas. This allows us to directly compare visual and text-based interfaces.

Since the tasks have the same structure for both target groups, only the structure and the requirements of the tasks will be described. The exact task questions can be seen in Appendix 18, Appendix 17, Appendix 19 and Appendix 20.

Visualisations: Task 1 - In the first task, the participants were instructed to use *uRank* to find the most important topics of a result set by inspecting the most dominant keywords occurring in the *uRank*'s tag cloud. Additionally, after selecting some predefined keywords from the tag cloud, they were asked to interpret the changed result set and find certain results. To ensure that all participants used the exact same search configuration, a link was provided at the beginning of the task. Opening the link with a browser triggered a search query in the MOVING Platform and the results were displayed in *uRank*. Task 1 included three sub-tasks:

Task 1.1 After using the link, the participants were asked to investigate the *uRank*'s tag cloud and identify the most important keywords.

Task 1.2 Continuing from the previous sub-task, the participants had to first select a list of given keywords in the *uRank*'s tag cloud. They were asked then to find a result that addressed a specific topic from a combination of two given keywords.

Task 1.3 Further continuing from the previous sub-task, one additional predefined keyword had to be selected from the tag cloud. The participants were then asked to interpret the change in the ranking, by finding a result were the selected keyword was the most dominant.

Faceted search interface: Task 1 - The first task for the faceted search interface was structured the same way as the first task for the visualisations. The participants were instructed to only use the default result list and the faceted filters. They had to find out the most important subject areas for a result set, filter the results and find results matching certain criteria. As in Task 1 for the visualisations, the participants were also given a link that lead them to the page with the results for the query. The task had the following three sub-tasks:

Task 1.1 After using the link, the participants had to investigate the faceted filters and identify the most important subject areas for the result set.

Task 1.2 Continuing from the previous sub-task, a list of filters had to be applied by the participants in the panel with the faceted filters. After the result set had been updated, the participants had to find a result about a specific topic composed out of two given subject areas that were selected in the filters.

Task 1.3 Continuing from task 1.2, another filter had to be set. The participants had to find a result about the topic from the filter that they set.

Comparison of Task 1 with and without visualisations - By examining the sub-tasks of the first task, it is apparent that they are structured in the same way. In Task 1.1. the participants were required to find the most relevant topics from a result set. Using *uRank* this could have been done by inspecting the keywords in the tag cloud. The faceted search interface on the other hand provided a filter called "Subject Areas" in the faceted filters where the "Subject Areas" from the results were listed according to occurrence. In Task 1.2 the participants had to make adaptations to the initial query and evaluate the results based on those adaptations. While the participants using *uRank* had to select keywords from the tag cloud to alter the result list, the participants using the faceted search interface had to select specific filters, mainly subject areas, in the panel with the faceted filters. After the result list had been altered, in both sub-tasks, the participants had to find a result that covered a specific topic created by a combination of two keywords or subject areas. The participants using *uRank* could solve this task by inspecting the coloring of the stacked bar charts in the results set and comparing them with the colors of the selected keywords, whereas the participants using the classic result list had to read through the results to identify if the topic occurred in them. Task 1.3 expanded on Task 1.2, in which the participants had to select one additional predefined keyword or subject area. In this task, after adding the keyword or subject area, the participants had to find a result in the new result set that covered this topic the most. In *uRank* this could have been done by inspecting the shift column (how much the ranking of the document changed) or the stacked bar chart of the result (impact of this keyword on the rank). In the result list of the faceted search interface, the users had to read through all the new results.

Visualisations: Task 2 - In the second task, the participants were instructed to use the *Concept Graph* to find results by exploiting the connectedness of the documents over their properties.

This task was also started by giving the participants a link that applied the correct query and displayed the *Concept Graph*. After the participants used the link, they had to change the starting node type from *Document* to *Keyword* and turn on all labels for the nodes. This task was split into three sub-tasks, namely:

Task 1.1 After using the link and performing the configuration of the graph, the participants were asked to find a document related to a certain keyword.

Task 1.2 Continuing from the previous task, the participants were asked to find one author from a specific document from the previously expanded document nodes.

Task 1.3 Task 1.3 built further on Task 1.2, and asked the participants to find another publication from one of the authors from the previous task.

Faceted search interface: Task 2 - Task 2 for the faceted search interface is again very similar to Task 2 from the visualisations. The participants also started from a given link and by adding a filter to limit the result set down to a certain topic. Following that, in this task they also had to find connections between documents and their properties. This task was split in the following three sub-tasks:

Task 1.1 After opening the link, the participants were asked to set a given filter and find a result that was about the same topic as the selected filter.

Task 1.2 Continuing from the previous task, the participants had to find out the authors of a specific document.

Task 1.3 Task 1.3 further built on the previous task. The participants were asked to find another publication by the same author they found in the previous task.

Comparison of Task 2 with and without visualisations - Task 2 was structured in the same way for the participants using the *Concept Graph* and for the participants using the faceted search interface. While the participants using the *Concept Graph* used keywords as starting points, the participants using the faceted search interface used subject areas. Task 2 demonstrated an exploratory approach to information retrieval. The participants started out with a predefined query and based on a sub-topic they began their search. In Task 2.1 the participants using the *Concept Graph* started by selecting a specific keyword as the starting point and the participants using the faceted search interface used the "Subject Area" filter to limit the result set. Subsequently, the participants had to name a document related to the selected sub-topic. Task 2.2 continued from Task 2.1 and asked the participants to find out who the author of a specific document from this result subset was. The last task continued with the exploration and the participants had to find out another publication by one of the authors of the previous document.

Metrics The following metrics have been measured during the study:

Success Rate The success rate, or completion rate, is a metric that shows the percentage of participants that successfully completed a task. Even though some of the sub-tasks could have been partially solved, for better statistical analysis it was decided to treat solved tasks either as a complete success (1) or failure (0).

Subjective Workload After every sub-task had been completed, the participants were asked to fill out a questionnaire that inquired about the perceived difficulty of the task and how the participants thought they performed. The questionnaire was based on the NASA Task Load Index (NASA-TLX) (Hart & Stavenland, 1988) and asked the participants for their perceived "Mental Demand", "Physical Demand", "Temporal Demand", "Performance" and "Frustration". The participants could rate each of the asked questions on a Likert scale from 1 - 7. In addition to the NASA-TLX questions, a comment field was added, where the participants could give feedback for this particular sub-task. The used questionnaire can be seen in Appendix 14.

System Usability Scale (SUS) When the participants completed all the sub-tasks with a particular tool, they were additionally asked to fill out a questionnaire that would measure the tools usability. The questionnaire contained ten questions that asked the participant to rate a statement regarding the usability of the tool on a Likert scale from 1-7. The asked questions were based on the standardised System Usability Scale, and can be seen in Appendix 15.

Subjective Feedback Beside the System Usability Scale questions, after finishing all the sub-tasks for a particular tool, the participants could moreover give their subjective opinion of the tool by answering four questions. The participants where asked what they liked/disliked about the tool, for what purpose would they use it and what other tools would they have preferred for this task. The questions can be seen in Appendix 16.

5.4.2 Results

Participant data From the 77 participants, 71.43% (55) were either fluent in English or native speakers, 27,27% (21) stated that their English proficiency was "Okay" and only 1,3% (1) of the users had basic English skills.

93.51 % (72) of the participants never used the MOVING Platform before the study, while 6,49% (5) were already familiar with it. All of the participants familiar with the platform were part of Use Case 2 (Auditors).

When asked about their usage of search engines, 88,31% (68) said they were using them on a daily basis, 6,49% (5) several times a week, 2,6% (2) several times a month and 2,6% (2) stated "Never".

Regarding the question about the usage of visualisation tools, 58,44% (45) of the participants stated, that they never used them. 15,58% (12) used them once a month or less, 11,69% several times a week, 7,79% (6) every day and 6,49% (5) used them several times a month.

62.34% (48) of the participants never used any data analysis tools before. 14.29% (11) use them once a month or less, 10.39% (8) every day, 9.09% (7) several times a week and 3.9% (3) several times a month.

Looking at the education of the participants, 63.64% (49) had a bachelor's degree, 19.48% (15) a secondary education or high school, 15.58% (12) a master's degree and 1.30% (1) a doctorate or higher.

Additionally, 80,52% or (62) of the participants were in the age range of 18-27, 18,18% (14) were between 28-37 and 1,3% (1) were aged between 48-57. 80.52% (62) of the participants were male, while 19.48% (15) were female.

Success rates of tasks The success rates of the performed tasks for each individual use case can be inspected in Table 15. The success rate of each task was calculated by averaging over the success rates of each sub-task performed for this task (Table 16). Task 1 compared the faceted search interface to *uRank*, while Task 2 compared it to the *Concept Graph*. In use case 1 (Researchers), Task 1, *uRank* had a higher success rate then the faceted search interface. In Task 2 for the same use case, the participants using the faceted search interface solved the tasks more successfully then the participants using the *Concept Graph*. For use case 2 (Auditors) in Task 1, the participants using the faceted search interface outperformed the participants using *uRank*, while in Task 2 the participants using the *Concept Graph* had a higher success rate then the ones using the faceted search interface. Looking at the success rates for the sub-tasks 2.1 and 2.1 Extra in Use case 2F in Table 16, it can be observed that they are noticeably lower then the other success rates. This indicates that there might have been a problem with how the task was stated.

Table 15: Success rates in percent for the tasks calculated by averaging over the sub-tasks.

Use case	Task 1	Task 2
Use Case 1F	82.54	85.71
Use Case 1V	90.48	77.38
Use Case 2F	87.50	56.25
Use Case 2V	77.19	82.89

Table 16: Success rates in percent for the sub-tasks performed during the user study.

Use case	Task 1.1	Task 1.2	Task 1.3	Task 2.1	Task 2.1 Extra	Task 2.2	Task 2.3
Use Case 1F	71.43	90.48	85.71	95.24	76.19	95.24	76.19
Use Case 1V	100.00	90.48	80.95	80.95	52.38	95.24	80.95
Use Case 2F	81.25	81.25	100.00	18.75	12.50	100.00	93.75
Use Case 2V	100.00	63.16	68.42	94.74	68.42	94.74	73.68

Subjective Workload The participants in the study used either the visual or the faceted search interface, thus we had two groups to compare. Additionally, the number of participants for each use case differed from each other and the gathered data from the subjective workload questionnaire was not normally distributed. Therefore, the Kruskal-Wallis test (Kruskal & Wallis, 1952) was used to compare these groups. The reported H is a test statistic that is used to detect if the null hypothesis should be rejected and is needed to calculate p. With p we can determine if there was a significant ($p < 0.05$) or not significant ($p > 0.05$) difference between the two groups.

Use Case 1 *uRank vs Faceted Search Interface* - The workload for *uRank* did not significantly differ from the workload of the faceted search interface over all tasks ($H(5) = 5.5$, $p = 0.36$). A focused comparison of the mean ranks between the groups showed that the workloads were not significantly different when using *uRank* or the faceted search interface for Task 1, Task 2 or Task 3.

Use Case 1 *Concept Graph vs Faceted Search Interface* - The workload for the *Concept Graph* did not significantly differ from the workload of the faceted search interface over all tasks ($H(5) = 10.85$, $p = 0.054$). A focused comparison of the mean ranks between the groups showed that the workloads were not significantly different when using the *Concept Graph* or the faceted search interface for Task 1, Task 2 or Task 3.

Use Case 2 *uRank vs Faceted Search Interface* The workload for *uRank* did not significantly differ from the workload of the faceted search interface over all tasks ($H(5) = 4.66$, $p = 0.45$). A focused comparison of the mean ranks between the groups showed that the workloads were not significantly different when using *uRank* or the faceted search interface for Task 1, Task 2 or Task 3.

Use Case 2 *Concept vs Faceted Search Interface* The workload was significantly affected depending what tool was used ($H(5) = 24.2$, $p < 0.001$). A focused comparison of the mean ranks between the groups showed that when the *Concept Graph* was used for Task 2.1, the workload was significantly higher compared to the workload of the same task when the faceted search interface was used ($difference = 35$). However, for Task 2.2 and Task 2.3, there were no significant differences in the overall workload for the *Concept Graph* and the faceted search interface.

Usability Analysis Report

uRank

Use Case 1 - After averaging over all questions and participants the obtained mean raw score amounted to 75.4 (standard deviation, σ = 15.8). *uRank* fell in the 70-79 percentile range in the curved grading scale interpretation of SUS scores, thus obtained a B+ score (Sauro & Lewis, 2012). These scores are subdivided into Usable (questions 1,2,3,5,6,7,8,9) and Learnable (questions 4 and 10) sub-scales following the approach presented in (Lewis & Sauro, 2009). After the multipliers were adjusted on the 7-point Likert scale (2.08 and 8.33), *uRank* scored an A+ for Learnable (mean, μ = 88.06, σ = 19.10) and B for Usable (μ = 76.06, σ = 16.2).

Use Case 2 - After averaging over all questions and participants the obtained mean raw score amounted to 73.5 (σ = 11.85). *uRank* fell in the 65-69 percentile range in the curved grading scale interpretation of SUS scores, thus obtained a B- score. Moreover, *uRank* scored an A- for Learnable (μ = 80.67, σ = 16) and B for Usable (μ = 75.42, σ = 12.42).

Concept Graph

Use Case 1 - After averaging over all questions and participants the obtained mean raw score amounted to 51.6 (σ = 28,7). The *Concept Graph* fell in the 0-14 percentile range in the curved grading scale interpretation of SUS scores, thus obtained a F score. Moreover, the *Concept Graph* scored a C for Learnable (μ = 66.64, σ = 31.5) and F for Usable (μ = 50.41, σ = 30.3).

Use Case 2 - After averaging over all questions and participants and obtained the mean raw score amounted to 40.6 ($\sigma = 22.02$). The *Concept Graph* fell in the 0-14 percentile range in the curved grading scale interpretation of SUS scores, thus obtained a F score. Moreover, the *Concept Graph* scored a D for Learnable ($\mu = 53.5$, $\sigma = 27.67$) and F for Usable ($\mu = 40$, $\sigma = 22.66$).

Faceted Search Interface

Use Case 1 - After averaging over all questions and participants the obtained mean raw score amounted to 73.6 ($\sigma = 22.9$). The faceted search interface fell in the 65-69 percentile range in the curved grading scale interpretation of SUS scores, thus obtained a B- score. Moreover, the faceted search interface scored a A+ for Learnable (μ =90, $\sigma = 22.8$) and B- for Usable ($\mu = 73.30$, $\sigma = 25.15$).

Use Case 2 - After averaging over all questions and participants the obtained mean raw score amounted to 79.9 ($\sigma = 18.2$). The faceted search interface fell in the 85-89 percentile range in the curved grading scale interpretation of SUS scores, thus obtained an A- score. Moreover, the faceted search interface scored an A+ for Learnable (μ =90.06, $\sigma = 20.45$) and A for Usable ($\mu = 81.38$, $\sigma = 20.15$).

Subjective Feedback After all of the tasks have been completed using a particular tool, the participants had to describe what they liked/disliked about it, for what purpose would they use it and what other tool would they prefer instead. Additionally, the participants had the option to give subjective feedback after each performed sub-task. The feedback given after every sub-task has been added to the 4 feedback categories at the end of the tested tool. Since the feedback was in free text form it had to be manually analyzed and categorised. Only statements which were reoccurring are listed in the tables for each tool.

uRank

Table 17: Aggregated subjective feedback to the question: *What did you like about uRank?*

Statement	# Participants
Matching colors of keywords and bars in the results	8
Keyword coloring	7
Easy to use	7
Visual representation	6
Easy to read	3
Selection of keywords	3
List of extracted keywords	3
Simplicity	3
Re-ranking	2
Weighting of keywords	2
Fast	2
Seeing the impact of the keywords on the rank	2
Novel and interesting	2

Table 18: Aggregated subjective feedback to the question: *What did you dislike about uRank?*

Statement	# Participants
Loading time	16
Too many irrelevant keywords	4
Font-size too small	3
Slow animations	2
User interface could be improved (not modern enough)	2

Table 19: Aggregated subjective feedback to the question: *For which tasks would you personally use uRank?*

Statement	# Participants
Scientific & general research	20
Search for topics by inspecting the keywords	2
Search for articles containing specific keywords	2
For finding keywords	2

Table 20: Aggregated subjective feedback to the question: *If you could have solved any of the subtasks with other tools of your choice, which ones would you have used?*

Statement	# Participants
Google	14
Google Scholar	5

Concept Graph

Table 21: Aggregated subjective feedback to the question: *What did you like about the Concept Graph?*

Statement	# Participants
Design of visual interface	10
Being able to find documents with same properties	5
Intuitive	5
Getting overview of data	4
Being able to see connections to other documents	4
Find other publications of authors	3
Coloring of nodes and icons	3
Easy to use	2

Table 22: Aggregated subjective feedback to the question: *What did you dislike about the Concept Graph?*

Statement	# Participants
Interactions not clear (moving, closing, removing nodes)	19
Hard to use with too many nodes	16
No copy function for titles	10
Overlapping text with nodes	8
Node relevance not clear	7
Labels should be on by default	7
Not intuitive	4
Duplicate documents with different properties	4
Labels hiding nodes	4
Missing reset function	3
User Experience	2

Table 23: Aggregated subjective feedback to the question: *For which tasks would you personally use the Concept Graph?*

Statement	# Participants
Scientific & general research (specifically related work)	12
Searching for publications of author	10
Searching for similar articles	5
Seeing relations between publications	3
Exploring connections between keywords and documents	2
Searching for connections between two topics	2

Table 24: Aggregated subjective feedback to the question: *If you could have solved any of the subtasks with other tools of your choice, which ones would you have used?*

Statement	# Participants
Google	11
Google Scholar	5
uRank	2

Faceted search interface

Table 25: Aggregated subjective feedback to the question: *What did you like about the "Filter by" sidebar?*

Statement	# Participants
Simple	7
Intuitive	6
Easy to use	6
Well organised	4
High number of various filters	3

Table 26: Aggregated subjective feedback to the question: *What did you dislike about the "Filter by" sidebar?*

Statement	# Participants
Reload of page after every click on filter	17
Platform was slow	13
Not all authors listed in the facets	6
Not being able to copy text from title	5
Appearance	3
Underscore instead of whitespace	2
Document count disappeared in filter after selection	2
Unclear naming of filters (Dataset Collection, License)	2
Duplicate results	2

Table 27: Aggregated subjective feedback to the question: *For which tasks would you personally use the "Filter by" sidebar?*

Statement	# Participants
Find documents in specific area	8
Filtering	6
Scientific & general research	6
Author selection	5
Language selection	4
Publishing year selection	4

Table 28: Aggregated subjective feedback to the question: *If you could have solved any of the subtasks with other tools of your choice, which ones would you have used?*

Statement	# Participants
Google	10
Google Scholar	4
Advanced Search	4
IEEE	2

5.4.3 Discussion

Success rates and subjective workload Looking at the success rates in Table 15, the tools fared rather equally. In use case 1 (researchers), the participants using *uRank* where more successful than the participants using the faceted search interface, while in use case 2 (auditors) it was the opposite. Something similar can be observed in Task 2 for both use cases. In use case 1, the success rate of the participants using the faceted search interface was higher than of the ones using the *Concept Graph*. In Task 2 however, the participants using the *Concept Graph* fared noticeably better. Looking at the individual sub-tasks that made up this task (Table 16), it can be observed that the participants had great troubles solving Task 2.1 and Task 2.1 Extra in Use Case 2F. The success rates for those two task were particularly low, which strongly influenced the success rate for the whole task. This might indicate that the participants did not understand the task properly.

Furthermore, looking at the subjective workload of the participants for each individual sub-task, there were no significant differences between the usage of the visual interfaces and the usage of the faceted search interface, except for Task 2.1. The workload in this particular sub-task was significantly lower for the faceted search interface compared to the *Concept Graph*, even though the participants failed this task much more often. Again, this might indicate that the participants misunderstood the task and therefore, by thinking it was easily solvable, picked the wrong answer.

System usability scale and subjective feedback

uRank *uRank* received a B+ score on the system usability scale for use case 1 (researchers) and a B- in use case 2 (auditors), meaning that it has a good usability. When looking at the Learnable and Usable aspects of the system usability scale, one can see that *uRanks*'s strengths lie especially in its learnability, meaning that the participants were confident that they could easily learn to use *uRank*.

Looking at the subjective feedback that the participants gave after completing all the sub-tasks of task 1, it can be seen that they liked many aspects of the interface. The coloring of the keywords (7), the matching of those colors to the colors of the bars (8), as well as the overall visual representation (6) were mentioned often. Moreover, the participants liked the ease of use (7) and the simplicity (3) of the tool. All of those statements reinforce the good rating the tool received on the system usability scale. The key features of the tool (re-ranking, weighting of keywords, keyword extraction) were also positively mentioned.

When asked what they did not like about the tool, many participants stated that it took too long to load (16). This issue is more of an issue of the search engine than of *uRank*, since a majority of *uRank*'s loading time is spent by waiting on the results of the query. The participants also stated, that by default too many irrelevant keywords are displayed in the *uRank*'s tag cloud (4) and that the font-size is too small on high resolution displays (3). Both of those issues are easily solvable.

Furthermore, the participants mentioned that they would use *uRank* primarily for scientific and general research (20), which suits both use cases very well.

Concept Graph The *Concept Graph* received on the system usability scale in both use cases an F score. The Learnable aspect of the system usability scale was rated by the participants with a C and a D, while the Usable aspect was rated in both use case with a F.

Looking at the subjective feedback of the participants, it could be noted that many did not understand how the interactions were supposed to work in the graph (19) and therefore, it not being intuitive (4). Especially, problems with closing or removing the nodes were mentioned. Both of those features were already integrated in the graph, yet the participants could not find out how to use them, even though they could access the instructions inside the graph at any given time. A more suitable mapping of actions to functionalities in the *Concept Graph* will have to be found, that does not require any kind of help.

Furthermore, the participants mentioned the complexity of the graph when too many nodes were opened (16), specifically because of the limited area that is available for the visualisation and due to the labels overlapping nodes (8) and nodes overlapping other nodes (4). The relevance of the nodes was also something that was not easily discernible for the participants (7). Duplicate nodes with different properties (4) also cause confusion. Other easily resolvable issues in the *Concept Graph* included a missing functionality to copy text from the labels (10), the labels not being on by default (7) and a missing reset functionality (3) that would revert the graph to the starting configuration.

Nevertheless, there were also aspects of the *Concept Graph* that the participants liked. Most of them liked the design of the visual interface (10) and the coloring of the nodes and icons (4). The ability to find documents with the same properties (5) and connections to other documents via those properties (5) thus getting an overview over the data (4) was also highlighted. The participants emphasised that finding publications of authors (3) was especially useful. Moreover, there were participants which stated that the graph was intuitive (5) and easy to use (2).

Regarding the application of the *Concept Graph*, most of the participants would use it for scientific & general research (12), searching for publications of specific authors (10). Searching for similar articles (5) and inspecting relations between publications (3) or topics (2) was also stated.

In conclusion, the participants liked the idea and possible applications of the graph, while the offered interactions and functionalities will have to be further improved.

Faceted search interface The faceted search interface received a B- and A- score on the system usability scale in the measured for the two use cases. In the Usable aspect of the system usability scale it received a B- and A score. It also received twice an A+ score for Learnable, which was expected, due to the fact that most participant were already familiar with such an interface.

What the participants liked most about the faceted search interface was that it was simple (7) intuitive (6) and easy to use (6). The organisation of the filters was also well received (4), as well as the high number of filters (3).

Many participants complained that the filters were immediately applied after each click (17) and that the platform was slow (13). This could be solved by adding an apply filter button at the top or bottom of the panel with the faceted filters, which can be clicked after all the appropriate filters have been set. Other complaints included missing authors in the faceted filters (6), the document count disappearing after a filter has been applied (2), the white spaces being replaced with underscores (2) and that they did not understand some filters (2), specifically the *Dataset Collection* and *License* filter. In regards to the result list, the participant were missing the option to copy the title of a result (5) and that some results would be listed twice (2).

5.5 Future work

Since the revised requirements have been addressed and all the planned visualisations are already included in the MOVING platform, the next steps will be to address the pressing issues which came up from the user study. Particularly the mapping of user actions to the functionalities of the *Concept Graph* will have to be simplified in such a way, that the users can interact with the graph as much as possible without having to rely on any instructions. Furthermore, the labeling of the nodes will have to be adjusted in such a way, that other nodes do not get hidden behind them. The visualisation of entities in the *Concept Graph*, which are extracted from the documents in the result set, will be added as well. Additionally, all of the minor complaints about *uRank* will be addressed.

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 better support the new data processing components.

Regarding data acquisition, we improved the crawlers. The Social Stream Manager is now capable to crawl funding sources, notably H2020 funding topics. The MOVING crawlers now implement a basic mechanism to avoid to introduce duplicates in the index. This is a preliminary step to reduce the workload of the more advance duplicate detection and removal module. The crawlers pipeline has been extended to add temporal fragmentation metadata generated by VIA to the videos crawled from the Web. FDC has been equipped with a new adaptive crawling mechanism. The FLuID model has been extended to support incremental schema-level indexes. Unlike existing schema-level indexes, our incremental schema-level index has an efficient update mechanism to avoid costly re-computations of the entire index. This enables the administrators of the MOVING platform to monitor changes to data instances at a schema-level, trace this changes, and constantly keep an up-to-date schema level index for Linked Data.

Regarding data processing, we improved some techniques and we investigated new ones. We extended our technique for duplicate detection with a method for identifying duplicates based on the similarity of their metadata entries. The methods has a low computational complexity and has been successfully evaluated against a gold standard of annotated duplicates. Additionally, we introduced two different techniques for entity extraction and linking, which target the different needs of auditors and researchers both in terms of entities and documents. We also addressed the problem of detecting and forecasting trending topics. Specifically, we developed a trend detection and forecasting method, which is based on attention neural networks and can learn the trending patterns of time series as well as forecast the upcoming values. Our preliminary evaluation is promising, although further testing is ongoing and the integration of this methods in the platform will depend on the final results. We compared various machine learning techniques on multi-label classification tasks using only the documents' titles. We considered both the EconBiz dataset (which is integrated in the platform) and PubMed. Our results indicate that title-based methods can match or even outperform the full-text performance when enough training data is available.

About video processing, we extended the transcript-based lecture video fragmentation method. We integrated a keyword extraction procedure and introduced a new large-scale dataset of artificially-generated lectures. We also compare various configurations results of the developed method and show which was the best in this dataset. This method is applied to the VideoLectures.NET videos collected in the platform. We also developed a new machine learning technique for dimensionality reduction and concept annotation with complex concept labels for large-scale datasets.

WevQuery, the user logging solution, now tracks additional and more complex events to better support new services, such as the recommender system, as well as new functionalities such as the search history of a user. We evaluated the pattern mining features of WevQuery through a user study where participants performed typical analysis tasks with WevQuery. We found that workflows assisted by WevQuery facilitate higher order knowledge discoveries. The pattern mining extension has also been used to evaluate the Adaptive Training Support in the MOVING platform. These results will be presented in the upcoming deliverable D1.4.

Finally, we have introduced the new visualisations' features. Most notably, the Concept Graph's user interface has been improved and now enables users to explore the graph starting from different types of nodes, activate or deactivate explanatory labels and help text, and provides filters and standard graph properties, such as density and average degree. The uRank's user interface is now responsive in order to more efficiently use the limited screen space and a new export functionality has been added to save the current visualisation and its status. The two separate bar chart visualisations, *Top Concepts* and *Top Sources* are merged in a single tab which support different properties. Their UI has also been improved and they offer the export similarly to uRank. The tag cloud's UI is also better and provides the export as well. The Concept Graph and uRank has been evaluated through a user study. The results showed that uRank has a good usability and it easy to use, and participants liked the idea and application of the Concept Graph, although its interactions and functionalities can be further improved.

Integrating all the heterogeneous components and technologies for data analysis, visualisation, search, etc., described above, the MOVING platform offers such a combination of functionalities that cannot be found in other platforms.

7 Appendix - Common data model

Listing 4: Common data model v1.1 to integrate full-texts, metadata, HTML content and video data.

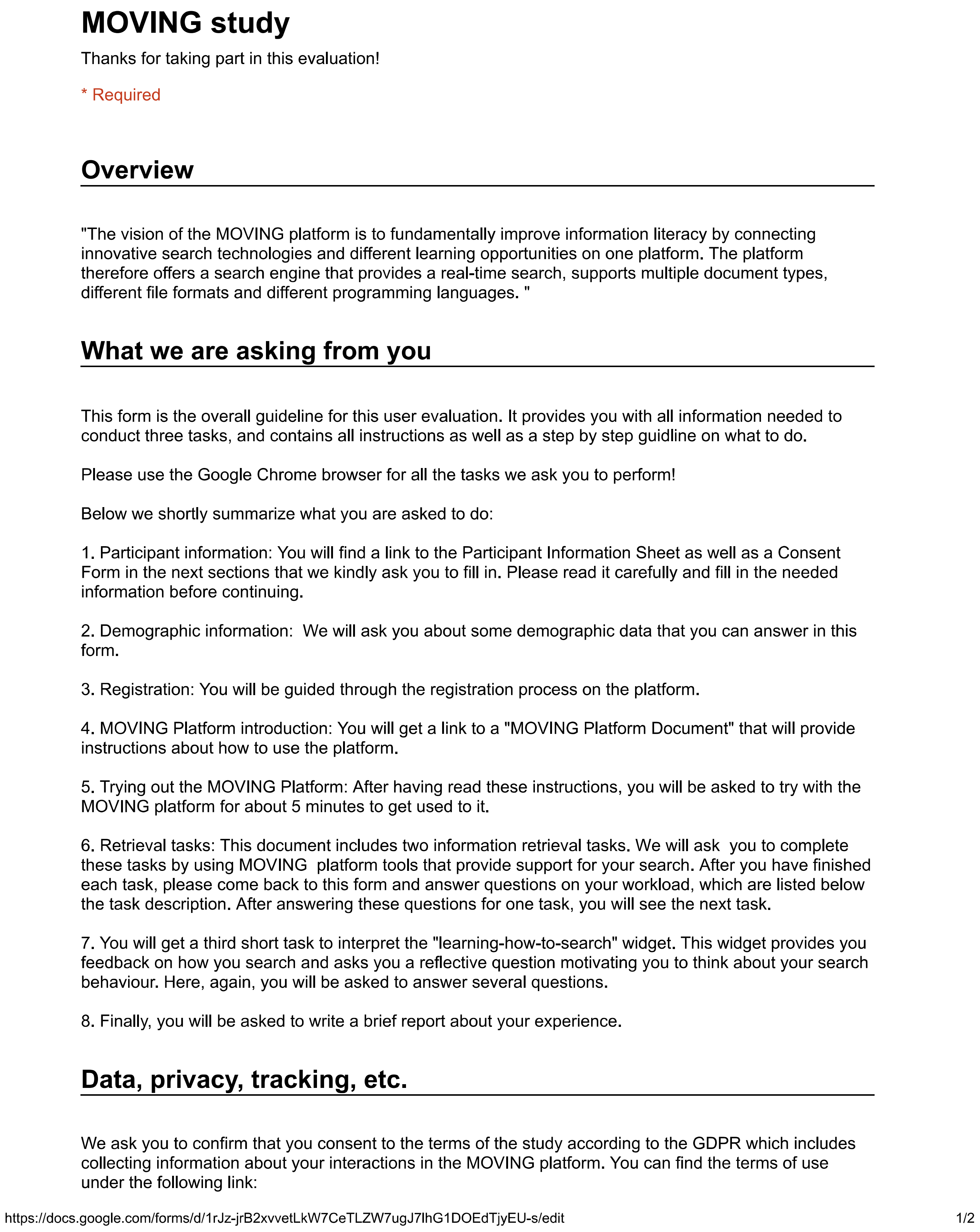
```
{
"$schema": "http://json-schema.org/draft-04/schema#",

"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", ... "Journal of Materials and
      Engineering Structures","UUM Repository"]

  },
  "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",
   "SAGE-METHOD", "SAGE-RESEARCHFIELD", "SAGE-THEORY", "SAGE-DATATYPE", "SAGE-MEASURE",
   "SAGE-TOOL"]
},
"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": { "$ref": "#/definitions/location" }
  },
  "required": ["name"],
  "additionalProperties": false
},
"video_fragment": {
  "type": "object",
  "properties": {
    "URL": {"type": "string", "format": "uri" },
    "thumbnailURL": {"type": "string", "format": "uri" },
    "start": {"type": "integer","minimum": 0},
    "end": {"type": "integer","minimum": 0},
    "video_concepts": {
      "type": "array",
      "items": { "$ref": "#/definitions/concept" }
    }
  },
  "required" : ["URL", "start", "end"],
  "additionalProperties": false
},
"concept": {
  "type": "object",
  "properties": {
    "label": { "type": "string" },
    "URL": { "type": "string", "format": "uri" },
    "relevanceScore": { "type": "number" }
    },
    "required": ["label"],
    "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": { "$ref": "#/definitions/documentType" },
  "language": {
    "type" : "string",
    "pattern" : "[a-z]{2}"
  },
  "concepts": {
    "type": "array",
    "items": { "$ref": "#/definitions/concept" }
  },
  "sectors": {
    "type": "array",
    "items": { "$ref": "#/definitions/concept" }
  },
  "subjects": {
    "type": "array",
    "items": { "$ref": "#/definitions/concept" }
  },
  "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" },
        "mention": { "type": "string" },
        "URIs": {
          "type": "array",
          "items": { "type": "string", "format": "uri" }
        },
        "label": { "type": "string" },
        "confidence": {
          "type": "number",
          "minimum": 0,
          "maximum": 1
        },
        "relevance": {
          "type": "number",
          "minimum": 0,
          "maximum": 1
        },
        "mentions": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "startChar": { "type": "integer", "minimum": 0 },
              "endChar": { "type": "integer", "minimum": 0 }
            }
          }
        },
        "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": { "$ref": "#/definitions/searchDomain" }
  },
  "lawSpecific_metadata" : {},
  "duplicateSpecific_metadata" : {},
  "temporal_metadata": {},
  "videoSpecific_metadata" : {
    "type" : "object",
    "properties": {
      "video_concepts": {
        "type": "array",
        "items": { "$ref": "#/definitions/concept" }
      },
      "video_fragments": {
        "type": "array",
        "items": {"$ref": "#/definitions/video_fragment" }
      }
    }
  },
  "fundingSpecific_metadata": {
    "type" : "object",
    "properties": {
      "external_identifier": {"type": "string"},
      "EU_pillar": {"type": "string"},
      "category": {"$ref": "#/definitions/concept" },
      "callDeadline": {
        "anyOf": [
          { "$ref": "#/definitions/date" },
          { "type": "string", "format": "date-time" }
        ]
      }
    }
  }
},
"required": ["title", "source", "docType"],
"additionalProperties": false
}
```

8 Appendix - User study: introduction to the study



MOVING study

Thanks for taking part in this evaluation!

* Required

Overview

"The vision of the MOVING platform is to fundamentally improve information literacy by connecting innovative search technologies and different learning opportunities on one platform. The platform therefore offers a search engine that provides a real-time search, supports multiple document types, different file formats and different programming languages. "

What we are asking from you

This form is the overall guideline for this user evaluation. It provides you with all information needed to conduct three tasks, and contains all instructions as well as a step by step guidline on what to do.

Please use the Google Chrome browser for all the tasks we ask you to perform!

Below we shortly summarize what you are asked to do:

1. Participant information: You will find a link to the Participant Information Sheet as well as a Consent Form in the next sections that we kindly ask you to fill in. Please read it carefully and fill in the needed information before continuing.

2. Demographic information: We will ask you about some demographic data that you can answer in this form.

3. Registration: You will be guided through the registration process on the platform.

4. MOVING Platform introduction: You will get a link to a "MOVING Platform Document" that will provide instructions about how to use the platform.

5. Trying out the MOVING Platform: After having read these instructions, you will be asked to try with the MOVING platform for about 5 minutes to get used to it.

6. Retrieval tasks: This document includes two information retrieval tasks. We will ask you to complete these tasks by using MOVING platform tools that provide support for your search. After you have finished each task, please come back to this form and answer questions on your workload, which are listed below the task description. After answering these questions for one task, you will see the next task.

7. You will get a third short task to interpret the "learning-how-to-search" widget. This widget provides you feedback on how you search and asks you a reflective question motivating you to think about your search behaviour. Here, again, you will be asked to answer several questions.

8. Finally, you will be asked to write a brief report about your experience.

Data, privacy, tracking, etc.

We ask you to confirm that you consent to the terms of the study according to the GDPR which includes collecting information about your interactions in the MOVING platform. You can find the terms of use under the following link:

https://docs.google.com/forms/d/1rJz-jrB2xvvetLkW7CeTLZW7ugJ7lhG1DOEdTjyEU-s/edit 1/2 https://drive.google.com/open?id=1qy3-SuEju03tCkTmadMI4qefbrq3Ut3b

Please read this Participant Information Sheet carefully!
You will of course be free to revoke this consent at anytime and stop participating.

Please confirm....

I agree that the information provided above can be used by the MOVING team for the purpose of the MOVING study and for aggregated, anonymous reporting. I understand that I can withdraw my consent to participate at any time by informing my contact person within the MOVING team. The data collected will not be willingly shared by the MOVING team with any party other than the ones necessary to handle their collection and processing, and will be destroyed after the study is completed.

I have read and understood the information about the MOVING study.

I agree to participate in the study and my participation is voluntary. I confirm that I am at least 18 years old.

I know that I have the possibility to withdraw from this study at any point in time and without providing an reasons and that I will not suffer any negative consequences.

Furthermore I was informed I can only ask for the deletion of my data (or part of it) by using my email address which I have used to register to the MOVING platform and which I provided in this questionnaire.

I am aware, that this data will be anonymously stored for at least five years.

Privacy Clause:

I consent to agree that the data collected in the context of this study will be used in a pseudo- and anonyimized way for scientific purposes including the publication in scientific journals.

1. **I consent to the above described intended use of my data.** *

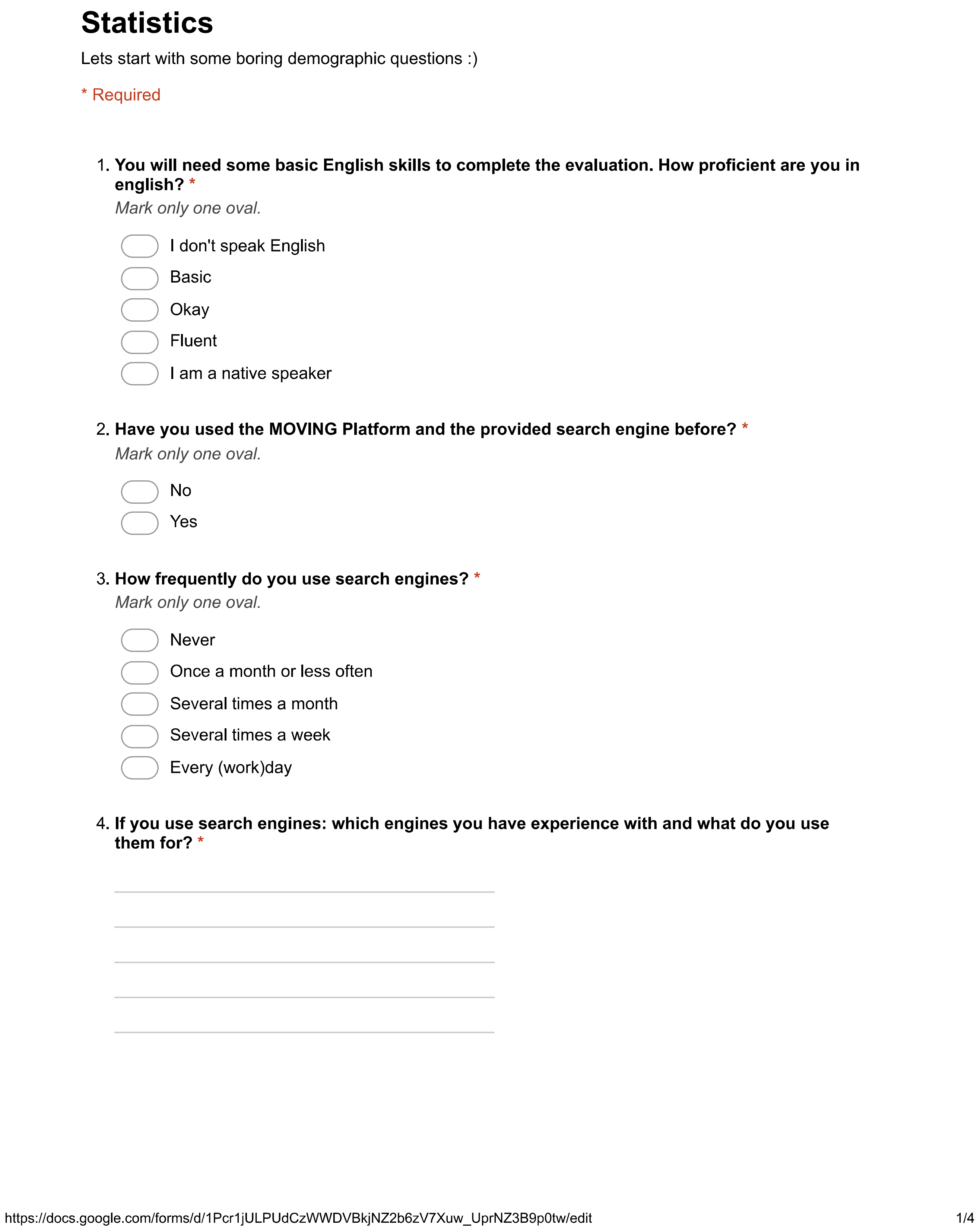
   Mark only one oval.

   ○ Yes

2. **The e-mail adress I will use to register to the MOVING platform** *

   ____

9 Appendix - User study: statistics



Statistics

Lets start with some boring demographic questions :)

* Required

1. **You will need some basic English skills to complete the evaluation. How proficient are you in english? ***

   Mark only one oval.

   ○ I don't speak English
   ○ Basic
   ○ Okay
   ○ Fluent
   ○ I am a native speaker

2. **Have you used the MOVING Platform and the provided search engine before? ***

   Mark only one oval.

   ○ No
   ○ Yes

3. **How frequently do you use search engines? ***

   Mark only one oval.

   ○ Never
   ○ Once a month or less often
   ○ Several times a month
   ○ Several times a week
   ○ Every (work)day

4. **If you use search engines: which engines you have experience with and what do you use them for? ***

https://docs.google.com/forms/d/1Pcr1jULPUdCzWWDVBkjNZ2b6zV7Xuw_UprNZ3B9p0tw/edit 1/4

5. **How frequently do you use visualization tools?** *
Mark only one oval.

○ Never
○ Once a month or less often
○ Several times a month
○ Several times a week
○ Every (work)day

6. **If you use visualization tools: which tools you have experience with and what do you use them for?**

____
____
____
____
____

7. **How frequently do you use data analysis tools?** *
Mark only one oval.

○ Never
○ Once a month or less often
○ Several times a month
○ Several times a week
○ Every (work)day

8. **If you use data analysis tools: which tools do you have experience with and what do you use them for?** *

____
____
____
____
____

9. **What's your age?** *
Mark only one oval.

○ 18-27
○ 28-37
○ 38-47
○ 48-57
○ 58-67
○ 68-77

10. **Gender** *

Mark only one oval.

○ F
○ M

11. **Country** *

12. **What's your Primary Job Function** *

Mark only one oval.

○ Student
○ Researcher
○ Educator
○ Manager
○ Healthcare
○ Engineer
○ Finance
○ Other

13. **Education** *

Mark only one oval.

○ No formal education
○ Primary education
○ Secondary education or high school
○ Work-related qualifications
○ Bachelor's degree
○ Master's degre
○ Doctorate or higher

14. **Primary Interest** *

Check all that apply.

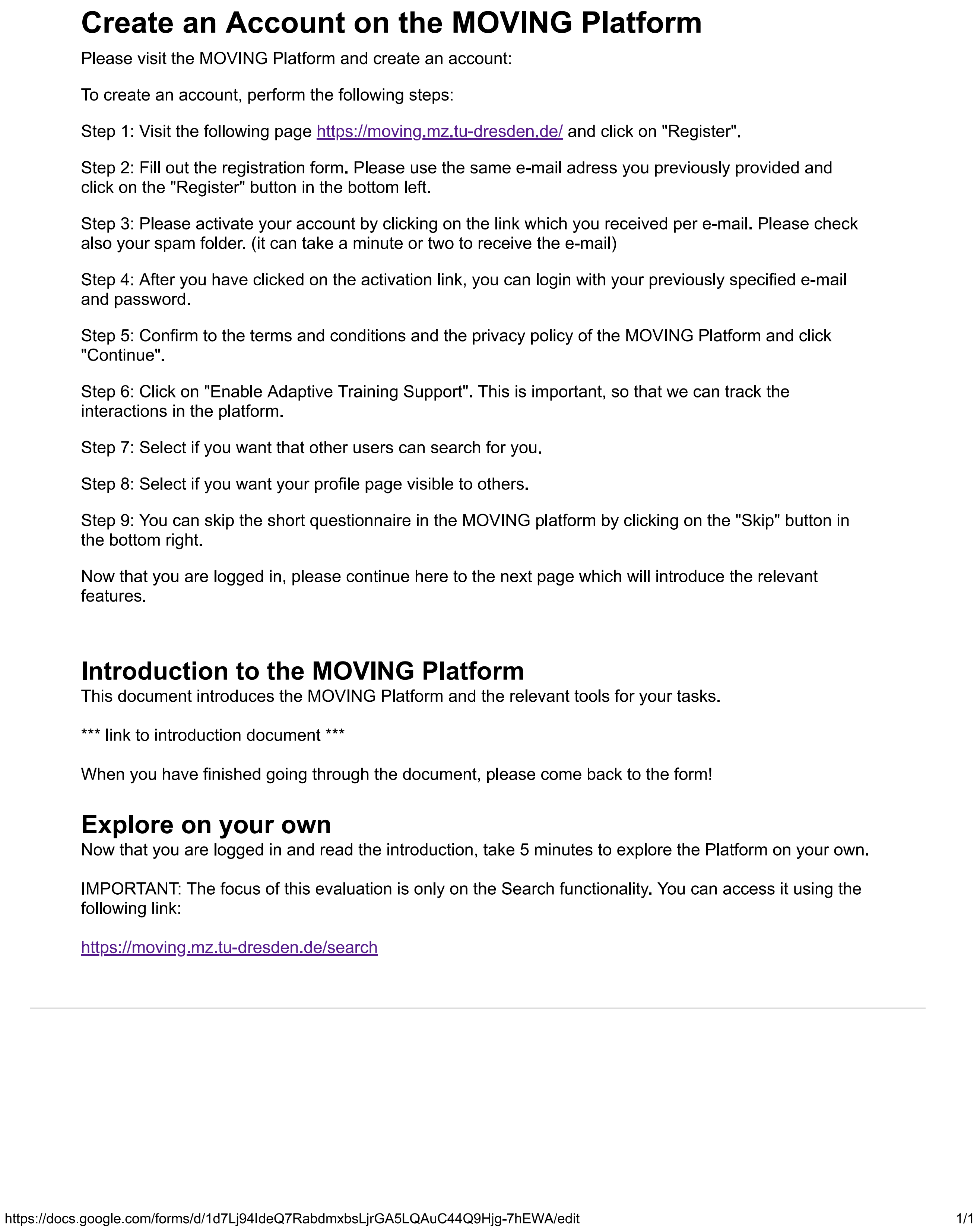
□ Computer Applications
□ Computer Systems Organization
□ Computing Methologies
□ Hardware
□ Software
□ Information systems
□ Mathematics of Computing
□ Theory of Computation
□ Data
□ Biological informatics
□ Human medicine
□ Psychology
□ Business
□ Economics
□ Social science
□ Stocks
□ Trading
□ Other: ____________________

15. **Please provide your level of expertise in each of the following keywords: range 1 (Basic knowledge) – 5 (Expert)** *

Mark only one oval per row.

	1	2	3	4	5
Programming/Development Support	○	○	○	○	○
User Experience Design	○	○	○	○	○
Information Retrieval	○	○	○	○	○
Social Media	○	○	○	○	○
Virtual/ Augmented Reality	○	○	○	○	○
Medical and health support	○	○	○	○	○
Visualization	○	○	○	○	○
World wide web and Hypermedia	○	○	○	○	○
Education/Learning	○	○	○	○	○
Information Seeking/Search	○	○	○	○	○
SWOT Analysis	○	○	○	○	○
PEST Analysis	○	○	○	○	○

10 Appendix - User study: account creation and introduction to the platform



Create an Account on the MOVING Platform

Please visit the MOVING Platform and create an account:

To create an account, perform the following steps:

Step 1: Visit the following page https://moving.mz.tu-dresden.de/ and click on "Register".

Step 2: Fill out the registration form. Please use the same e-mail adress you previously provided and click on the "Register" button in the bottom left.

Step 3: Please activate your account by clicking on the link which you received per e-mail. Please check also your spam folder. (it can take a minute or two to receive the e-mail)

Step 4: After you have clicked on the activation link, you can login with your previously specified e-mail and password.

Step 5: Confirm to the terms and conditions and the privacy policy of the MOVING Platform and click "Continue".

Step 6: Click on "Enable Adaptive Training Support". This is important, so that we can track the interactions in the platform.

Step 7: Select if you want that other users can search for you.

Step 8: Select if you want your profile page visible to others.

Step 9: You can skip the short questionnaire in the MOVING platform by clicking on the "Skip" button in the bottom right.

Now that you are logged in, please continue here to the next page which will introduce the relevant features.

Introduction to the MOVING Platform

This document introduces the MOVING Platform and the relevant tools for your tasks.

*** link to introduction document ***

When you have finished going through the document, please come back to the form!

Explore on your own

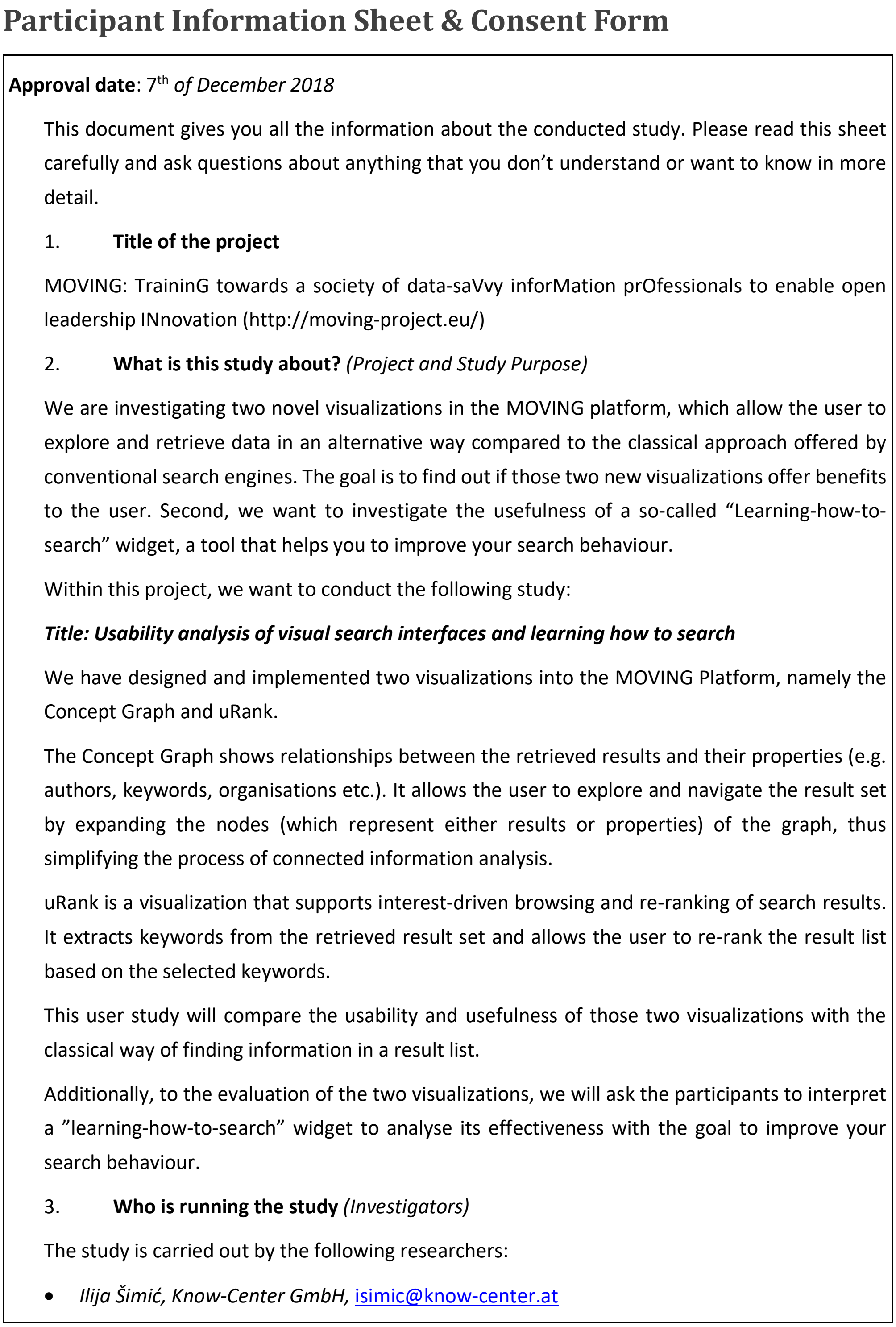
Now that you are logged in and read the introduction, take 5 minutes to explore the Platform on your own.

IMPORTANT: The focus of this evaluation is only on the Search functionality. You can access it using the following link:

https://moving.mz.tu-dresden.de/search

https://docs.google.com/forms/d/1d7Lj94IdeQ7RabdmxbsLjrGA5LQAuC44Q9Hjg-7hEWA/edit 1/1

11 Appendix - User study: participant information sheet



Participant Information Sheet & Consent Form

Approval date: 7th *of December 2018*

This document gives you all the information about the conducted study. Please read this sheet carefully and ask questions about anything that you don't understand or want to know in more detail.

1. **Title of the project**

MOVING: TraininG towards a society of data-saVvy inforMation prOfessionals to enable open leadership INnovation (http://moving-project.eu/)

2. **What is this study about?** *(Project and Study Purpose)*

We are investigating two novel visualizations in the MOVING platform, which allow the user to explore and retrieve data in an alternative way compared to the classical approach offered by conventional search engines. The goal is to find out if those two new visualizations offer benefits to the user. Second, we want to investigate the usefulness of a so-called "Learning-how-to-search" widget, a tool that helps you to improve your search behaviour.

Within this project, we want to conduct the following study:

Title: Usability analysis of visual search interfaces and learning how to search

We have designed and implemented two visualizations into the MOVING Platform, namely the Concept Graph and uRank.

The Concept Graph shows relationships between the retrieved results and their properties (e.g. authors, keywords, organisations etc.). It allows the user to explore and navigate the result set by expanding the nodes (which represent either results or properties) of the graph, thus simplifying the process of connected information analysis.

uRank is a visualization that supports interest-driven browsing and re-ranking of search results. It extracts keywords from the retrieved result set and allows the user to re-rank the result list based on the selected keywords.

This user study will compare the usability and usefulness of those two visualizations with the classical way of finding information in a result list.

Additionally, to the evaluation of the two visualizations, we will ask the participants to interpret a "learning-how-to-search" widget to analyse its effectiveness with the goal to improve your search behaviour.

3. **Who is running the study** *(Investigators)*

The study is carried out by the following researchers:

- *Ilija Šimić, Know-Center GmbH,* isimic@know-center.at

- Vedran Sabol*, Know-Center GmbH,* vsabol@know-center.at
- *Angela Fessl, Know-Center GmbH, afessl@know-center.at*

4. **Why have I been invited to participate in the study?** *(Eligibility)*

You have been invited to take part in this research study because you are either a student of the University of Technology Graz registered for the Web Technologies course (706.704), an employee of the Know-Center GmbH, or an employee of Ernst & Young GmbH.

5. **What would I be asked to do if I take part?** *(Overall Description of Participation)*

As participant of this study, you will be asked to

- perform some predefined search tasks on the MOVING platform using specific features
- fill in questionnaires after performing each task

There are no right or wrong answers, every answer given is important for our research.

6. **What is the duration of the research?** *(Length of Participation)*

Altogether, the duration of the evaluation will last from the 6rd of December to the 14th of December 2018 and will last for about 1 hour in total.

7. **What are the risks associated to the study?** *(Risks of Participation)*

Aside from providing your time, we do not expect any risks/costs associated with taking part in this study.

8. **What are the benefits associated to the study?** *(Benefits of Participation)*

We cannot guarantee or promise that employees will receive any direct benefits from participating in the study. Students will be able to learn how such a user study can be prepared and performed. Your input will be used as for investigating if the developed visualizations fulfil their purpose.

9. **Is there any compensation/payment/incentives?** *(Compensation/Payment/Incentives)*

No, you will not receive any compensation/payment/incentives.

10. **What happens if I do not want to take part or change my mind?** *(Volunteer Statement)*

It is up to you to decide whether or not to take part. If you decide to participate in the study, you may withdraw from it at any time without giving a reason and with detriment to yourself. However, students will miss the opportunity for gathering hands on experience on performing user evaluations of Web-based applications.

11. **Will the collected information about me be kept confidential?** *(Confidentiality Statement)*

By providing your consent, you agree that we are collecting personal information about you for the purposes of this research study. This information will only be used for the purposes outlined in this Participant Information Sheet. Your information will be stored securely and your identity/information will be kept strictly **confidential**. Study findings may be published, but all data for analysis will be **anonymised**. In reporting on the research findings, we will not reveal the names of any participants. Although every effort will be made to protect your identity, there is a risk that your participation (but no individual data) might be identifiable in publications due to the nature of the study and/or the results.

12. **What if something goes wrong?** *(Formal complaint about the conduct)*

If you want to make a formal complaint about the conduct of the study, please contact:

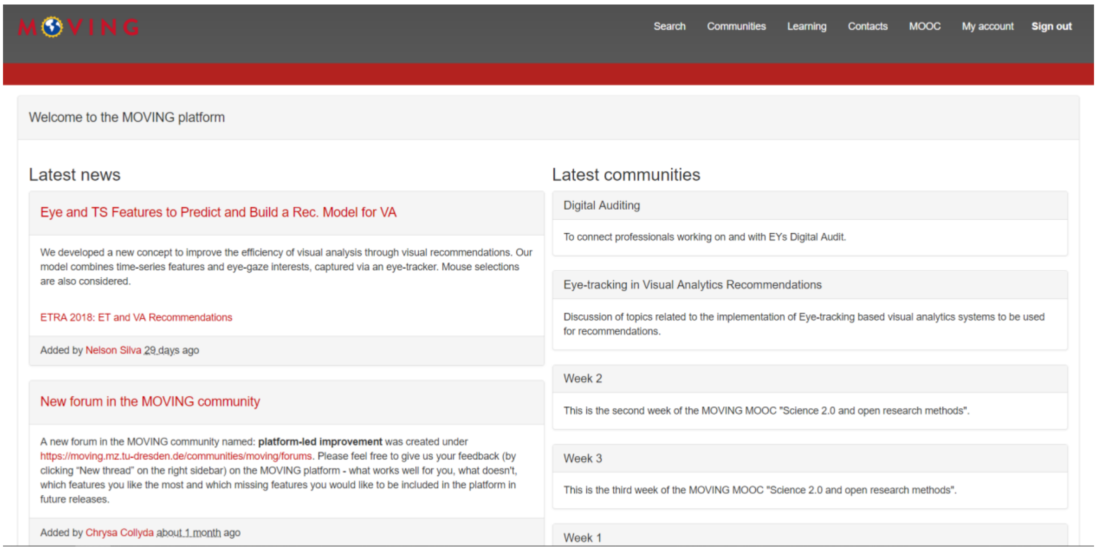
- *Ilija Šimić, Know-Center GmbH, isimic@know-center.at*
- *Vedran Sabol, Know-Center GmbH, vsabol@know-center.at*
- *Angela Fessl, Know-Center GmbH, afessl@know-center.at*

Thank you for reading this information sheet and for considering taking part in this research.

12 Appendix - User study: introduction to the MOVING platform (visualisations)

Introduction to the MOVING Platform

After you have logged in into the MOVING Platform, you see the page as shown in Figure 1.



Welcome to the MOVING platform

Latest news

Eye and TS Features to Predict and Build a Rec. Model for VA

We developed a new concept to improve the efficiency of visual analysis through visual recommendations. Our model combines time-series features and eye-gaze interests, captured via an eye-tracker. Mouse selections are also considered.

ETRA 2018: ET and VA Recommendations

Added by Nelson Silva 29 days ago

New forum in the MOVING community

A new forum in the MOVING community named: **platform-led improvement** was created under https://moving.mz.tu-dresden.de/communities/moving/forums. Please feel free to give us your feedback (by clicking "New thread" on the right sidebar) on the MOVING platform - what works well for you, what doesn't, which features you like the most and which missing features you would like to be included in the platform in future releases.

Added by Chrysa Collyda about 1 month ago

Latest communities

Digital Auditing

To connect professionals working on and with EYs Digital Audit.

Eye-tracking in Visual Analytics Recommendations

Discussion of topics related to the implementation of Eye-tracking based visual analytics systems to be used for recommendations.

Week 2

This is the second week of the MOVING MOOC "Science 2.0 and open research methods".

Week 3

This is the third week of the MOVING MOOC "Science 2.0 and open research methods".

Week 1

Figure 1: Starting page of the MOVING Platform

The focus of this evaluation will lie in the search functionality of the platform and the exploration of the result set; hence we will not explain all the features of the platform, but only the ones which are relevant for you to perform your evaluation tasks.

Clicking on the "Search" button in the navigation bar, at the top of the page, will bring you to the simple search page (Figure 2). This is the main entry point for performing searches in the MOVING Platform.

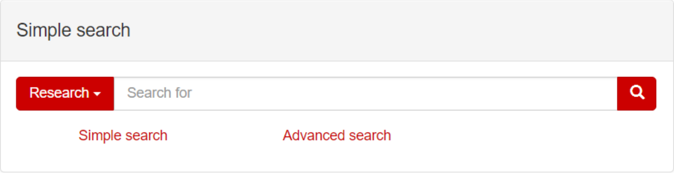
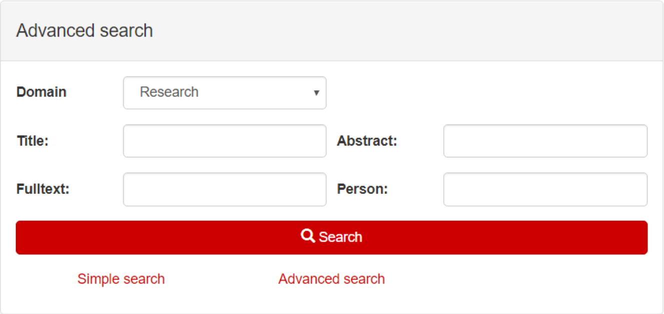


Figure 2: Simple search

The Advanced search (Figure 3) offers an even more fine-grained way to search in the platform. You can search there explicitly for an author, and select if you want to search the titles, the abstracts, or the bodies of the documents.



Advanced search

Domain: Research

Title:

Abstract:

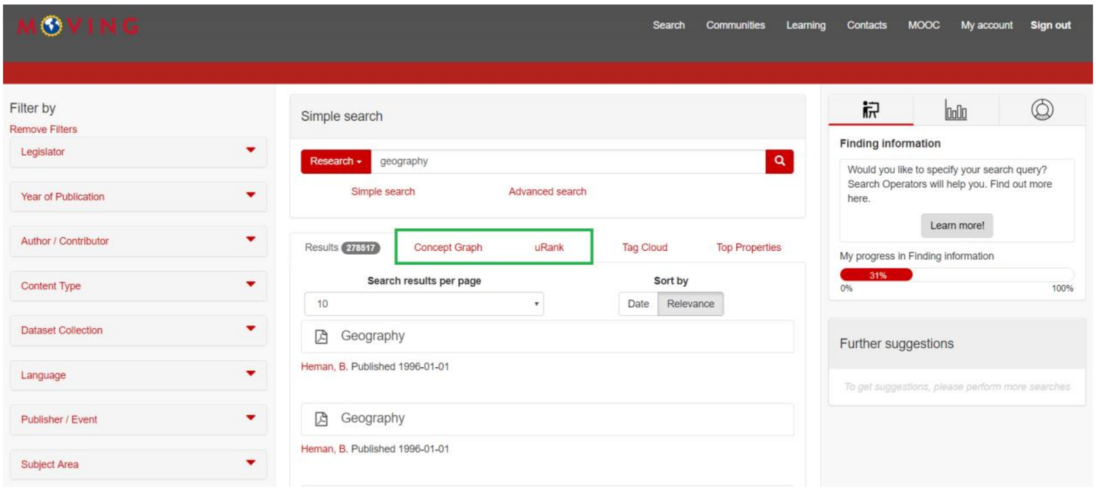
Fulltext:

Person:

Figure 3: Advanced search

After you have performed a search, you will be presented with the results page (Figure 4). The results page offers a wide variety of options to further explore the retrieved documents. On one hand, there is a classic results list, and on the other, there are various visualizations which allow you to visually explore and analyze the result set.

In your tasks, you will be asked to use the "Concept Graph" and "uRank" visualizations to analyze the result set and retrieve some specific information. They can be accessed by clicking on their respective tabs (highlighted in green).



Search: geography

Search results per page: 10

Sort by: Relevance

Geography

Heman, B. Published 1996-01-01

Geography

Heman, B. Published 1996-01-01

Finding information

Would you like to specify your search query? Search Operators will help you. Find out more here.

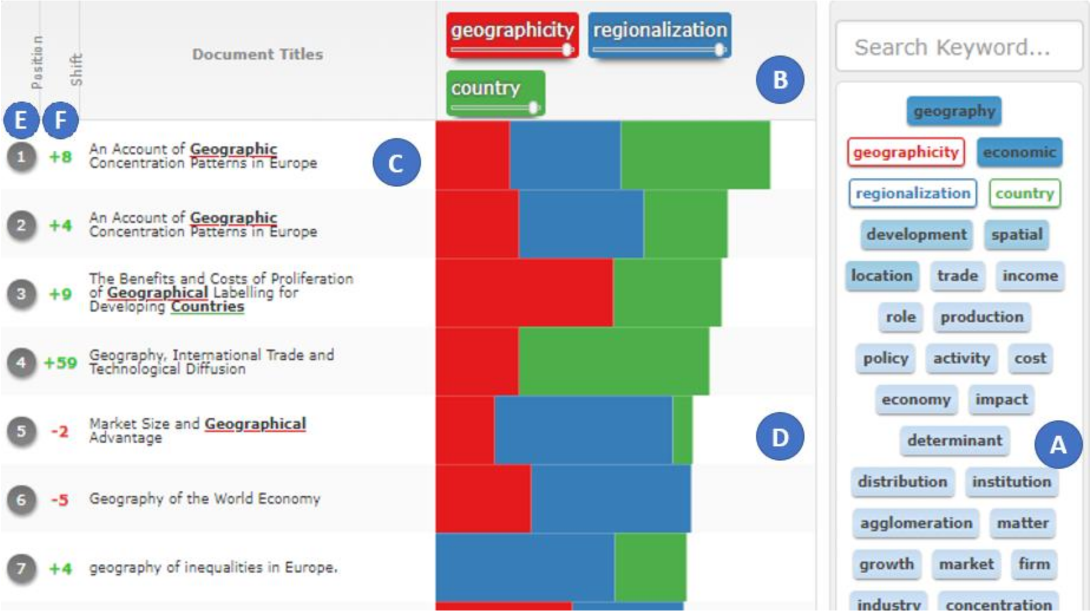
My progress in Finding information: 31% (0% – 100%)

To get suggestions, please perform more searches

Figure 4: Results page – The Concept Graph and uRank can be accessed by clicking on their respective tabs (highlighted in green)

uRank

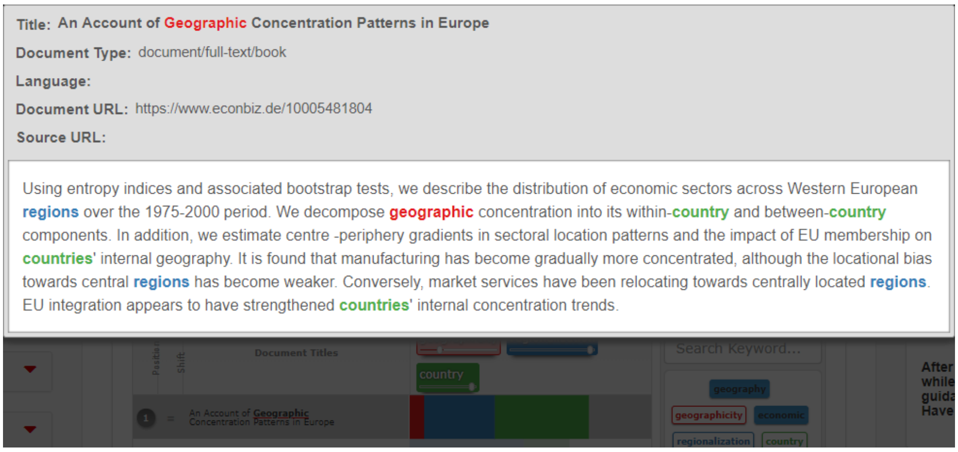
uRank (Figure 5) is a novel visualization, which supports interest-driven browsing and re-ranking of search results. uRank extracts keywords from the retrieved documents and displays them in a tag cloud (Figure 5A). Selecting tags from this tag-cloud adds them to the selected tags area (Figure 5B) and re-ranks the result list (Figure 5C). The more often a selected keyword occurs in a document from the result list, the higher it will move in the ranking. The "position" (Figure 5E) column shows the current position of the document, while the "Shift" (Figure 5F) column shows, how much the ranking of the result has changed since the last tag was added or removed. For example, the document at position 4 "Geography, International Trade and Technological Diffusion" moved up 59 positions, since the last keyword was selected. The stacked bar charts (Figure 5D) show how much each of the selected tags influence the ranking of this particular document in the displayed results list. Additionally, it is possible to fine-tune the importance of each of the selected keywords, by moving the slider associated with the keyword.



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 5: uRank – A) uRanks tag cloud, B) selected tags, C) result list, D) stacked bar chart visualization showing how much a keyword influences the ranking of each document, E) position of the document in the result list, F) The number of places the document went up or down after adding or removing the last keyword

Clicking on one of the items in the result list opens a preview of the document (Figure 6). This preview shows additional information about the document, like the URL for accessing the document and the abstract. The selected keywords are additionally highlighted in the abstract and title of the document.



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 6: uRank– Document preview

Concept Graph

In the Concept Graph, the documents and their attached properties are represented as nodes of a graph. The Concept Graph allows you to explore the relations of the retrieved documents and their properties by expanding the nodes of the graph. There are 6 types of nodes in the Concept Graph: Document, Author, Affiliation (Organization), Publication Year, Concept and Keyword. Each one of the different node types is represented with a different color (Figure 7).

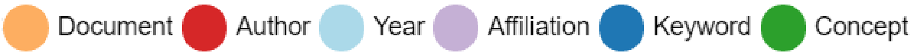
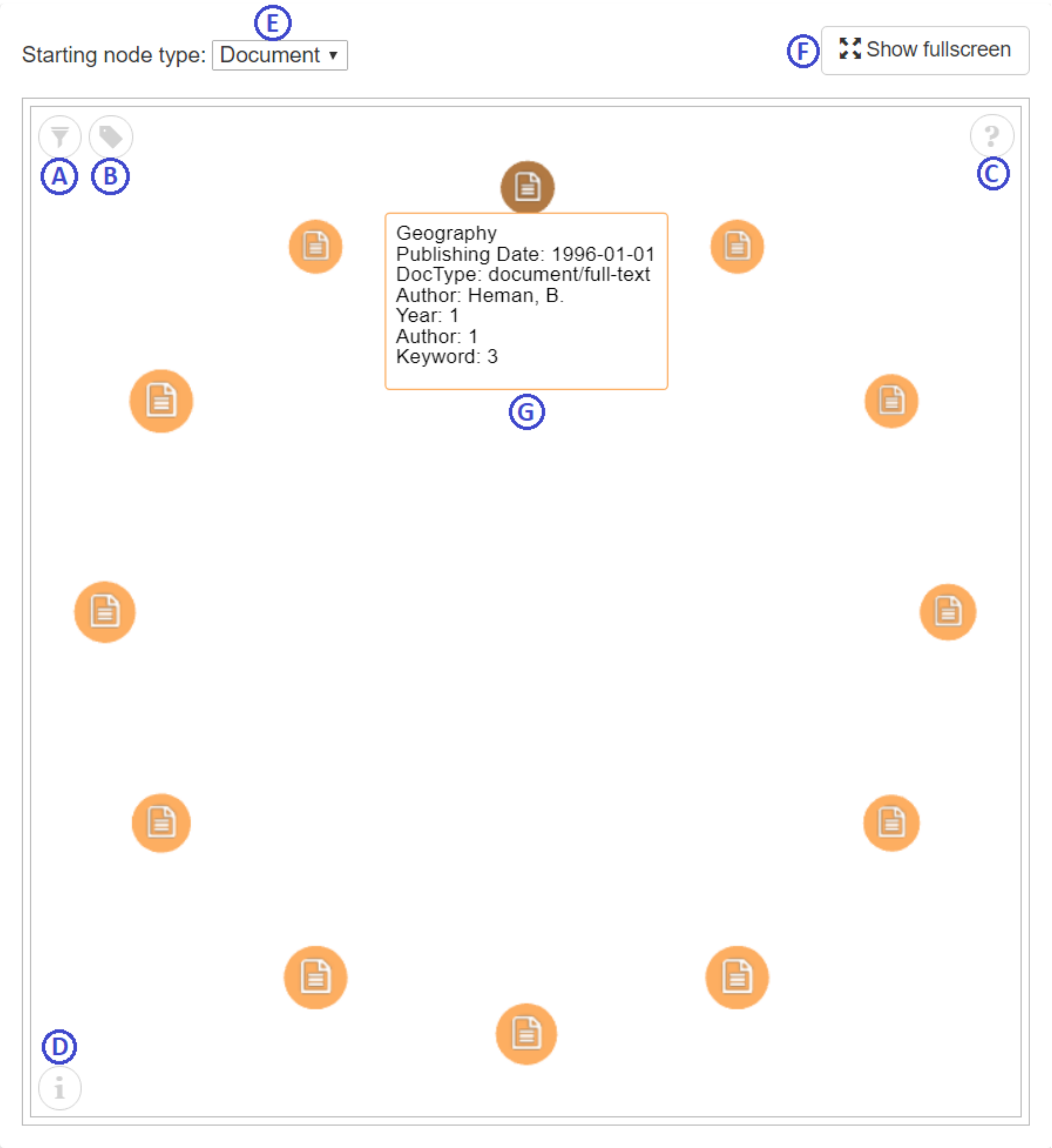


Figure 7: Concept Graph – Node legend

As shown in Figure 8, the initial starting node type is Document (Figure 8E), and the top (12) results are ordered in a circular way. The starting node type can be changed to any one of the available node types. Hovering over a node shows additional information (Figure 8G), e.g. the name of the document, the publishing date and what kind of other nodes are connected to it. The exploration process in the Concept Graph is further supported by the possibility to filter nodes (Figure 8A), toggle all the labels in the graph (Figure 8B) and display statistical information about the graph (Figure 8D). The Concept Graph can also be inspected in more detail by switching the view to Fullscreen (Figure 8F). All of the available interactions in the Concept Graph are listed in the help dialog, which is available by clicking on the “Help” toggle button (Figure 8C).



Geography
Publishing Date: 1996-01-01
DocType: document/full-text
Author: Heman, B.
Year: 1
Author: 1
Keyword: 3

Figure 8: Concept Graph – UI Elements: A) Filters, B) Labels toggle, C) Help dialog toggle, D) Graph information toggle, E) Starting node type, F) Fullscreen button, G) Additional node information

Clicking on a node, will open all the nodes which are directly connected to it and draw connections (if they are any) to already existing nodes. If all of its connected neighbor nodes have been displayed, the node will get a ring indicating this (Figure 9).

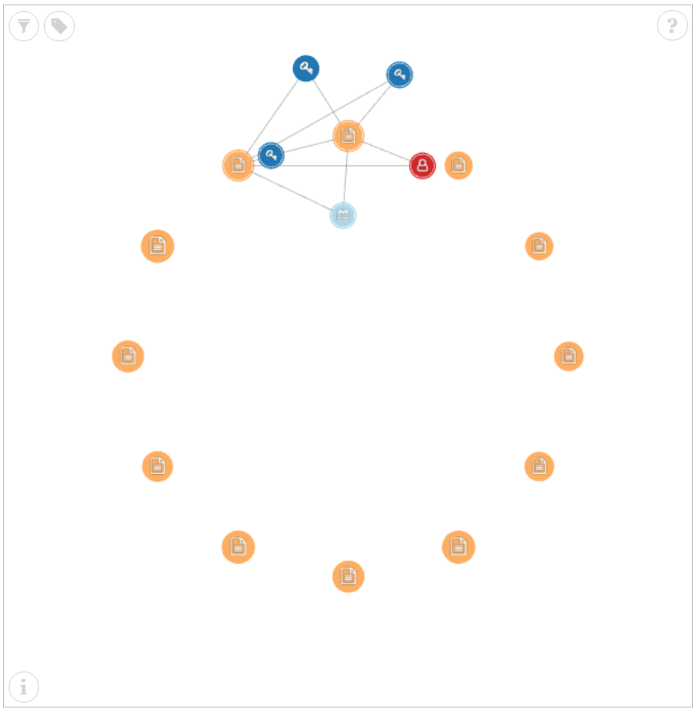


Figure 9: Concept Graph – Opened node with completion ring and connections drawn to an existing one

Hovering over each individual node to see additional information might be very tedious, therefore it is possible to enable/disable all the node labels by clicking on the toggle label button in the top left (Figure 9). Still, hovering over a node will display additional information. Note that, to avoid overlap, only a limited number of labels can be displayed on screen at once. If a node's label is not visible, zooming in might reveal it.

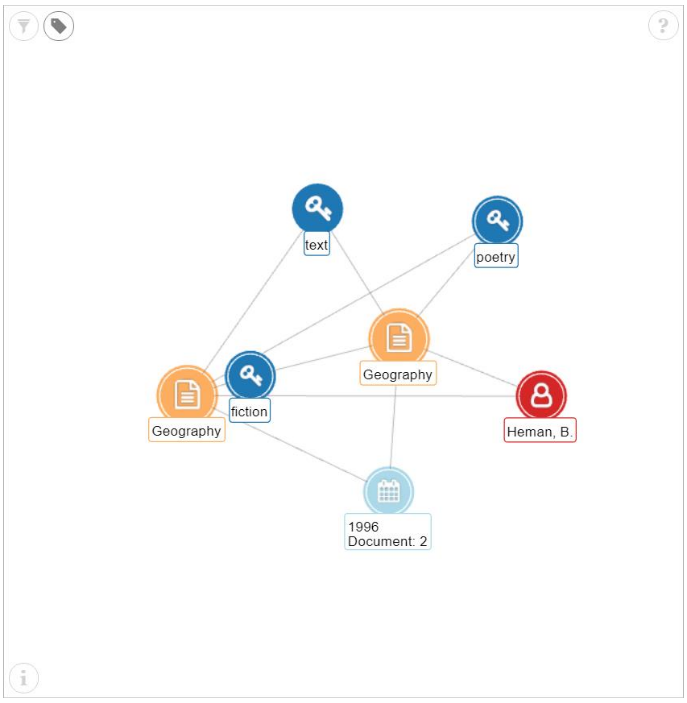
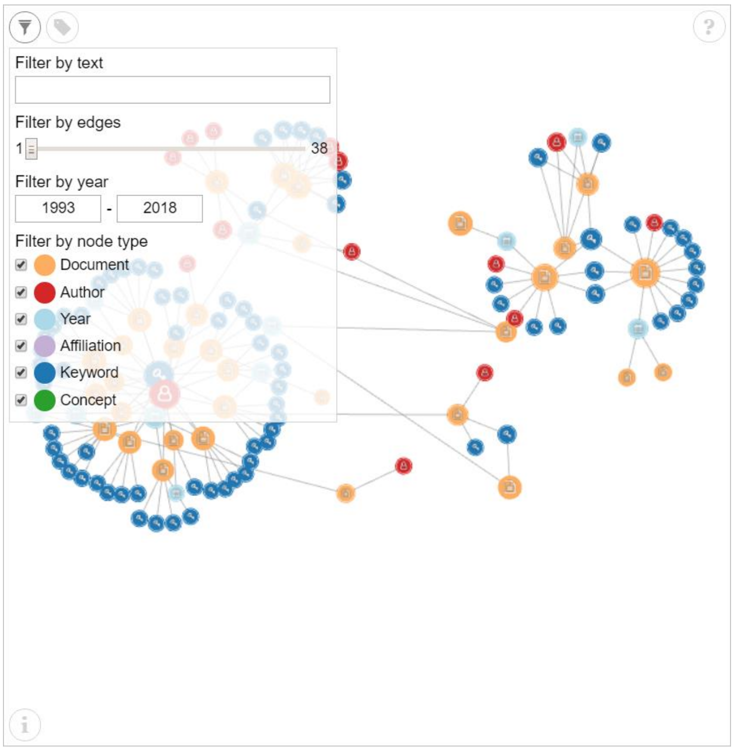


Figure 10: Concept Graph – Enable labels

Additionally, if too many nodes are displayed on the screen, it is possible to reduce the number by using the graph filter functionality. The filtering menu is in the top left next to the "Toggle labels" button. You can filter the nodes by label, by number of edges, by year or by node type (Figure 10).

A document of interest can be accessed by double-clicking on the respective document node.



Filter by text

Filter by edges

1 38

Filter by year

1993 - 2018

Filter by node type

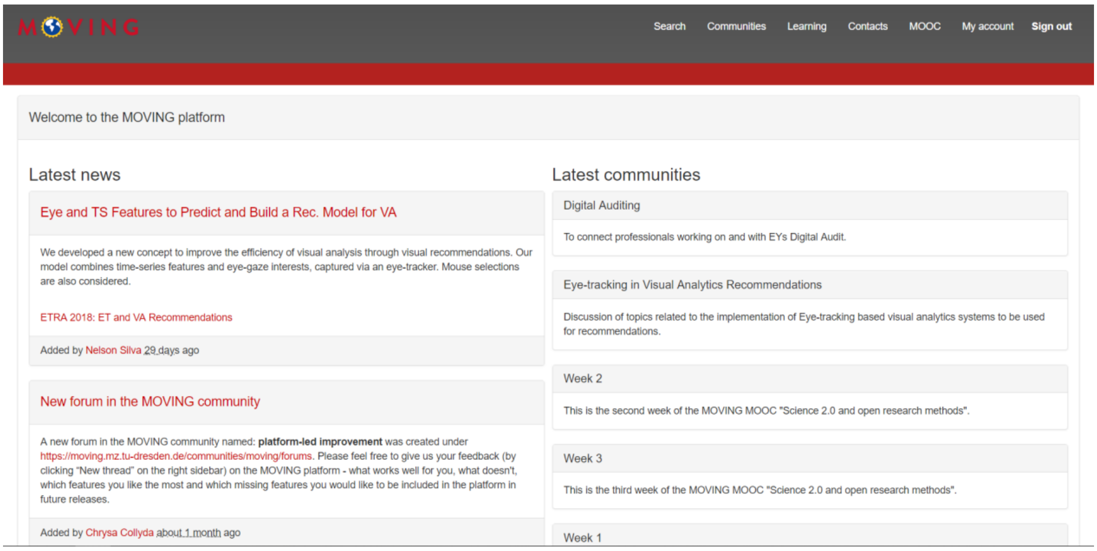
- ☑ ● Document
- ☑ ● Author
- ☑ ● Year
- ☑ ● Affiliation
- ☑ ● Keyword
- ☑ ● Concept

Figure 11: Concept Graph – Opened filter menu

13 Appendix - User study: introduction to the MOVING platform (faceted search interface)

Introduction to the MOVING Platform

After you have logged in into the MOVING Platform, you see the page as shown in Figure 1.



Welcome to the MOVING platform

Latest news

Eye and TS Features to Predict and Build a Rec. Model for VA

We developed a new concept to improve the efficiency of visual analysis through visual recommendations. Our model combines time-series features and eye-gaze interests, captured via an eye-tracker. Mouse selections are also considered.

ETRA 2018: ET and VA Recommendations

Added by Nelson Silva 29 days ago

New forum in the MOVING community

A new forum in the MOVING community named: **platform-led improvement** was created under https://moving.mz.tu-dresden.de/communities/moving/forums. Please feel free to give us your feedback (by clicking "New thread" on the right sidebar) on the MOVING platform - what works well for you, what doesn't, which features you like the most and which missing features you would like to be included in the platform in future releases.

Added by Chrysa Collyda about 1 month ago

Latest communities

Digital Auditing

To connect professionals working on and with EYs Digital Audit.

Eye-tracking in Visual Analytics Recommendations

Discussion of topics related to the implementation of Eye-tracking based visual analytics systems to be used for recommendations.

Week 2

This is the second week of the MOVING MOOC "Science 2.0 and open research methods".

Week 3

This is the third week of the MOVING MOOC "Science 2.0 and open research methods".

Week 1

Figure 1: Starting page of the MOVING Platform

The focus of this evaluation will lie in the search functionality of the platform and the exploration of the result set; hence we will not explain all the features of the platform, but only the ones which are relevant for you to perform your evaluation tasks.

Clicking on the "Search" button in the navigation bar, at the top of the page, will bring you to the simple search page (Figure 2). This is the main entry point for performing searches in the MOVING Platform.

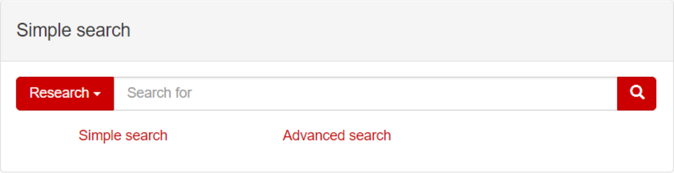
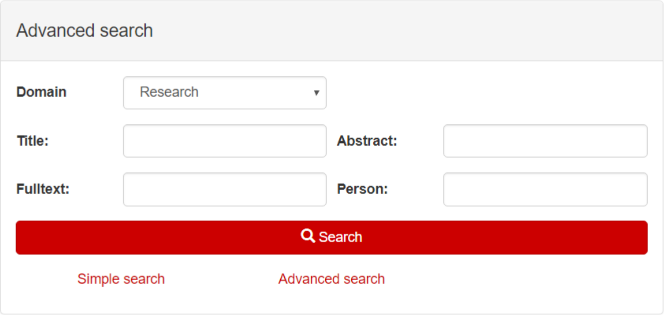


Figure 2: Simple search

The Advanced search (Figure 3) offers an even more fine-grained way to search in the platform. You can search there explicitly for an author, and select if you want to search the titles, the abstracts, or the bodies of the documents.



Advanced search

Domain: Research

Title:

Abstract:

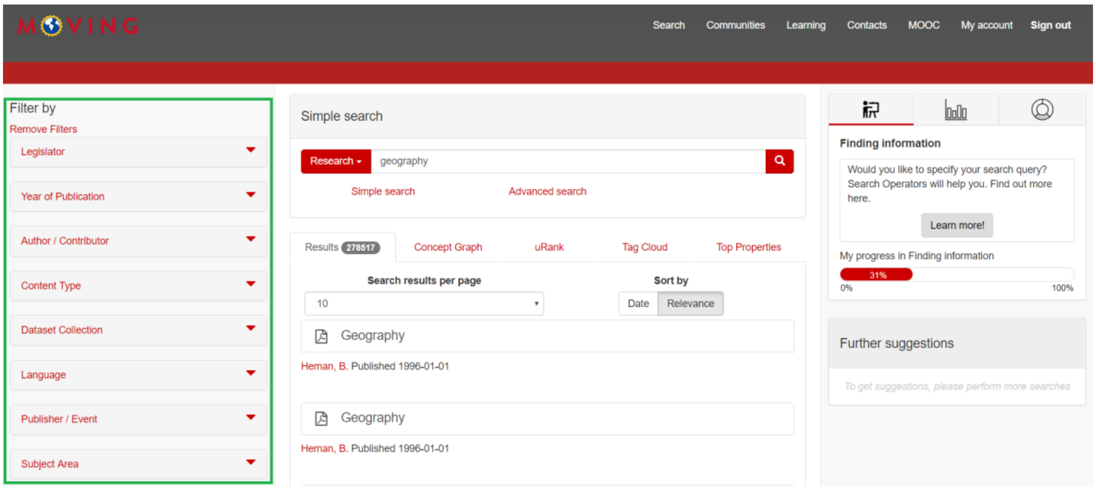
Fulltext:

Person:

Figure 3: Advanced search

After you have performed a search, you will be presented with the results page (Figure 4). The results page offers a wide variety of options to further explore the retrieved documents. On one hand, there is a classic results list, and on the other, there are various visualizations which allow you to visually explore and analyze the result set.

In your tasks, you will be asked to use the "Filter by" functionality (highlighted in green) to analyze the result set and retrieve some specific information.



Filter by

Remove Filters

- Legislator
- Year of Publication
- Author / Contributor
- Content Type
- Dataset Collection
- Language
- Publisher / Event
- Subject Area

Simple search

Research ▾ geography

Search results per page: 10

Sort by: Date | Relevance

Geography

Heman, B. Published 1996-01-01

Geography

Heman, B. Published 1996-01-01

Finding information

Would you like to specify your search query? Search Operators will help you. Find out more here.

Learn more!

My progress in Finding information

31%

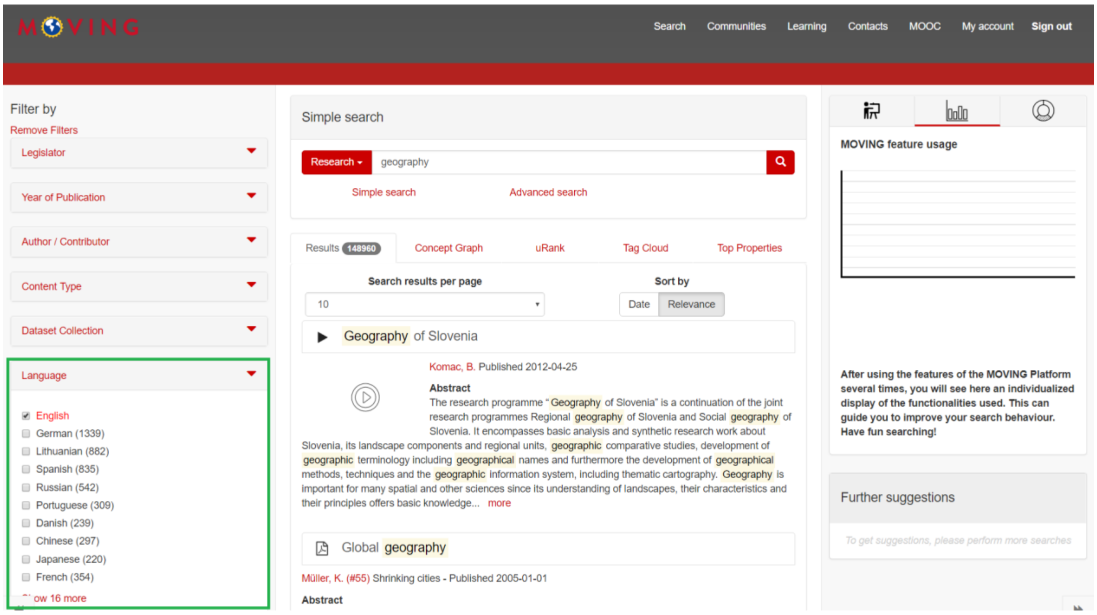
0% 100%

Further suggestions

To get suggestions, please perform more searches

Figure 4: Results page – The "Filter by" functionality (green), can be used to reduce the retrieved documents

With the "Filter by" panel you can further narrow down your search. This does not only affect the classic result list, but also the visualizations. For example, as shown in Figure 5, you can open the "Language" dropdown and select only documents and results that are in English.



Filter by
Remove Filters
Legislator
Year of Publication
Author / Contributor
Content Type
Dataset Collection
Language
☑ English
□ German (1339)
□ Lithuanian (882)
□ Spanish (835)
□ Russian (542)
□ Portuguese (309)
□ Danish (239)
□ Chinese (297)
□ Japanese (220)
□ French (354)

Simple search

geography

Results 148960

Search results per page: 10

Sort by: Date | Relevance

Geography of Slovenia

Komac, B. Published 2012-04-25

Abstract
The research programme "Geography of Slovenia" is a continuation of the joint research programmes Regional geography of Slovenia and Social geography of Slovenia. It encompasses basic analysis and synthetic research work about Slovenia, its landscape components and regional units, geographic comparative studies, development of geographic terminology including geographical names and furthermore the development of geographical methods, techniques and the geographic information system, including thematic cartography. Geography is important for many spatial and other sciences since its understanding of landscapes, their characteristics and their principles offers basic knowledge... more

Global geography

Müller, K. (#55) Shrinking cities - Published 2005-01-01

Abstract

MOVING feature usage

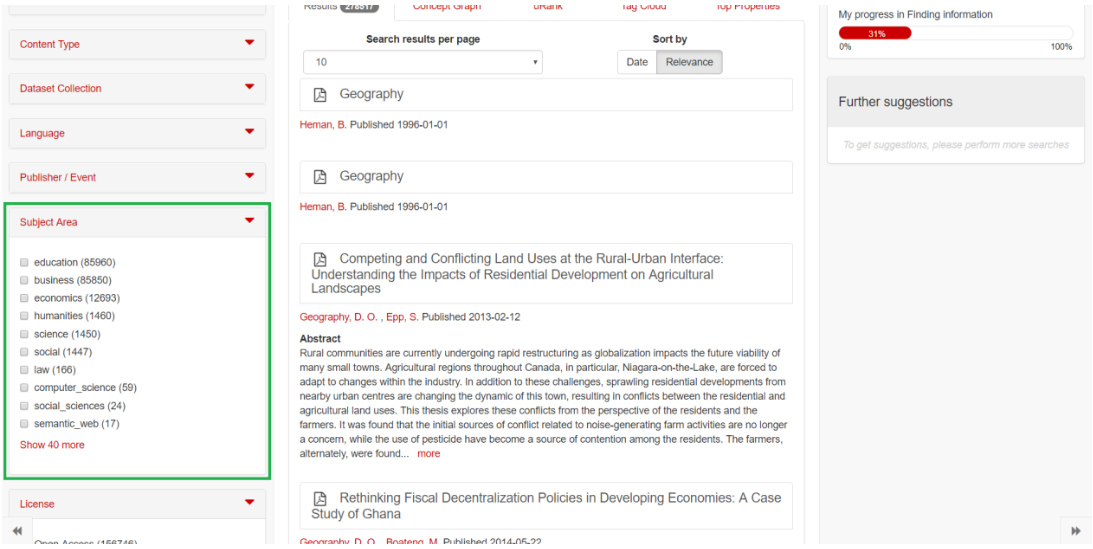
After using the features of the MOVING Platform several times, you will see here an individualized display of the functionalities used. This can guide you to improve your search behaviour. Have fun searching!

Further suggestions

To get suggestions, please perform more searches

Figure 5: Results page – Only results in English are displayed

In addition to simple filtering, this panel can also be used to analyze what kind of data has been retrieved. Figure 6 shows the opened "Subject Area" dropdown of the "Filter by" panel. As it can be seen, the retrieved documents are mostly from the area of "education", "business" and "economics".



Content Type
Dataset Collection
Language
Publisher / Event
Subject Area
□ education (85960)
□ business (85850)
□ economics (12693)
□ humanities (1460)
□ science (1450)
□ social (1447)
□ law (166)
□ computer_science (59)
□ social_sciences (24)
□ semantic_web (17)
License
Open Access (156746)

Search results per page: 10

Sort by: Date | Relevance

Geography

Heman, B. Published 1996-01-01

Geography

Heman, B. Published 1996-01-01

Competing and Conflicting Land Uses at the Rural-Urban Interface: Understanding the Impacts of Residential Development on Agricultural Landscapes

Geography, D. O. , Epp, S. Published 2013-02-12

Abstract
Rural communities are currently undergoing rapid restructuring as globalization impacts the future viability of many small towns. Agricultural regions throughout Canada, in particular, Niagara-on-the-Lake, are forced to adapt to changes within the industry. In addition to these challenges, sprawling residential developments from nearby urban centres are changing the dynamic of this town, resulting in conflicts between the residential and agricultural land uses. This thesis explores these conflicts from the perspective of the residents and the farmers. It was found that the initial sources of conflict related to noise-generating farm activities are no longer a concern, while the use of pesticide have become a source of contention among the residents. The farmers, alternately, were found... more

Rethinking Fiscal Decentralization Policies in Developing Economies: A Case Study of Ghana

Geography, D. O. , Boateng, M. Published 2014-05-22

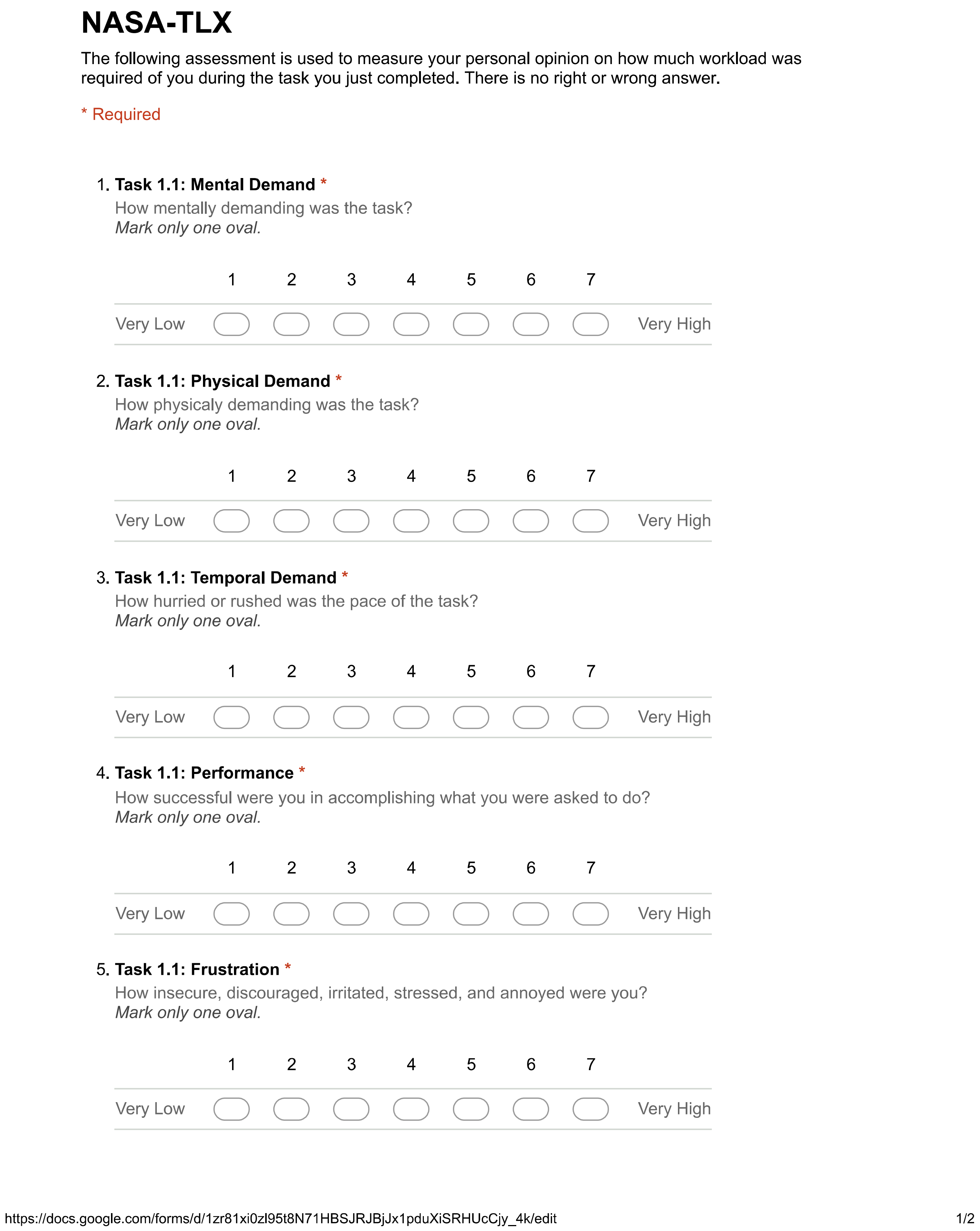
My progress in Finding information
31%
0%
100%

Further suggestions

To get suggestions, please perform more searches

Figure 6: Results page – The "Subject Area" dropdown in the "Filter by" panel shows from which area the retrieved documents are, and allow to limit the search only to certain area

14 Appendix - User study: NASA-TLX questionnaire



NASA-TLX

The following assessment is used to measure your personal opinion on how much workload was required of you during the task you just completed. There is no right or wrong answer.

* Required

1. **Task 1.1: Mental Demand** *

How mentally demanding was the task?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

2. **Task 1.1: Physical Demand** *

How physicaly demanding was the task?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

3. **Task 1.1: Temporal Demand** *

How hurried or rushed was the pace of the task?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

4. **Task 1.1: Performance** *

How successful were you in accomplishing what you were asked to do?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

5. **Task 1.1: Frustration** *

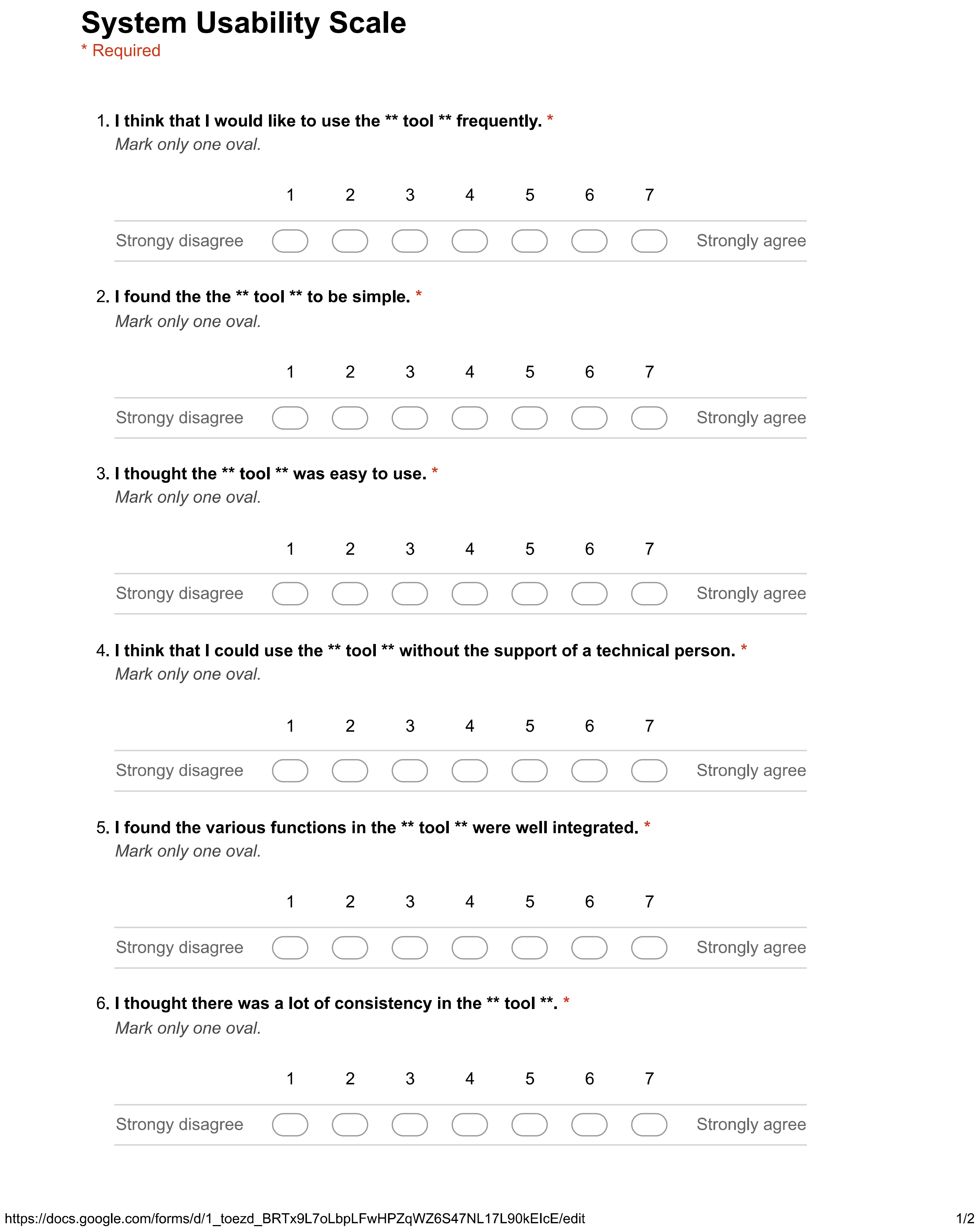
How insecure, discouraged, irritated, stressed, and annoyed were you?
Mark only one oval.

	1	2	3	4	5	6	7	
Very Low	○	○	○	○	○	○	○	Very High

https://docs.google.com/forms/d/1zr81xi0zl95t8N71HBSJRJBjJx1pduXiSRHUcCjy_4k/edit 1/2

6. **Task 1.1: Any comments? What was good / bad / unexpected / difficult?**

15 Appendix - User study: system usability scale questionnaire



System Usability Scale

* Required

1. **I think that I would like to use the ** tool ** frequently.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

2. **I found the the ** tool ** to be simple.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

3. **I thought the ** tool ** was easy to use.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

4. **I think that I could use the ** tool ** without the support of a technical person.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

5. **I found the various functions in the ** tool ** were well integrated.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

6. **I thought there was a lot of consistency in the ** tool **.** *
Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

https://docs.google.com/forms/d/1_toezd_BRTx9L7oLbpLFwHPZqWZ6S47NL17L90kEIcE/edit 1/2

7. **I would imagine that most people would learn to use the ** tool ** very quickly.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

8. **I found the ** tool ** very intuitive.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

9. **I felt very confident using the ** tool **.** *

Mark only one oval.

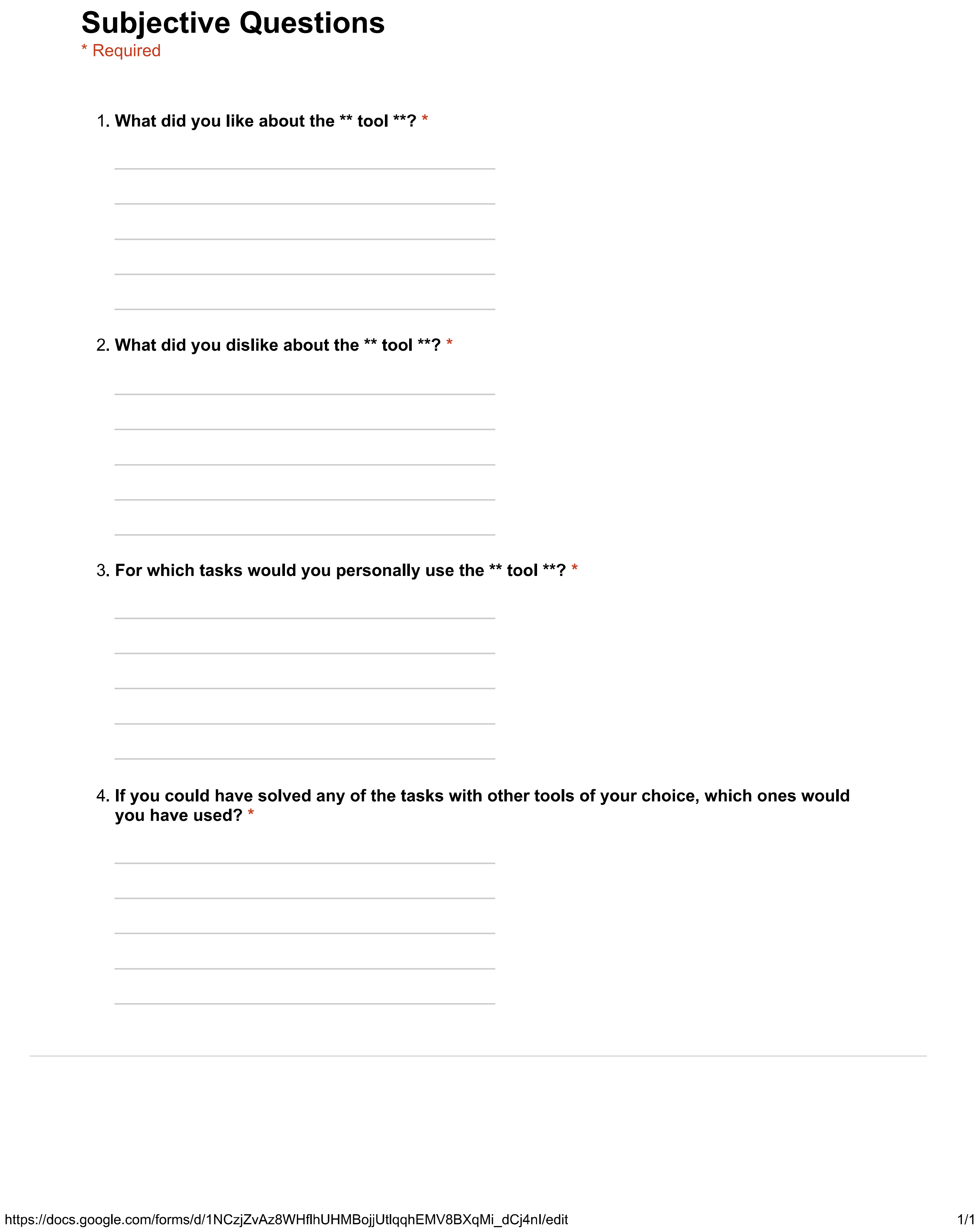
	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

10. **I could use the ** tool ** without having to learn anything new.** *

Mark only one oval.

	1	2	3	4	5	6	7	
Strongy disagree	○	○	○	○	○	○	○	Strongly agree

16 Appendix - User study: subjective questions



Subjective Questions

* Required

1. What did you like about the ** tool **? *

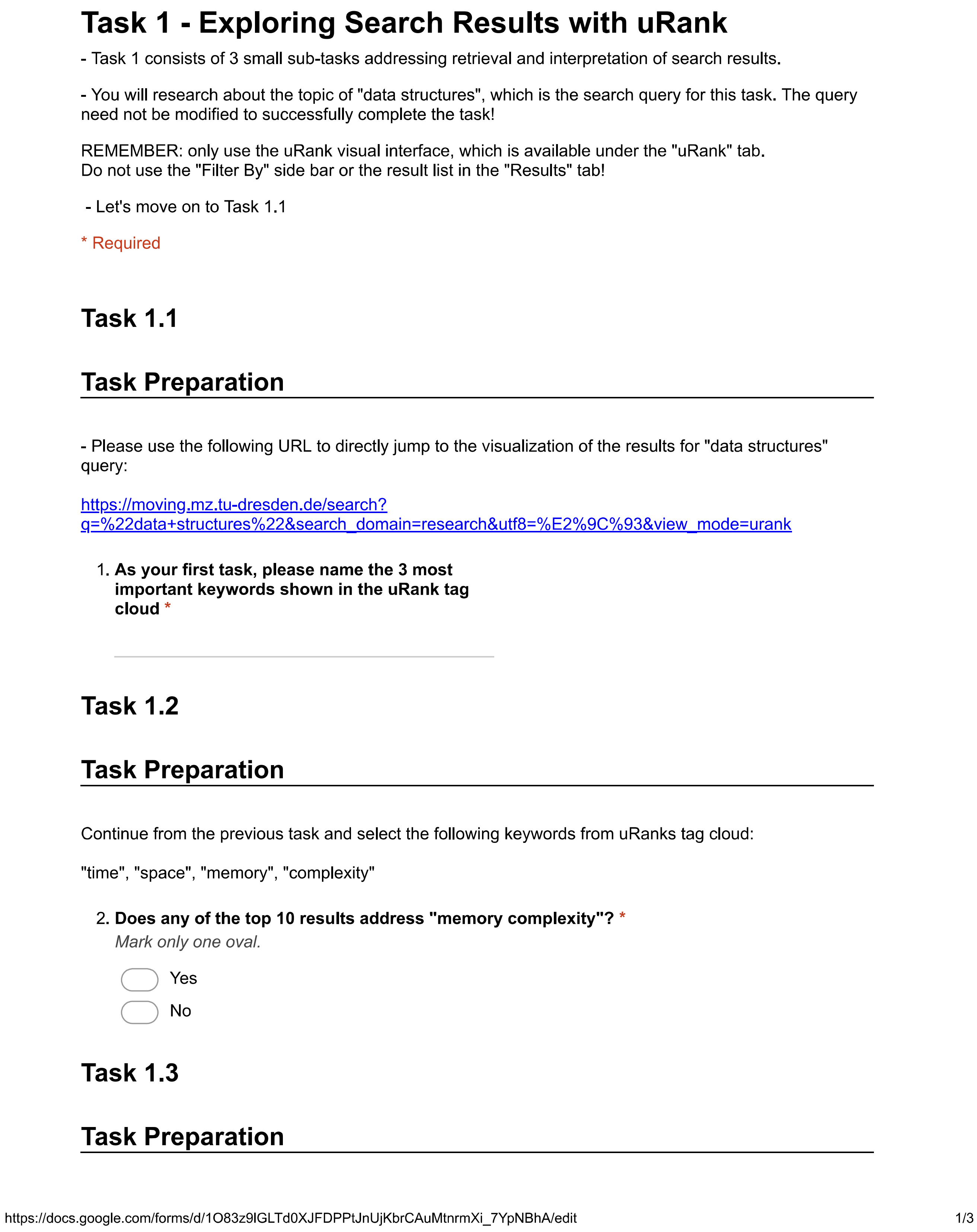
2. What did you dislike about the ** tool **? *

3. For which tasks would you personally use the ** tool **? *

4. If you could have solved any of the tasks with other tools of your choice, which ones would you have used? *

https://docs.google.com/forms/d/1NCzjZvAz8WHflhUHMBojjUtlqqhEMV8BXqMi_dCj4nl/edit 1/1

17 Appendix - User study: tasks for use case 1V



Task 1 - Exploring Search Results with uRank

- Task 1 consists of 3 small sub-tasks addressing retrieval and interpretation of search results.

- You will research about the topic of "data structures", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the uRank visual interface, which is available under the "uRank" tab.
Do not use the "Filter By" side bar or the result list in the "Results" tab!

- Let's move on to Task 1.1

* Required

Task 1.1

Task Preparation

- Please use the following URL to directly jump to the visualization of the results for "data structures" query:

https://moving.mz.tu-dresden.de/search?q=%22data+structures%22&search_domain=research&utf8=%E2%9C%93&view_mode=urank

1. **As your first task, please name the 3 most important keywords shown in the uRank tag cloud** *

Task 1.2

Task Preparation

Continue from the previous task and select the following keywords from uRanks tag cloud:

"time", "space", "memory", "complexity"

2. **Does any of the top 10 results address "memory complexity"?** *

Mark only one oval.

○ Yes

○ No

Task 1.3

Task Preparation

https://docs.google.com/forms/d/1O83z9lGLTd0XJFDPPtJnUjKbrCAuMtnrmXi_7YpNBhA/edit 1/3

Continue from the previous task and additionally add the keyword "algorithm", so that you have the following keywords selected:

"time", "space", "memory", "complexity", "algorithm"

3. **Name the title of a document from the top 10 results, in which "algorithm" is most dominant** *

4. **Optional question: Do you think you would have found that document without uRank? Could you explain your answer.**

Task 2 - Exploring Relations with the Concept Graph

- Task 2 consists of 3 small sub-tasks addressing exploration of relationships between search results and their properties.

- You will research about the topic of "data science", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the Concept Graph visualization, which is available under the "Concept Graph" tab.
Do not use the "Filter By" side bar or the result list in the "Results" tab!

- Let's move on to Task 2.1

Task 2.1

Task Preparation

- Please use the following URL to directly go to the page with the Concept Graph for the query "data science"

https://moving.mz.tu-dresden.de/search?filters%5Blanguage%5D%5B%5D=en&q=data+science&search_domain=research&utf8=%E2%9C%93&view_mode=concept_graph

- After the page has loaded, select "Keyword" as the Starting Node Type (in the drop-down list).

- You can close the help text dialog by clicking on the "Close" button

- To make it easier to identify what the nodes are representing, please enable the labels by clicking on the label symbol in the top left corner of the graph window (second button from the left)

5. **Explore the Concept Graph and name the title of one document related to the "social science" node.** *

6. **Is the document you picked the most relevant to social science** *

Mark only one oval.

○ Yes

○ No

Task 2.2

Task Preparation

Continue from the previous task

7. **Name one author in connection with the document node "A perspective on social science data management"** *

____

Task 2.3

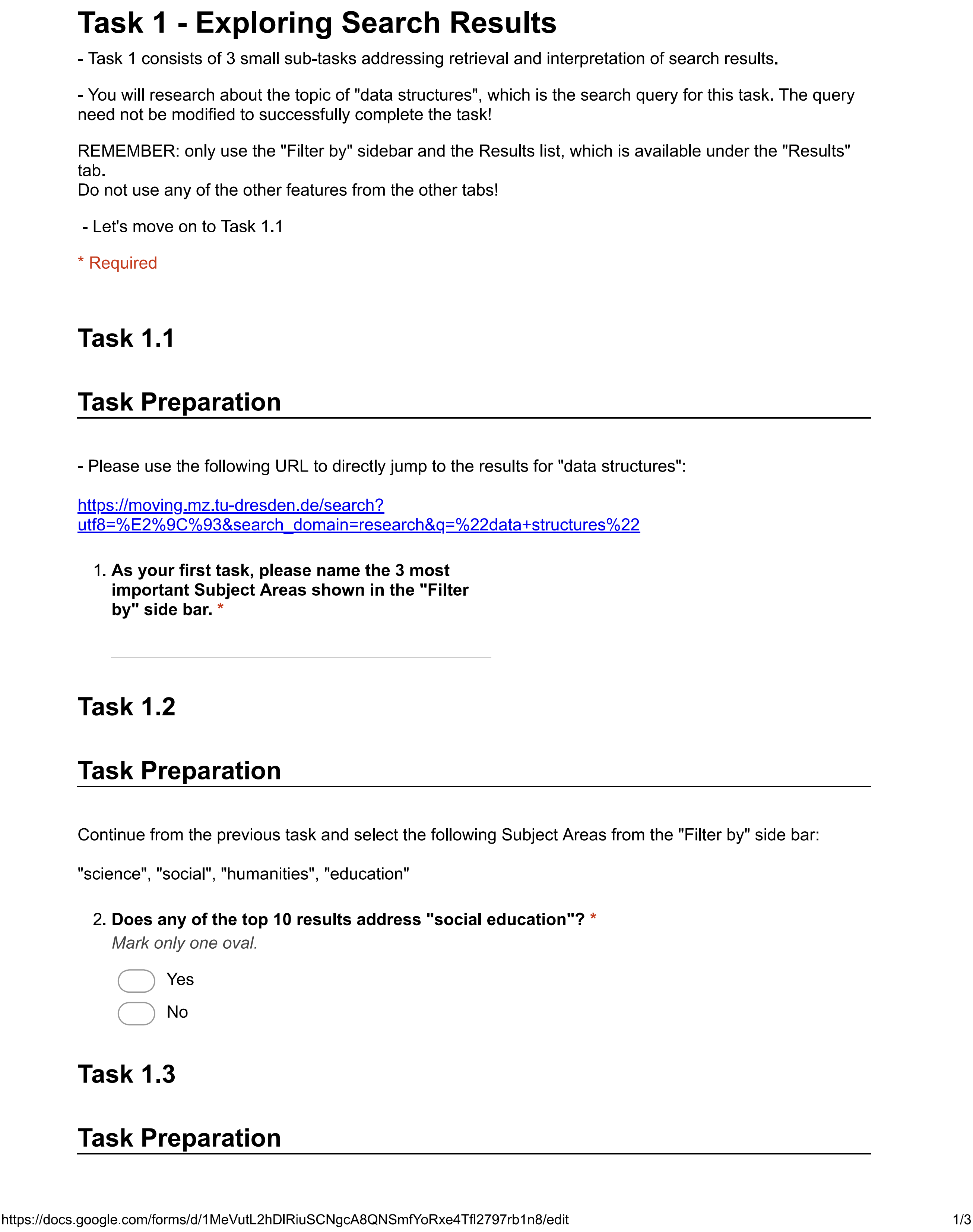
Task Preparation

Continue from the previous task

8. **Name another publication from the author discovered from the previous task.** *

____

18 Appendix - User study: tasks for use case 1F



Task 1 - Exploring Search Results

- Task 1 consists of 3 small sub-tasks addressing retrieval and interpretation of search results.

- You will research about the topic of "data structures", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the "Filter by" sidebar and the Results list, which is available under the "Results" tab.
Do not use any of the other features from the other tabs!

- Let's move on to Task 1.1

* Required

Task 1.1

Task Preparation

- Please use the following URL to directly jump to the results for "data structures":

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&q=%22data+structures%22

1. **As your first task, please name the 3 most important Subject Areas shown in the "Filter by" side bar.** *

Task 1.2

Task Preparation

Continue from the previous task and select the following Subject Areas from the "Filter by" side bar:

"science", "social", "humanities", "education"

2. **Does any of the top 10 results address "social education"?** *
   Mark only one oval.

   ○ Yes
   ○ No

Task 1.3

Task Preparation

https://docs.google.com/forms/d/1MeVutL2hDlRiuSCNgcA8QNSmfYoRxe4Tfl2797rb1n8/edit 1/3

Continue from the previous task and additionally add the Subject Area "big_data", so that you have the following Subject Areas selected:

"science", "social", "humanities", "education", "big_data"

3. **Name a title of the result from the top 10, where the word "data" is most dominant** *

Task 2 - Exploring Relations in the Results

- Task 2 consists of 3 small sub-tasks addressing exploration of relationships between search results and their properties.

- You will research about the topic of "data science", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the "Filter by" sidebar and the Results list, which is available under the "Results" tab.
Do not use any of the other features from the other tabs!

- Let's move on to Task 2.1

Task 2.1

Task Preparation

- Please use the following URL to directly go to the results page for the query "data science"

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&view_mode=results&q=data+science&filters%5Blanguage%5D%5B%5D=en

4. **Explore and select the Facets (Filters) "social" and "science" in the "Filter by" sidebar. Name the title of one result related to social science.** *

5. **Is the result you picked the most relevant one to social science?** *

Mark only one oval.

○ Yes

○ No

Task 2.2

Task Preparation

Use the following URL to go directly to the page with the pre-selected facets from the previous task:

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&view_mode=results&q=data+science&filters%5Blanguage%5D%5B%5D=en&filters%5Bsubjects%5D%5B%5D=science&filters%5Bsubjects%5D%5B%5D=social

6. **Name one author of the result "A perspective on social science data management"** *

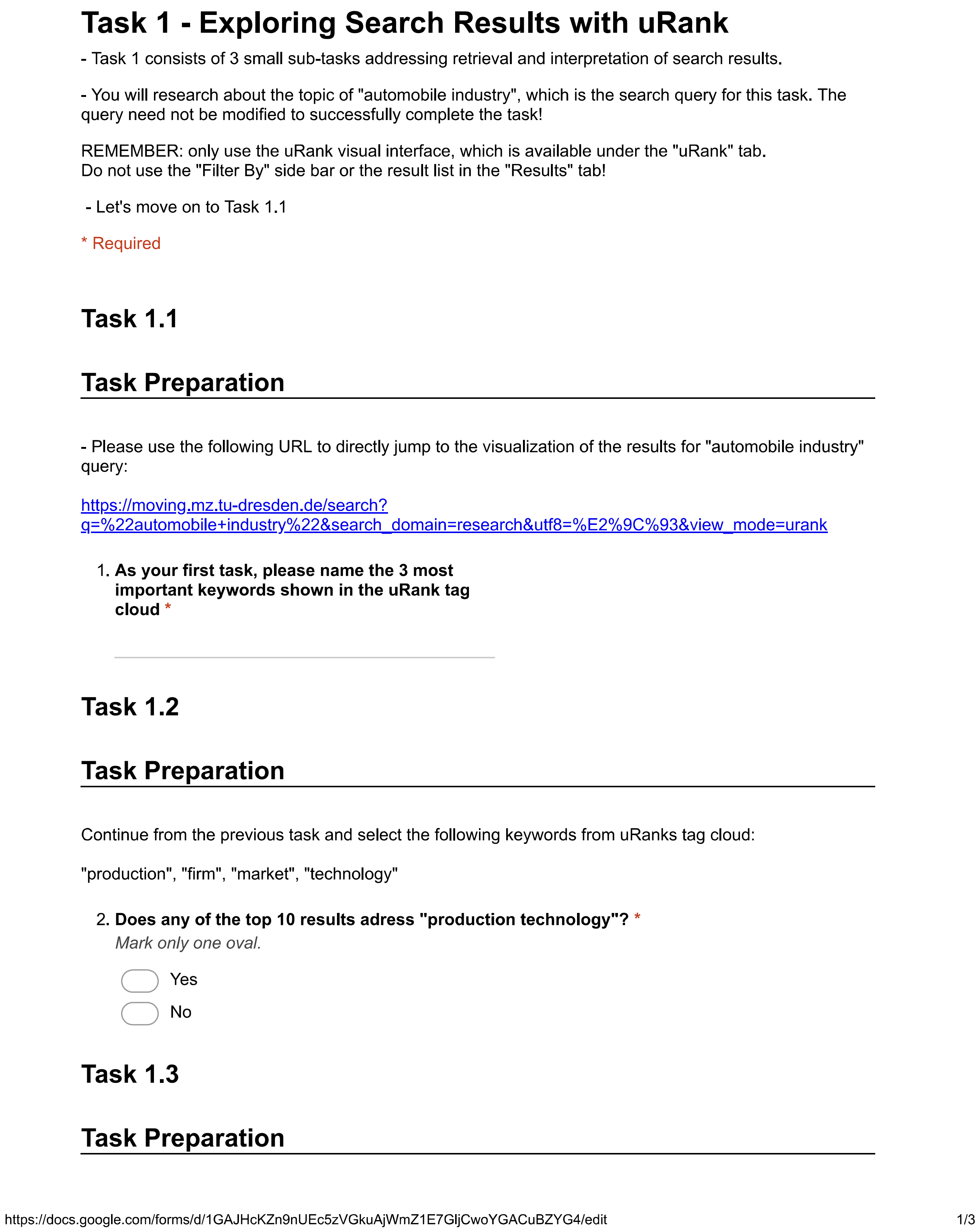
Task 2.3

Task Preparation

Continue from the previous task

7. **Name another publication from the author discovered in the previous task.** *

19 Appendix - User study: tasks for use case 2V



Task 1 - Exploring Search Results with uRank

- Task 1 consists of 3 small sub-tasks addressing retrieval and interpretation of search results.

- You will research about the topic of "automobile industry", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the uRank visual interface, which is available under the "uRank" tab.
Do not use the "Filter By" side bar or the result list in the "Results" tab!

- Let's move on to Task 1.1

* Required

Task 1.1

Task Preparation

- Please use the following URL to directly jump to the visualization of the results for "automobile industry" query:

https://moving.mz.tu-dresden.de/search?q=%22automobile+industry%22&search_domain=research&utf8=%E2%9C%93&view_mode=urank

1. **As your first task, please name the 3 most important keywords shown in the uRank tag cloud** *

Task 1.2

Task Preparation

Continue from the previous task and select the following keywords from uRanks tag cloud:

"production", "firm", "market", "technology"

2. **Does any of the top 10 results adress "production technology"?** *
   Mark only one oval.

   ○ Yes
   ○ No

Task 1.3

Task Preparation

https://docs.google.com/forms/d/1GAJHcKZn9nUEc5zVGkuAjWmZ1E7GljCwoYGACuBZYG4/edit 1/3

Continue from the previous task and additionally add the keyword "development", so that you have the following keywords selected:

"production", "firm", "market", "technology", "development"

3. **Name the title of a document from the top 10 results, in which "market" is most dominant** *

4. **Optional question: Do you think you would have found that document without uRank? Could you explain your answer.** *

Task 2 - Exploring Relations with the Concept Graph

- Task 2 consists of 3 small sub-tasks addressing exploration of relationships between search results and their properties.

- You will research about the topic of "automobile industry", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the Concept Graph visualization, which is available under the "Concept Graph" tab.
Do not use the "Filter By" side bar or the result list in the "Results" tab!

- Let's move on to Task 2.1

Task 2.1

Task Preparation

- Please use the following URL to directly go to the page with the Concept Graph for the query "automobile industry"

https://moving.mz.tu-dresden.de/search?filters%5Blanguage%5D%5B%5D=en&q=automobile+industry&search_domain=research&utf8=%E2%9C%93&view_mode=concept_graph

- After the page has loaded, select "Keyword" as the Starting Node Type (in the drop-down list).

- You can close the help text dialog by clicking on the "Close" button

- Before you start, please enable node labels by clicking on the label symbol in the top left corner of the graph window (second button from the left). This will make it easier to identify what the nodes are representing.

5. **Explore the Concept Graph and name the title of one document related to the "usa" node** *

6. **Is the result you picked the most relevant one to usa?** *

   Mark only one oval.

   ○ Yes

   ○ No

Task 2.2

Task Preparation

Continue from the previous task

7. **Name one author of the document node "Aspects of technology, competitiveness and capital formation in the automobile industry"** *

   ____

Task 2.3

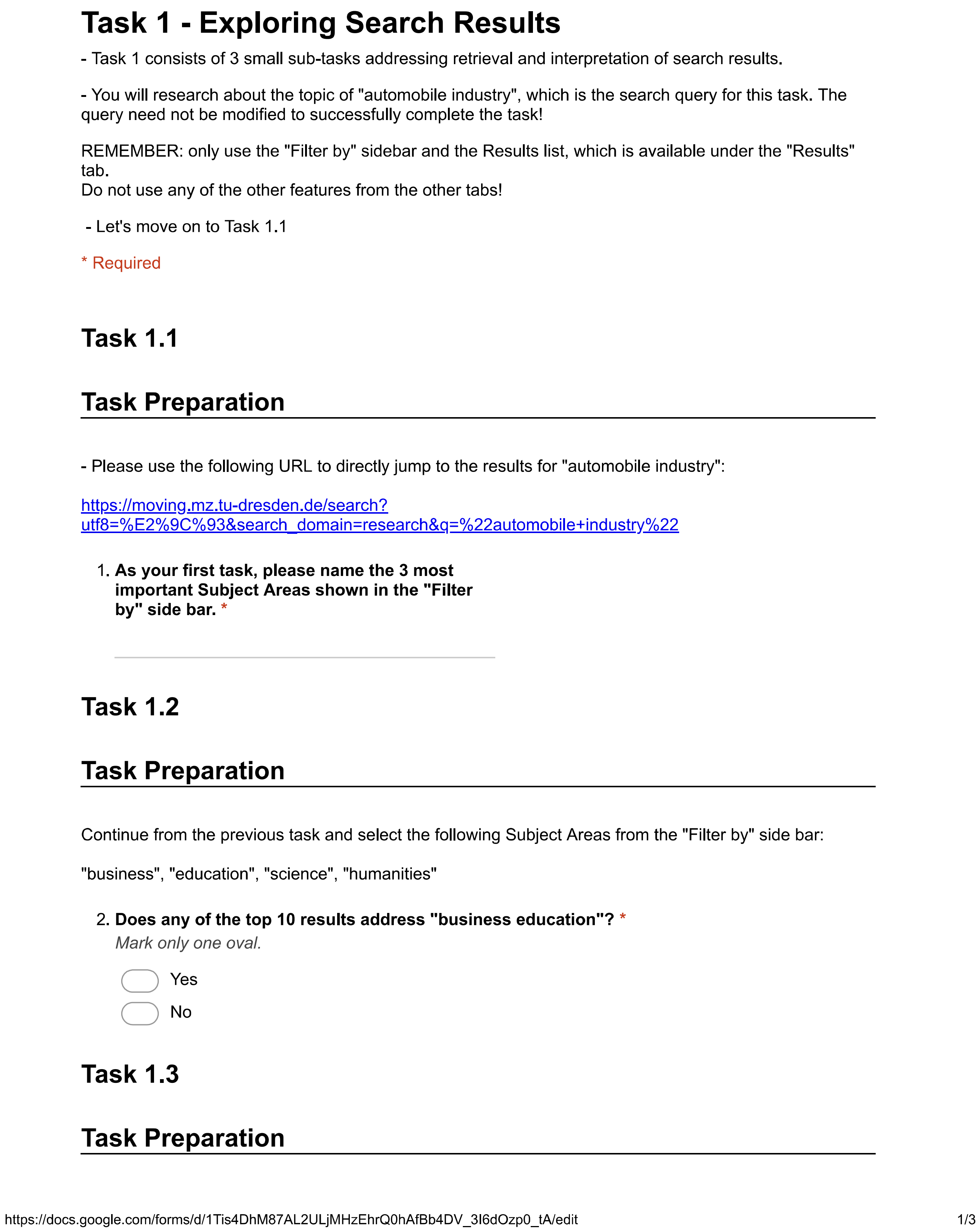
Task Preparation

Continue from the previous task

8. **Name another publication from the author discovered in the previous task.** *

   ____

20 Appendix - User study: tasks for use case 2F



Task 1 - Exploring Search Results

- Task 1 consists of 3 small sub-tasks addressing retrieval and interpretation of search results.

- You will research about the topic of "automobile industry", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the "Filter by" sidebar and the Results list, which is available under the "Results" tab.
Do not use any of the other features from the other tabs!

- Let's move on to Task 1.1

* Required

Task 1.1

Task Preparation

- Please use the following URL to directly jump to the results for "automobile industry":

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&q=%22automobile+industry%22

1. **As your first task, please name the 3 most important Subject Areas shown in the "Filter by" side bar.** *

Task 1.2

Task Preparation

Continue from the previous task and select the following Subject Areas from the "Filter by" side bar:

"business", "education", "science", "humanities"

2. **Does any of the top 10 results address "business education"?** *

   Mark only one oval.

   ○ Yes
   ○ No

Task 1.3

Task Preparation

https://docs.google.com/forms/d/1Tis4DhM87AL2ULjMHzEhrQ0hAfBb4DV_3I6dOzp0_tA/edit 1/3

Continue from the previous task and additionally add the Subject Area "economics", so that you have the following Subject Areas selected:

"business", "education", "humanties", "science", "economics"

3. **Name a title of the result from the top 10, where the word "industry" is most dominant** *

Task 2 - Exploring Relations in the Results

- Task 2 consists of 3 small sub-tasks addressing exploration of relationships between search results and their properties.

- You will research about the topic of "automobile industry", which is the search query for this task. The query need not be modified to successfully complete the task!

REMEMBER: only use the "Filter by" sidebar and the Results list, which is available under the "Results" tab.
Do not use any of the other features from the other tabs!

- Let's move on to Task 2.1

Task 2.1

Task Preparation

- Please use the following URL to directly go to the results page for the query "automobile industry"

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&view_mode=results&q=automobile+industry&filters%5Blanguage%5D%5B%5D=en

4. **Explore and select the Facet (Filter) "technology" in the "Filter by" sidebar. Name the title of one result related to technology.** *

5. **Is the result you picked the most relevant one to technology?** *

Mark only one oval.

○ Yes

○ No

Task 2.2

Task Preparation

Use the following URL to go directly to the page with the pre-selected facets from the previous task:

https://moving.mz.tu-dresden.de/search?utf8=%E2%9C%93&search_domain=research&view_mode=results&q=automobile+industry&filters%5Blanguage%5D%5B%5D=en&filters%5Bsubjects%5D%5B%5D=technology

6. **Name one author of the result "MDA Technology to Support China Aviation Industry"** *

Task 2.3

Task Preparation

Continue from the previous task

7. **Name another publication from the author discovered in the previous task.** *

References

Abdel-Qader, M., Scherp, A., & Vagliano, I. (2018). Analyzing the evolution of vocabulary terms and their impact on the LOD Cloud. In *Extended semantic web conference.*

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., & Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

Adhikari, R., & Agrawal, R. (2013). An introductory study on time series modeling and forecasting..

Adya, M., & Collopy, F. (1998). *How effective are neural networks at forecasting and prediction? journal of forecasting.* Elsevier.

Agerri, R., & Rigau, G. (2016). Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, *238*, 63–82.

Althoff, T., Borth, D., Hees, J., & Dengel, A. (2013). Analysis and forecasting of trending topics in online media streams. In *Proceedings of the 21st acm international conference on multimedia* (pp. 907–916).

Amatriain, X., Jaimes, A., Oliver, N., & Pujol, J. M. (2011). Data mining methods for recommender systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 39–71). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-0-387-85820-3_2 doi: 10.1007/978-0-387-85820-3_2

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

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Augenstein, I., Derczynski, L., & Bontcheva, K. (2017). Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, *44*, 61–83.

Baazizi, M. A., Ben Lahmar, H., Colazzo, D., Ghelli, G., & Sartiani, C. (2017). Schema inference for massive JSON datasets. In *EDBT* (pp. 222–233). OpenProceedings.org.

Banko, M., & Brill, E. (2001, January). Scaling to very very large corpora for natural language disambiguation. In *Proc. ACL* (pp. 26–33).

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997, July). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, *19*(7), 711–720.

Bengio, Y., Simard, P. Y., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, *5*(2), 157–166. Retrieved from https://doi.org/10.1109/72.279181;https://dblp.org/rec/bib/journals/tnn/BengioSF94 (Vanishing and exploding gradient problems in RNNs) doi: 10.1109/72.279181

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

Bird, S., Dale, R., Dorr, B. J., Gibson, B., Joseph, M. T., Kan, M.-Y., … Tan, Y. F. (2008). The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives, handbook i: The cognitive domain* (Vol. 19). New York: David McKay Co Inc.

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

Blume, T., & Scherp, A. (2018a). Towards flexible indices for distributed graph data: The formal schema-level index model FLuID. In *30th gi-workshop on foundations of databases.* CEUR-WS.org. Retrieved from https://dbs.cs.uni-duesseldorf.de/gvdb2018/wp-content/uploads/2018/05/GvDB2018_paper_4.pdf

Blume, T., & Scherp, A. (2018b). Towards flexible indices for distributed graph data: The formal schema-level index model fluid. In *Grundlagen von datenbanken* (Vol. 2126, pp. 23–28). CEUR-WS.org.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *TACL*, *5*, 135–146.

Brain, D., & Webb, G. (1999). On the effect of data set size on bias and variance in classification learning. In *Proceedings of the fourth australian knowledge acquisition workshop, university of new south wales* (pp. 117–128).

Brants, T., Chen, F., & Tsochantaridis, I. (2002). Topic-based document segmentation with probabilistic latent semantic analysis. In *Proc. of the 11th int. conf. on inf. and knowl. manag.* (pp. 211–218). New York, NY, USA: ACM.

Brinkman, W. P., Haakma, R., & Bouwhuis, D. G. (2009). The theoretical foundation and validity of a component-based usability questionnaire. *Behav. Inf. Technol.*, *28*(2), 121–137. Retrieved from `http://dx.doi.org/10.1080/01449290701306510` doi: 10.1080/01449290701306510

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467–479.

Cai, D., He, X., & Han, J. (2008, January). SRDA: an efficient algorithm for large-scale discriminant analysis. *IEEE Trans. Knowl. Data Eng.,*, *20*(1), 1–12.

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

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.

Chiticariu, L., Li, Y., & Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 827–832).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp).*

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Coleman, T. F., & Moré, J. J. (1983). Estimation of sparse jacobian matrices and graph coloring blems. *SIAM journal on Numerical Analysis*, *20*(1), 187–209.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Crone, S. F., & Kourentzes, N. (2009). Input-variable specification for neural networks-an analysis of forecasting low and high time series frequency. In *Neural networks, 2009. ijcnn 2009. international joint conference on* (pp. 619–626).

Cunningham, H. (2002). Gate, a general architecture for text engineering. *Computers and the Humanities*, *36*(2), 223–254.

Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

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.

Dividino, R. Q., Gottron, T., & Scherp, A. (2015). Strategies for efficiently keeping local linked open data caches up-to-date. In *ISWC.* Springer.

Dividino, R. Q., Gottron, T., Scherp, A., & Gröner, G. (2014). From changes to dynamics: Dynamics analysis of linked open data sources. In *Profiles@eswc* (Vol. 1151). CEUR-WS.org.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis..

Eckle-Kohler, J., Nghiem, T.-D., & Gurevych, I. (2013). Automatically assigning research methods to journal articles in the domain of social sciences. In *Proceedings of the 76th asis&t annual meeting: Beyond the cloud: Rethinking information boundaries* (p. 44).

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, *87*(3), 215–251. doi: 10.1037/0033-295X.87.3.215

Fan, Q., Zhang, D., Wu, H., & Tan, K. (2016). A general and parallel platform for mining co-movement patterns over large-scale trajectories. *PVLDB*, *10*(4), 313–324.

Feige, U. (2006). On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, *35*(4), 964–984.

Finkel, J. R., Grenager, T., & Manning, C. (2005a). 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).

Finkel, J. R., Grenager, T., & Manning, C. (2005b). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of the 43rd annual meeting on assoc. for computational* *linguistics* (pp. 363–370).

Florian, R., Ittycheriah, A., Jing, H., & Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on natural language learning at hlt-naacl 2003-volume 4* (pp. 168–171).

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

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

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016, december 5-10, 2016, barcelona, spain* (pp. 1019–1027). Retrieved from http://papers.nips.cc/paper/6241-a-theoretically-grounded-application-of-dropout-in-recurrent-neural-networks

Galke, L., Mai, F., Schelten, A., Brunsch, D., & Scherp, A. (2017). Using titles vs. full-text as source for automated semantic document annotation. In *Proceedings of knowledge capture, austin, texas, united states, december 4th-6th* (p. 9).

Gildea, D., Kan, M.-Y., Madnani, N., Teichmann, C., & Villalba, M. (2018). The acl anthology: Current state and future directions. In *Proceedings of workshop for nlp open source software (nlp-oss)* (pp. 23–28).

Gkalelis, N., & Mezaris, V. (2018). Logistic regression discriminant analysis.
(In Preparation)

Goldberg, D., & Shan, Y. (2015). The importance of features for statistical anomaly detection. In *Hotcloud.*

Golub, G. H., & Loan, C. F. V. (2013). *Matrix computations* (4th ed.). Baltimore, MD, USA: The Johns Hopkins University Press.

Gómez, S. N., Etcheverry, L., Marotta, A., & Consens, M. P. (2018). Findings from two decades of research on schema discovery using a systematic literature review. In *AMW* (Vol. 2100). CEUR-WS.org.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Sequence modeling: Recurrent and recursive nets.* MIT Press. (http://www.deeplearningbook.org)

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 *Proc. of the 7th k-cap* (pp. 105–108). ACM.

Grave, E., Mikolov, T., Joulin, A., & Bojanowski, P. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics, EACL 2017, valencia, spain, april 3-7, 2017, volume 2: Short papers* (pp. 427–431).

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*.

Guo, J., Che, W., Wang, H., & Liu, T. (2014). Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 110–120).

Gupta, S., & Manning, C. (2011). Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th international joint conference on natural language processing* (pp. 1–9).

Hachey, B., Radford, W., & Curran, J. R. (2011). Graph-based named entity linking with wikipedia. In *International conference on web information systems engineering* (pp. 213–226).

Hart, S. G., & Stavenland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–183). Elsevier. Retrieved from http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20000004342_1999205624.pdf

Harvey, A. C. (1990). *Forecasting, structural time series models and the kalman filter.* Cambridge university press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. In (Vol. 42, pp. 1082–1092). INFORMS.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., … Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 782–792).

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 A. Polleres et al. (Eds.), *Reasoning web. - int. summer school*. Springer.

Hou, C., Nie, F., Yi, D., & Tao, D. (2015, December). Discriminative embedded clustering: A framework for grouping high-dimensional data. *IEEE Trans. Neural Netw. Learn. Syst*, *26*(6), 1287–1299.

Howland, P., & Park, H. (2006, August). Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, *26*(8), 995–1006.

Hu, C., Cao, H., & Ke, C. (2014). Detecting influence relationships from graphs. In *SDM* (pp. 821–829). SIAM.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning, ICML 2015, lille, france, 6-11 july 2015* (pp. 448–456). Retrieved from http://jmlr.org/proceedings/papers/v37/ioffe15.html

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, *29*(3), 31–44.

Ji, H., Nothman, J., Hachey, B., et al. (n.d.). Overview of tac-kbp2014 entity discovery and linking tasks..

Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., & Hogan, A. (2013). Observing linked data dynamics. In *Proc. of the 10th eswc* (Vol. 7882, pp. 213–227). Springer.

Kageura, K., & Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *3*(2), 259–289.

Kansal, A., & Spezzano, F. (2017). A scalable graph-coarsening based index for dynamic graph databases. In *Proc. of cikm*. New York, NY, USA: ACM. doi: 10.1145/3132847.3133003

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, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, october 25-29, 2014, doha, qatar, A meeting of sigdat, a special interest group of the ACL* (pp. 1746–1751). Retrieved from http://aclweb.org/anthology/D/D14/D14-1181.pdf

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv: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. Retrieved from doi.org/10.1016/j.websem.2012.06.002

Koshorek, O., Cohen, A., Mor, N., Rotman, M., & Berant, J. (2018). Text segmentation as a supervised learning task. In *Proc. of the 2018 conf. of the north american ch. of the assoc. for comp. ling.: Human language technologies, volume 2 (short papers)* (pp. 469–473).

Kripke, S. A. (1972). Naming and necessity. In *Semantics of natural language* (pp. 253–355). Springer.

Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, *47*(260), 583–621. Retrieved from http://www.jstor.org/stable/2280779

Kulkarni, V., Mehdad, Y., & Chevalier, T. (2016). Domain adaptation for named entity recognition in online media with word embeddings. *CoRR*, *abs/1612.00148*. Retrieved from http://arxiv.org/abs/1612.00148

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition..

Le, H. T., Cerisara, C., & Denis, A. (2018). Do convolutional networks need to be deep for text classification? In *Proceedings of the aaai-18 workshop on affective content analysis, new orleans, louisiana, united states, february 3rd*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lewis, J. R., & Sauro, J. (2009). The factor structure of the system usability scale. In M. Kurosu (Ed.), *Human centered design* (pp. 94–103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lewis-Beck, M., Bryman, A. E., & Liao, T. F. (2003). *The sage encyclopedia of social science research methods*. Sage Publications.

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

Liu, J., Chang, W., Wu, Y., & Yang, Y. (2017). Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information* *retrieval, shinjuku, tokyo, japan, august 7-11, 2017* (pp. 115–124).

Livingston, G. R., Rosenberg, J. M., & Buchanan, B. G. (n.d.). Closing the loop: An agenda-and justification-based framework for selecting the next discovery task to perform. In *Data mining, 2001. ICDM 2001, proceedings of the 2001 IEEE international conference on data mining* (pp. 385–392). IEEE.

Lund, A. M. (2001). Measuring usability with the use questionnaire 12. *Usability interface*, *8*(2), 3–6.

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf..

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics (acl) system demonstrations* (pp. 55–60). Retrieved from http://www.aclweb.org/anthology/P/P14/P14-5010

Markatopoulou, F., Galanopoulos, D., Mezaris, V., & Patras, I. (2017). Query and keyframe representations for ad-hoc video search. In *Proc. of the 2017 acm on int. conf. on multimedia retrieval.* (pp. 407–411). ACM.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, *35*(5), 482–489.

McDonald, J. H. (2009). *Handbook of biological statistics* (Vol. 2).

Melis, G., Dyer, C., & Blunsom, P. (2017). On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*.

Mesbah, S., Bozzon, A., Lofi, C., & Houben, G.-J. (2018). Long-tail entity extraction with low-cost supervision.

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 26: 27th annual conference on neural information processing systems 2013. proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states.* (pp. 3111–3119).

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., & Fürnkranz, J. (2014). Large-scale multi-label text classification - revisiting neural networks. In *Machine learning and knowledge discovery in databases - european conference, ECML PKDD 2014, nancy, france, september 15-19, 2014. proceedings, part II* (pp. 437–452).

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2018). Information extraction from scientific articles: a survey. *Scientometrics*, *117*(3), 1931–1990.

Nishioka, C., & Scherp, A. (2015). Temporal patterns and periodicity of entity dynamics in the linked open data cloud. In *K-CAP* (pp. 22:1–22:4). ACM.

Ohsaka, N., Akiba, T., Yoshida, Y., & Kawarabayashi, K. (2016). Dynamic influence analysis in evolving networks. *PVLDB*, *9*(12), 1077–1088.

Passos, A., Kumar, V., & McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage..

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, october 25-29, 2014, doha, qatar, A meeting of sigdat, a special interest group of the ACL* (pp. 1532–1543). Retrieved from http://aclweb.org/anthology/D/D14/D14-1162.pdf

Pradhan, S., Moschitti, A., Xue, N., Ng, H. T., Björkelund, A., Uryupina, O., … Zhong, Z. (2013). Towards robust linguistic analysis using ontonotes. In *Conll.*

QasemiZadeh, B., & Schumann, A.-K. (2016). The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Lrec.*

Qiao, M., Zhang, H., & Cheng, H. (2017). Subgraph matching: on compression and computation. *PVLDB*, *11*(2), 176–188.

Ribeiro, N. F., & Yarnal, C. M. (2010). The perceived difficulty assessment questionnaire (pdaq): Methodology and applications for leisure educators and practitioners. *Schole*, *25*.

Robertson, S. (2016, July). A new interpretation of average precision. In *Proc. int. acm sigir conf. research and development in information retrieval* (pp. 689–690). Singapore, Singapore.

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

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

Sauro, J., & Lewis, J. R. (2012). *Quantifying the user experience: Practical statistics for user research* (1st ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

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* (pp. 101–117).

Schnabel, T., Labutov, I., Mimno, D. M., & Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing, EMNLP 2015, lisbon, portugal, september 17-21, 2015* (pp. 298–307).

Schwartz, A. S., & Hearst, M. A. (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific symposium on biocomputing* (p. 451-462).

Seok, M., Song, H.-J., Park, C.-Y., Kim, J.-D., & Kim, Y.-s. (2016). Named entity recognition using word embedding as a feature. *Int. J. Softw. Eng. Appl*, *10*(2), 93–104.

Shah, R. R., Yu, Y., Shaikh, A. D., & Zimmermann, R. (2015, Dec). Trace: Linguistic-based approach for automatic lecture video segmentation leveraging wikipedia texts. In *2015 ieee int. symp. on multimedia (ism)* (p. 217-220).

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, *15*(1), 1929–1958.

Strauss, B., Toma, B., Ritter, A., de Marneffe, M.-C., & Xu, W. (2016). Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd workshop on noisy user-generated text (wnut)* (pp. 138–144).

Sutton, C., & McCallum, A. (2006). *An introduction to conditional random fields for relational learning* (Vol. 2). Introduction to statistical relational learning. MIT Press.

Teplitskiy, M., Lu, G., & Duede, E. (2017). Amplifying the impact of open access: Wikipedia and the diffusion of science. *Journal of the Association for Information Science and Technology*, *68*(9), 2116–2127.

Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on natural language learning at hlt-naacl 2003-volume 4* (pp. 142–147).

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of the 2003 conf. of the north american ch. of the ass. for comp. ling. on human lang. tech. - volume 1* (pp. 173–180).

Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M. R., … Paliouras, G. (2015). An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, *16*, 138.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `http://dl.acm.org/citation.cfm?id=1858681.1858721`

Vagliano, I., Abdel-Qader, M., Blume, T., Böschen, F., Galke, L., Saleh, A., … Mutschke, P. (2018). *Deliverable 3.2: Technologies for MOVING data processing and visualisation v2.0* (Tech. Rep.). MOVING. Retrieved from `http://moving-project.eu/wp-content/uploads/2018/03/moving_d3.2_v1.0.pdf`

Vapnik, V. (1998). *Statistical learning theory*. New York: Willey.

Wang, L., Zhang, S., Shi, J., Jiao, L., Hassanzadeh, O., Zou, J., & Wangz, C. (2015, May). Schema management for document stores. *Proc. VLDB Endow.*, *8*(9), 922–933.

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

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., … Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., & Hovy, E. H. (2016). Hierarchical attention networks for document classification. In *NAACL HLT 2016, the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies, san diego california, usa, june 12-17, 2016* (pp. 1480–1489).

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). Recent trends in deep learning based natural language processing. *arXiv:1708.02709*.

Yuan, D., Mitra, P., Yu, H., & Giles, C. L. (2012). Iterative graph feature mining for graph indexing. In *ICDE.* IEEE Computer Society.

Yuan, D., Mitra, P., Yu, H., & Giles, C. L. (2015). Updating graph indices with a one-pass algorithm. In *SIGMOD.* ACM.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. In (Vol. 14, pp. 35–62). Elsevier.

Zhang, W., Wang, L., Yan, J., Wang, X., & Zha, H. (2017). Deep extreme multi-label learning. *arXiv preprint arXiv:1704.03718*.

Zhang, X., Zhao, J. J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems 28: Annual conference on neural information processing systems 2015, december 7-12, 2015, montreal, quebec, canada* (pp. 649–657).