Rheinische Friedrich-Wilhelms-Universität Bonn

Identifying and Improving Dataset References in Social Sciences Full Texts

Supervisors:

Prof. Dr. Sören Auer,

Dr. Philipp Mayr

A thesis submitted in partial fulfillment for the degree of Master of Science

by

Behnam Ghavimi

in the

Computer Science Department

May 2016

Declaration of Authorship

Rheinische Friedrich-Wilhelms-Universität Bonn

Abstract

Computer Science Department

Master of Science

by Behnam Ghavimi

Scientific full-text papers are usually stored separately from their underlying research datasets. Many quantitative social science papers include references to datasets, but in most cases the papers do not contain explicit links that would provide direct access to referenced datasets.

While registries that make datasets citable do exist, they are usually not integrated with authoring tools. Therefore, in practice, authors typically mention references to datasets by often using their titles and the year of publication, while explicit links are generally missing.

The manual identification of references to datasets in full-text papers is time consuming and requires hiring some experts in the paper's domain. In order to make all links to datasets in papers that have been published already explicit, we suggest and evaluate a semi-automatic approach for finding references to datasets in social science papers.

Our approach extracts all special features (i.e abbreviations and special phrases) from datasets' titles in da|ra registry. Afterwards, It detects datasets' references through using the extracted features. Finally, Our approach matches references with corresponding datasets' titles in da|ra through using a combination of tf-idf and cosine similarity. Users can assist the approach to improve its accuracy, for example, they can review the list of special features generated by the approach to remove or add items.

Our approach does not need a corpus of papers (no cold start problem) and it performs well on a small test corpus (gold standard). Our approach achieved an F-measure of 0.84 for detecting references in full texts and an F-measure of 0.83 for finding correct matches of detected references in the da|ra dataset registry.

Acknowledgements

I would like to express my deepest appreciation to Prof. Dr. Sören Auer for giving me the chance to complete my thesis in his group and under his priceless supervision.

I also cannot find words to express my sincere gratitude to Dr. Philipp Mayr for his time and patience. He always was there to support me whenever I encountered any trouble during the thesis research and writing.

I am indebted to Dr. Christoph Lange whose invaluable advices, kind supports and time that he has provided to me were extraordinary.

My thanks also go to Sahar Vahdati who has supported me with invaluable comments during the research and the writing process.

I like to thank my colleagues at GESIS – Leibniz Institute for the Social Sciences, especially Katarina Boland from the InFoLiS II project (MA 5334/1–2) for helpful discussions and for generating the gold standard for our evaluation.

Finally, I would also like to express significant thanks to my family, for their boundless love and support in all aspects of my life.

A summarized paper[1] of the thesis was also accepted in the ELPUB2016[1] conference and this accomplishment would not have been possible without the mentioned people.

This work has been funded partly by the DFG project "Opening Scholarly Communication in Social Sciences" (grant agreements SU 647/19–1 and AU 340/9–1).

[1] http://meetings.copernicus.org/elpub2016/

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Digital libraries have been growing enormously in recent years. They provide resources with high metadata quality, easy subject access, and support for retrieving information [2]. We are specifically interested in scientific full-text papers in digital libraries. Today, many papers in the quantitative social sciences make references to datasets. However, in most cases the papers do not provide explicit links that would give readers direct access to referenced datasets.

Explicit links from scientific publications to the underlying datasets and vice versa can be useful in multiple cases. For example, if a reviewer wants to check the evaluation mentioned in a paper, which was performed on a dataset, a link would give him or her straightforward access to the data and enable them to check the evaluation. Or, if other researchers want to perform further analysis on a dataset that was used in a paper, they would also be able to do so.

Today, the majority of papers do not have such direct links to datasets. While registries that make datasets citable do exist, e.g., by assigning a digital object identifier (DOI) to them, they are usually not integrated with authoring tools. Therefore, in practice, authors typically cite datasets by mentioning those using combinations of title, abbreviation and year of publication. (See e.g. Mathiak and Boland [3]).

Manually detecting references to datasets in papers is time consuming and requires an expert in the paper's domain. Although it is better to discuss each approach individually and avoid generalizing a problem, automatically detecting dataset references is challenging, since in most cases approaches need a huge corpus of papers or a training set.

It is difficult to create such a set, as there are wide varieties of styles and places for dataset citation in full texts. As illustrated in Figure 1.1, references to datasets can appear in different places in a paper.

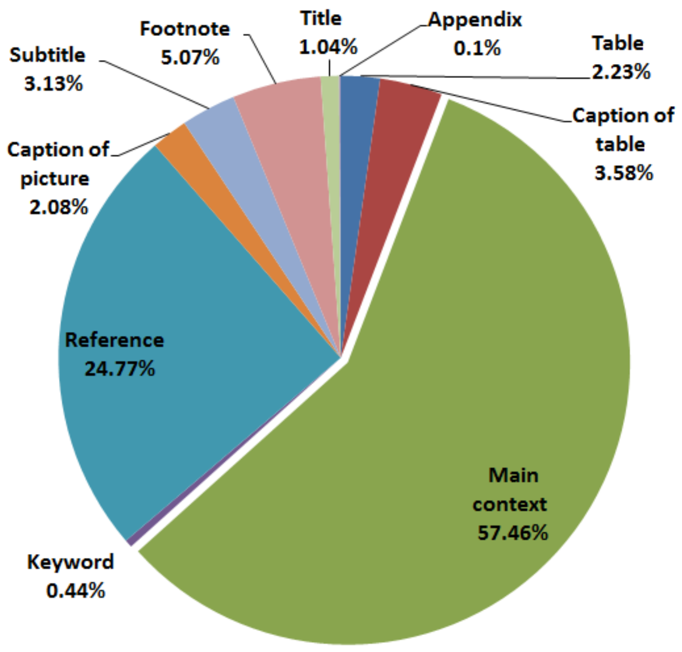


Figure 1.1: The distribution of dataset references in 15 random mda papers

This variance even makes rule-based approaches difficult, as it is hard to cover all cases. Possessing a huge corpus of papers also requires a large amount of space on hard disks, and processing the corpus is time consuming.

We therefore suggest a semi-automatic approach. The system parses full texts quickly and tries to find exact matches without possessing any training set. The approach only uses the context of the imported paper and the titles of datasets whose metadata are stored in a repository. The approach and the repository are explained in more detail in the following chapters.

1.2 Problem Statement

Whereas a lot of effort has been spent on information extraction in general [4], fewer attempts have focused on the specific task of dataset extraction (see e.g. [5]).

When referring to the same dataset, different authors often use different names or keywords. There are some standards for dataset citation in full texts. For example, Altman and King suggested a standard for dataset citation in [2007], but other authors still ignore or neglect such standards. Therefore, simple keyword or name extraction approaches do not solve the problem [7]. Table 1.1 shows five full-text papers that have cited different versions of a study (ALLBUS/GGSS (Allgemeine Bevölkerungsumfrage der Sozialwissenschaften/German General Social Survey)). In each paper, the reference style differs from the other papers.

	Citation style
Paper A	ALLBUS (2010)
Paper B	GESIS – Leibniz-Institute for the Social Sciences: ALLBUS 2010 – German General Social Survey. GESIS, Cologne, Germany, ZA4610 Data File version 1.0.0. (2011–05–30), doi:10.4232/1.10445.
Paper C	ALLBUS (Allgemeinen Bevölkerungsumfrage der Sozialwissenschaften)
Paper D	(e.g., in the German General Social Survey, ALLBUS; see Wasmer, Scholz, Blohm, Walter and Jutz, 2012)
Paper E	Die Einstellungen zu Geschlechterrollen wurden mit Hilfe von Items aus den ALLBUS – Wellen 1994 und 2008 operationalisiert.

Table 1.1: Citation styles for a study in five different papers.

Each detected dataset reference in a paper should be turned into an explicit link. This task can be done, for example, by using the DOI (Digital Object Identifier) of the dataset in a dataset registry. It means that each dataset reference in a paper should have a DOI code. These registries usually include some more metadata about the study, such as creators, publication date, description, and temporal coverage as well. In our case, these references to datasets should be linked to items in the da|ra registry, which is for social and economic data.

1.3 Thesis Contributions

This thesis makes the following contributions:

- A quantitative analysis of typical naming patterns used in the titles of social sciences datasets,
- A semi-automatic approach to finding references to datasets in social science papers with two alternative interactive disambiguation workflows, and
- An evaluation of the implementation of our approach on a corpus of journal articles.

1.4 Thesis Structure

Chapter 2 introduces the preliminaries of techniques and metrics that are applied in the thesis. Chapter 3 describes the prior existing literature related to this problem. All data used by the suggested approach is explained in Chapter 4, while Chapter 5 introduces the proposed solution. Implementation details of the approach are described in Chapter 6. The evaluation is then outlined in Chapter 7, and Chapter 8 concludes with an outlook to future work.

Chapter 2

Preliminaries

Our work and the related work of other researchers employ certain terminology and standard metrics for ranking the results of a search query (here: a text in a paper that refers to a dataset) over a corpus of documents (here: titles of datasets), and for evaluating the accuracy of information retrieval algorithms. The following four subsections introduce the terminology and the definitions of concepts used in this thesis.

2.1 Terminology

This subsection is about special words or expressions used in the thesis.

2.1.1 Dataset

"Dataset" is an ambiguous term since authors suggested varieties of meanings for it [8]. Therefore, Renear et al. introduced a general notion of the term based on the definitions in the technical and scientific literature. They define a dataset through using "grouping", "content", "relatedness", and "purpose" as the four basic characteristics of the dataset.

In their definition, the grouping feature considers a dataset as a group of data. "Set", "Collection", "aggregation", and "atomic unit" are some cases of this feature type. For instance, a dataset may be a "set" so it will not lose or accept any member (e.g. ”Set of RDf triples”)[9] or it may be a “Collection” so the deletion and addition of data don’t have any effect on the dataset identity. The content feature describes the data in a dataset. For example, “observation” describes the content is propositional while “value” is related to measurable content. The entities of a dataset are usually collected from activities such as measuring or observing.

A dataset is a group of data which are related to each other. The relatedness feature clarifies the relation of data in a dataset. “Syntactic”, and “semantic” are some examples of this feature. A group of data about a specific subject can be assumed to have a “semantic” relation. If all entities of a dataset have a specific structure, their relation is syntactic. Finally, purpose feature is about the idea of the scientific research which the dataset is created for.

2.1.2 Digital Repositories

Digital Repositories provide an infrastructure to manage digital materials. They may include different functions or appear in variant forms [10]. These Repositories contain not only digital data, but also their metadata. These repositories also facilitate reuse of stored materials for new research with an easy and persistent access to deposits [10].

2.2 Weighting Terms in Documents Using Tf-idf

The bag of words model represents a text as a set of terms of the text and does not consider the order of terms. Based on the model, documents and the query can be displayed in different ways, such as binary vectors, count vectors, and weight vectors. Each bit in a binary vector shows the absence or presence of a term in the related document of the vector.

In a count matrix, each row represents a term and each column represents the vector of a document or query. Each cell in the matrix shows the number of occurrences of a term in a related document or query.

Rows and columns are similar for weight and count matrices, but each cell in a weight matrix represents the weight of a term in a document. Tf-idf is one way of computing the weights.

Term frequency (tf) measures the number of occurrences of a given term (t) in a given document (d) or query text [11]. Weighing the score based on tf is calculated by the following formula.

$$w_{t,d}=\begin{cases} 1+log_{10}tf_{t,d}, & \text{if } tf_{t,d}>0 \\ 0, & \text{if } tf_{t,d}=0 \end{cases}$$

The reason for possessing a logarithm in the formula is that number of occurrences of a term does not make the document linearly more relevant. The total weight for a document is calculated by summing the weights of all terms in the document. These terms should appear in both q and d. It is zero if none of the query terms exist in the document.

$$\text{tf\_score}=\textstyle\sum_{t\in q\cap d} w_{t,d}$$

Df_t is the number of documents in the corpus that contain t, so as much as it repeats in the corpus, the term is less informative. This reason leads to a new measure, which is idf *(Inverse document frequency)*. Idf is effective for queries that have more than one term. The following formula is for idf, and N is the number of all documents in a corpus.

$$\mathit{idf} = \log_{10}(N/\mathit{df}_t)$$

Tf-idf is defined as the product of *tf* and *idf*. It increases through the repetition of a term in the document where the term is rare in the corpus.

$$\text{tf-idf (q,d)}=\textstyle\sum_{t\in q\cap d} \mathit{tf}.\mathit{idf}_{t,d}$$

When ranking documents that contain a term being searched, tf-idf returns high scores for documents for which the given term is *characteristic*, i.e. documents that have many occurrences of the term, while the term has a low occurrence rate in *all* documents of the corpus. In other words, the tf-idf algorithm assigns a weight to each word in a document, giving high weights to keywords and low weights to frequent words such as stop words.

2.3 The Cosine Similarity Metric

A Boolean search makes a user capable of finding a pattern, and if it is matched with document, the result will be one; otherwise the result will be zero. This means that documents either do or do not satisfy a query expression. But a ranked retrieval model returns a ranked list of documents in a corpus by considering a query.

Similarity measures such as Matching, Dice, Overlap Coefficient, and Jaccard are some example of the approaches for ranking a list of documents within a query (cf. Manning and Schütze [12]).

Matching Coefficient finds the numbers of terms that occur in both the query and document vectors. It calculates the cardinality of intersection of each document and query.

$$MatchingCoefficient = |d \cap q|$$

Dice Coefficient, Overlap Coefficient, and Jaccard try to normalize the Matching Coefficient, and their formulas are mentioned below.

- ☐ Dice Coefficient=$\frac{2|d \cap q|}{|d|+|q|}$
- ☐ Overlap Coefficient=$\frac{|d \cap q|}{min(|d|,|q|)}$
- ☐ Jaccard Coefficient=$\frac{|d \cap q|}{|d \cup q|}$

For example, Jaccard neither applies optimal normalization on the length of documents nor considers term frequency in a document and the corpus of documents. A document can be considered as a vector (point) in a vector space, each dimension of which corresponds to one term in the document corpus. A document can be converted into a weight vector, which looks like $d = (w_1, \ldots, w_n)$, and tf-idf is one way of computing the weight w_i of terms.

Search results for a multi-word query in a corpus of documents can be ranked by the similarity of each document with the query. Given a query vector q and a document vector d, their *cosine similarity* is defined as the cosine of the angle θ

between the two vectors [11, 12], i.e.

$$\cos(\overrightarrow{q}, \overrightarrow{d}) = \cos\theta = \frac{\overrightarrow{q} \cdot \overrightarrow{d}}{\|\overrightarrow{q}\| \, \|\overrightarrow{d}\|} = \frac{\overrightarrow{q}}{\|\overrightarrow{q}\|} \cdot \frac{\overrightarrow{d}}{\|\overrightarrow{d}\|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

It normalizes vectors by converting them to their unit vector, which makes documents of different lengths comparable (Figure 2.1). Since Euclidean distance is not effective for vectors of different lengths, it scores documents by considering angle instead of distance. Cosine decreases between 0 and 180 degrees monotonically, and therefore larger angles mean less similarity.

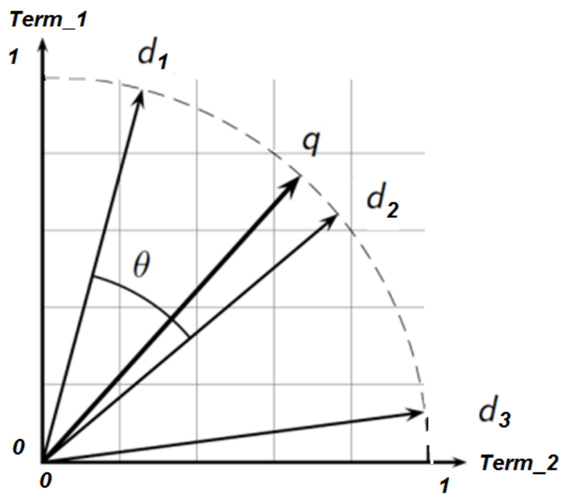


FIGURE 2.1: Length normalization by cosine similarity

Combining tf-idf and cosine similarity yields a ranked list of documents. In practice, it may furthermore be necessary to define a cut-off threshold in order to distinguish documents that are considered to match the query from those that do not [13].

2.4 Precision and Recall of a Classifier

We aim at implementing a binary classifier that tells us whether or not a certain dataset has been referenced by a paper. The algorithm tries to find references of datasets in a paper, and then as the next step, it attempts to detect a perfect match for each detected reference in a text. These matches are to be selected from titles of datasets in the da|ra repository.

Evaluation metrics such as *precision and recall* determine the reliability of binary classifiers, and F-measure is a harmonic mean of precision and recall. These three metrics are defined as follows [14].

- ☐ Precision=$\frac{\#\text{True positives}}{\#\text{True positives}+\#\text{False positives}}$
- ☐ Recall=$\frac{\#\text{True positives}}{\#\text{True positives}+\#\text{False negatives}}$
- ☐ F-measure=$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

If an algorithm returns less wrong predictions, it will lead to high precision. The algorithm should predict most of the relevant results to achieve high recall.

The calculation of these three metrics requires ground truth, which is illustrated in Figure 2.2. Lamiroy and Sun suggested a probabilistic model should be used to estimate precision and recall since it is usually too difficult – and in some cases even impossible – to find out [15].

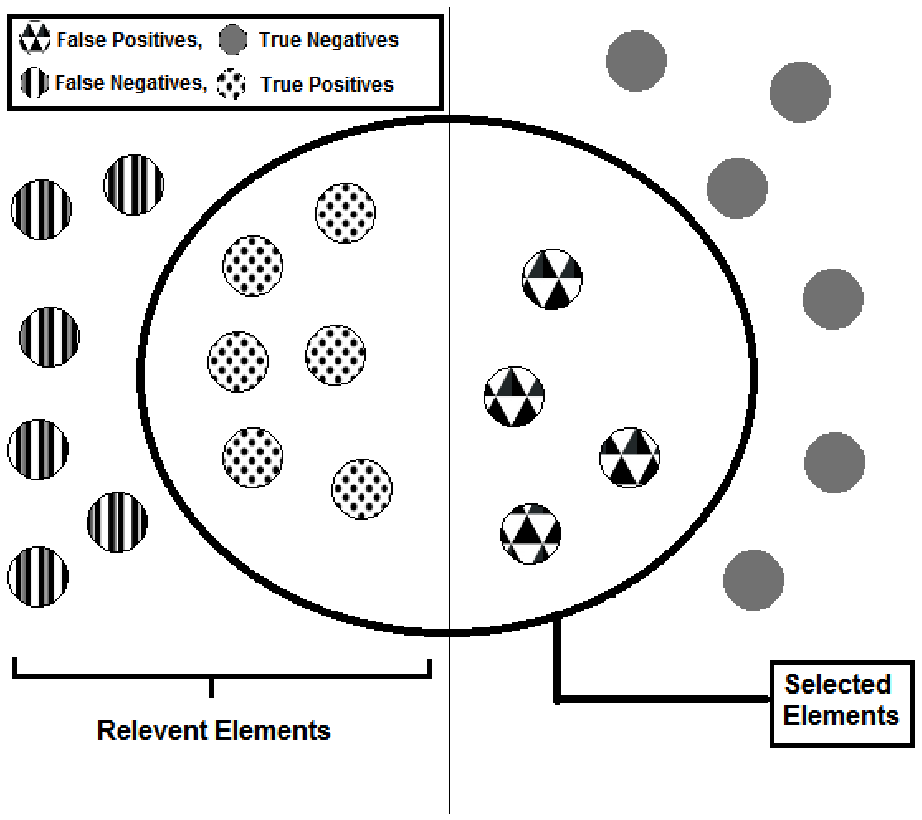


FIGURE 2.2: A predictor on a distribution of data (True Positives, True Negatives, False Positives, and False Negatives)

Chapter 3

Related Work

While only a few scientific works have been found on the specific task of extracting dataset references from scientific publications, a lot of research has been done on its general foundations, including metadata extraction and string similarity algorithms.

Solutions for extracting keywords from papers are useful for us, as references to datasets can be considered a special case of keywords. Related works can be categorized in different ways. We decided to divide them into the categories covered by the following sub-chapters.

3.1 Methods Based on the Bag of Words Model

As a short recall to a previous chapter, a text can be considered as a set of words and represented as a vector by, for example, indicating the absence or presence of a word in the text. In other words, we can assume a vector space, each of whose dimensions corresponds to one word. Weights for terms in such vectors need to be adjusted by weighting algorithms such as tf-idf. Lee and Kim proposed an unsupervised keyword extraction method by using a tf-idf model with some heuristics [2008].

Our approach uses similarity measures for identifying a perfect match for each dataset reference in a paper by comparing dataset titles in a repository to sentences in papers. Similarity measures such as Matching, Dice 2, Jaccard, and Cosine can be easily applied to a vector representation of a text (cf. Manning and Schütze [12]).

The accuracy of algorithms based on such similarity measures can be improved by making them semantics-aware, which can be done by representing a set of synonyms as a single vector space dimension.

3.2 Corpus and Web Based Methods

Methods in this category often use information about the co-occurrence of two texts in documents, and are used for measuring texts' semantic similarity. Turney introduced a simple unsupervised learning algorithm for detecting synonyms in 2001, which searches queries through an online search engine and analyzes the results. The quality of the algorithm depends on the number of search results returned.

Singhal and Srivastava proposed an approach to extract dataset names from articles [2013]. They employed the NGD algorithm, which estimates both the probability of two terms existing separately in a document, as well as of their co-occurrence.

$$NGD(x,y) = \frac{\max(\log f(x), f(y)) - \log f(x,y)}{\log M - \min(\log f(x), \log f(y))}$$

In the formula, M is the number of all web pages searched. F(x) means the number of returned pages for x as a query term and f(x,y) represents the number of pages for the intersection of x and y. They used two research engines – Google Scholar and Microsoft Academic Search – instead of a local corpus.

Schaefer et al. proposed the Normalized Relevance Distance (NRD) [2014]. This metric measures the semantic relatedness of terms. NRD is based on the co-occurrence of terms in documents, and extends NGD by using relevance weights of terms. The quality of these methods depends on the size of the corpus used.

Sahami and Heilman suggest a similarity function based on query expansion [2006]. Their algorithm determines the degree of semantic similarity between two phrases. Each of these phrases is searched by an online search engine and then expanded by using returned documents. Afterwards, the new phrases are used for computing similarity.

The problem that we aim to solve is detecting dataset references in a paper and then finding at least one correct match for each of these identified references. The task can be split into two subtasks: the identification and matching of dataset references in a paper. Literature citation mining is the process of determining the number of citations that a specific paper receives. It constructs a literature citation network, which can be used for detecting the quality of a paper [21]. Citation mining can usually be handled by three subtasks. First, literature references should be extracted from the bibliography section of a document, and afterward, metadata extraction should be applied on the extracted references from the first phase. Finally, each reference should be linked to the cited paper by using extracted metadata from the second step [21].

Dataset and literature citation mining from documents are not capable of being compared in the detecting phase, since dataset mining needs to be applied to the entire paper, but literature mining must only consider the bibliography of the paper. Unlike the detection phase, they can mostly use the same strategy for the matching phase, but of course are not completely the same.

Afzal et al. proposed a rule-based citation mining technique [21]. Their approach detects literature references from each document and then extracts citation metadata from each of them, such as title, authors, and venue. Based on the venue, it then extracts all related titles from DBLP, which is a computer science bibliography and contains more than three million papers. Finally, it tries to link the title of each extracted literature reference and those found in DBLP. Our approach tries to match data references in a paper to the titles of registered datasets in the da|ra repository.

3.3 Machine Learning Methods

Many different machine-learning approaches have been employed for extracting metadata, and in a few cases also for detecting dataset references. For example, Zhang et al. [2006] and Han et al. [2003] proposed keyword extraction methods based on support vector machines (SVM).

Kaur and Gupta conducted a survey on several effective keyword extraction techniques, such as selection based on informative features, position weight, and conditional random field (CRF) algorithms [2010]. Keyword extraction from a paper can be considered as a labeling task. CRF classifiers can assign labels to sequences of input, and, for instance, define which parts in a paper can be assumed to be keywords [25].

Cui and Chen proposed an approach using Hidden Markov Model (HMM) to extract metadata from texts [2010]. HMM is a language-independent and trainable algorithm [27]. Marinai described a method for extracting metadata from documents by using a neural classifier [2009]. Kern et al. proposed an algorithm that uses a maximum entropy classifier for extracting metadata from scientific papers [29]. Lu et al. used the feature-based Llama classifier for detecting dataset references in documents [2012]. Since there are many different styles of datasets' references, large training sets are necessary for these approaches.

Boland et al. proposed a pattern induction method for extracting dataset references from documents in order to overcome the necessity of such a large training set [2012]. Their algorithm starts with either the name of a dataset or with an abbreviation of this name, and then drives patterns of all phrases that contain that name or abbreviation in papers. The patterns are applied to papers in order to extract more dataset names and abbreviations. This process repeats with new abbreviations and names until no more datasets can be detected in papers. It derives patterns of phrases that contain dataset references iteratively by using a bootstrapping approach.

Chapter 4

Data Sources

This section describes the three types of data sources that we use. We use full-text articles from the journal mda to evaluate the performance of our dataset linking approach, and metadata of datasets in the da|ra dataset registry to identify datasets. Finally, we use metadata of the registered papers in the SSOAR[1] repository for exporting the suggestions of our approach for a paper as a JSON file.

4.1 Papers from mda Journal

Methods, data, analyses (mda[2]) is an open-access journal that publishes studies on all questions that are important to quantitative methods. The journal focuses particularly on survey methodology. It publishes research on different aspects of methodical study of surveys, such as data collection, measure, and data analysis. All of mda's content is freely available online, and can be distributed without any constraint to guarantee the free flow of information, which is important for scientific progress. We use a random sample of full-text articles from mda as our test corpus.

[1]http://www.ssoar.info
[2]http://www.gesis.org/en/publications/journals/mda/

4.2 The da|ra Dataset Registry

4.2.1 da|ra Overview

Our proposed approach aims at social science datasets since it uses registered datasets in da|ra registry[3], and the registry offers the DOI registration service for social science and economic data. In addition, the selected papers used as evaluation data were from mda, which focuses mostly on survey methodology in social science research areas.

Different institutions have collected research data in the social sciences and made them available. Although the accessibility of such datasets for further analyses and reusing is important, information about where to find and how to access them is often missing in papers.

da|ra makes social science research data referable and thus improves its availability when needed. This is achieved by assigning a digital object identifier (DOI) – a unique string that identifies an item (here: a dataset), and supplies a link to its place on the web – to each dataset. da|ra therefore does a job similar to that of publishers assigning DOIs to articles when they are published electronically.

At the present time, da|ra holds 432,312 records such as datasets, texts, collections, videos, and interactive resources, 32,858 of which are datasets. For each dataset, da|ra provides metadata including title, author, language, and publisher. This metadata is exposed to harvesters employing a freely accessible API using OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) [4].

This API supplies a set of services for displaying and harvesting repositories' metadata. The set enables a user to work with a repository by, for example, retrieving an individual record or full records and the structure of the repository. It also lets the user implement a selective harvester by manipulating some parameters such as "*identifier*", which specifies a record identifier and "*resumptionToken* ", which is a token for resuming a request where it last stopped.

[3]http://www.da-ra.de
[4]http://da-ra.de/oaip/

4.2.2 Analysis of Dataset Titles in da|ra

We analyzed the titles of all datasets in da|ra and the titles were harvested by using the API of da|ra. The analysis shows that about one third of the titles follow a special pattern, which makes them easier to be detected in the text of a paper. We have identified three such special patterns. First, there are titles that contain *abbreviations*, which are often used to refer to the datasets.

Consider, for example, the full title "Programme for the International Assessment of Adult Competencies (PIAAC), Cyprus ", which contains the abbreviation "PIAAC ". Secondly, there are *filenames*, as in the example "Southern Education and Racial Discrimination, 1880–1910: Virginia: VIRGPT2.DAT", where "VIRGPT2.DAT" is the name of the dataset file. Finally, there are *phrases* that explicitly denote the existence of datasets in a text, such as "Exit Poll" or "Probation Survey". "Czech Exit Poll 1996" is an example of such a dataset title.

We assume these three categories as special characteristics in the titles. Abbreviations and special phrases can be found in about 17 and 19 percent of the da|ra dataset titles respectively. The intersection of these two groups is only 1.49 percent. Filenames occur in less than one percent of the titles. This distribution is illustrated in Figure 4.1. The proposed approach in this thesis uses only the first and the last categories, since the second one, which is the filename category, only covers a small amount of titles.

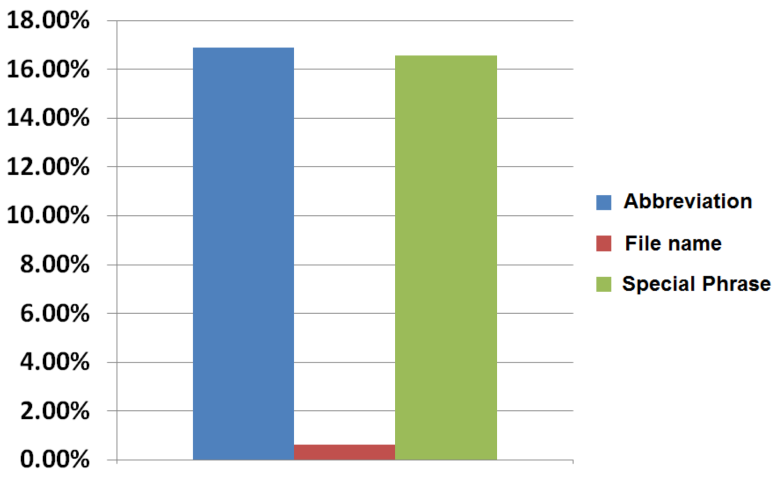


FIGURE 4.1: Distribution of titles in three categories

4.3 The Social Science Open Access Repository (SSOAR)

The repository provides full-text social science available documents without any charge and it is another product of GESIS Institute. It covers scholarly contributions related to different social science fields such as social psychology, communication sciences, and the historical social research. These full-texts are available in German or English language.

This repository provides some metadata such as abstract and keywords for each paper in both German and English language. It is assumed as a secondary publisher which publishes pre-prints, post-prints, and original publishers' versions of scholarly papers but it also let authors to publish their work for the first time.

This repository improve rankings of its scholarly papers in search engine queries so it facilitates discovering of the papers. Furthermore, the metadata of the papers inside the repository are able to be harvested easily.

It assigns a URN (Uniform Resource Name) as a persistent identifier (PID) to each full-text to establish a stable link to the paper and if the full-text is the pre-print or post-print version of a published work, the repository uses the Digital Object Identifier (DOI) of the paper.

Chapter 5

A Semi-Automatic Approach for Finding Dataset References

We have created a semi-automatic approach for finding references to datasets registered in da|ra in a given full text. It is possible to divide our approach into four main steps. The first step is related to generating special features dictionaries from datasets' titles. The second step deals with identifying and matching datasets' references in a paper, and the third step focuses on improving the results of the second step. Finally, a user exports the results in the fourth and final step.

It took a semi-automatic approach since the first and last steps of our algorithm require human interaction to improve the accuracy of the result. In the first step, the user should review two generated lists of abbreviations and special phrases. In the final step, the user should make the final decision regarding references suggested by our approach.

The main differences between our approach and the other related works are that ours do not need a huge corpus of papers or a large training set. Our approach is quite fast and is able to prepare results for a paper in few minutes or even seconds depending on the number of datasets' references in the paper that we want to analyze.

5.1 Step 1: Preparing the Dictionary

The preparation of a *dictionary* of abbreviations and special phrases is the first step. *Abbreviations* are initially obtained by applying some algorithms and rules to the dataset titles harvested from da|ra.

The titles are preprocessed automatically before the abbreviations are extracted. Titles fully in capital letters are removed, the remaining titles are split based on ":", and then only the first parts are kept (in the case of including any colon mark).

The extraction of abbreviations from titles follows specific steps:

1. The titles are tokenized.
2. The tokens that are not completely in lowercase (not including the first letter) – not only a combination of digits and punctuation marks, not Roman numerals, and do not start with a digit are added to a new list (e.g. "SFB580-B2", "A*CENSUS", "L.A.FANS", "aDvANCE" and "GBF/DIME").
3. The titles are split based on dash and left parenthesis, and then the first parts with the length of a token are added to the list (e.g "euandi" in "euandi (Experteninterviews) - Reduzierte Version").
4. The items on the list of abbreviations should only contain dot, dash, slash, star and "&" from punctuation marks (e.g. "NHM&E").
5. The items that contain slashes or dashes and are also partially in lowercase are removed from the list (first letter of each part is not included) (e.g "Allbus/GGSS" is removed).
6. Words in German and English, as well as country names, are removed from the list. Words, fully or partially in capital letters will not be pruned by dictionary (first letter is not included) (e.g. "ALLBUS").

The titles fully in capital letters are converted into lowercase and tokenized. Afterwards, the dictionary prunes them, and then their tokens without definition are added to the list.

These algorithms and rules correctly detect, for example, "DAWN" in "Drug Abuse Warning Network (DAWN), 2008". However, it sometimes detects abbreviations that are not references to datasets, such as “NYPD” in “New York Police Department (NYPD) Stop, Question, and Frisk Database, 2006”. As their identification is hard to automate, we assigned this task to a human expert. The expert reviews the list and then makes a false positive list – such false positives will be removed from the dictionary automatically. The ratio of false positives is one out every three items on the list of abbreviations extracted automatically. This means that approximately 66 percent of the abbreviations are derived correctly, and the rest of the titles need very little effort in order to be pruned from the list.

The preparation of the dictionary of special phrases also needs human interaction. A list of terms that refer to datasets such as “Study” or “Survey” has been generated manually; this list contains about 30 items. Afterwards, phrases containing these terms were derived by some algorithms and rules from the titles of actual datasets in da|ra. Three types of phrases are considered here, the first of which are tokens that include an item in the dictionary such as “Singularisierungsstudie” where the phrase contains “studie”.

The second is a category of phrases that includes “Survey of” or “Study of” as a sub phrase as well as one more token that is not a stop word, such as “Survey of Hunting”. The last one is phrases that contain two tokens, where one of them is an item in the dictionary such as “Poll”, and the second token should not be a stop word such as “Freedom Poll”.

A human expert has finally verified the phrase list, and false positives are added to the related list. In the phrase list, there are few false positives, and most can be detected while processing papers. This means our approach will improve over time. Both dictionaries – abbreviations and phrases – can be generated on the first harvest of dataset titles from da|ra, and they can be required to update with every subsequent harvest. The delta update feature is not implemented in the thesis, but the idea makes our approach even faster.

5.2 Step 2: Detecting Dataset References and Ranking Matching Datasets

Next, the characteristic features (abbreviations or phrases) of dataset titles are detected in the full text of a given paper. A paper is split into sentences, and each of these features is searched for in each sentence. Any detection of the special features in a text means a dataset reference in the text exists.

A sentence is split into smaller pieces if a feature repeats inside the sentence more than once, since such a sentence may contain references to different versions of a dataset. Any phrase identified in this step might correspond to more than one dataset title.

For example, "ALLBUS"[1] is an abbreviation for a famous social science dataset, of which more than 150 versions are registered in da|ra. These versions have different titles and, for instance, the titles differ from year of study such as "German General Social Survey – ALLBUS 1998", "German General Social Survey – ALLBUS 2010", and "German General Social Survey (ALLBUS) – Cumulation 1980–2012".

In another example, two titles that both contain the "PIAAC" abbreviation are "Programme for the International Assessment of Adult Competencies (PIAAC), Cyprus" and "Programme for the International Assessment of Adult Competencies (PIAAC), Germany", i.e., two datasets that differ in their geographic coverage. The last example is the two versions of "EVS" dataset, "EVS – European Values Study 1999 – Italy" and "European Values Study 2008: Azerbaijan (EVS 2008) ", which differ in both their year of study and geographic coverage.

We solve the problem of identifying the most likely datasets referenced by the text in the paper by ranking their titles with a combination of tf-idf and cosine similarity. In this ranking algorithm, we apply the definitions of Chapter 2, where the query is a candidate dataset reference found in the paper and the documents are the titles of all datasets in da|ra. It means that our approach tries to identify the most similar dataset title in the da|ra repository with a sentence that contains any of the special features where the sentence belongs to the analyzed paper.

5.3 Step 3: Heuristics to Improve Ranking in Step 2

For each reference detected in the full text of a paper, the approach as presented so far computes tf-idf over the full text of the paper and over the list of the titles

[1]Allgemeine Bevölkerungsumfrage der Sozialwissenschaften = German General Social Survey

of datasets in da|ra, which contain a specific characteristic feature (abbreviation or phrase) detected in the reference. As it leads to many false positives based on our observation, comparing datasets' titles with a sentence in a paper, and, afterwards, ranking titles based on their score was not useful. Therefore we solved the problems by involving special features.

Our approach considers only the list of titles that contain the special feature detected in the reference, since they are related titles and the rest of the titles in the repository are irrelevant. We limit our options in order to improve the accuracy of our approach. We decided to use the list of titles and whole sentences of the paper, and not only the reference sentence, since this consideration enables us to have a bigger corpus of documents and to reach a better weight for each word. The utilization of titles that contain the special features reduces the weight score of the feature and raises the weight scores of other terms in the reference sentence. It therefore has a positive impact on accuracy.

While a corpus of papers is typically huge, the size of all da|ra dataset titles and the size of the full text of an average paper are less than 4 MB each. Given this limited corpus size, our algorithm may detect some false keywords in a query, thus adversely affecting the result. For instance, Figure 5.1 illustrates a model example of this problem.

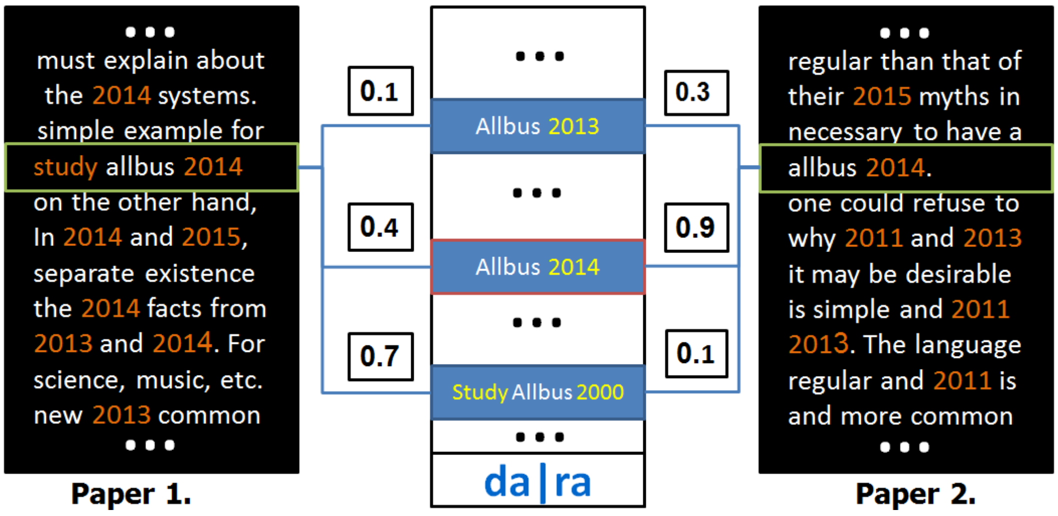


FIGURE 5.1: A model example of cosine similarity, where tf-idf is computed over phrases in two papers.(The numbers are not from a real example)

In paper 1, "2014" repeats many times, whereas the word "study" occurs only once, which means the tf-idf assigns a high weight to "study" and a low weight to “2014”. When the query string is “study allbus 2014”, cosine similarity gives a higher rank to “Study Allbus 2000” than “Allbus 2014”.

To address this problem in a better way, our implementation employs some heuristics. This includes an algorithm that improves dataset rankings based on matching years in the candidate strings in both the paper and the datasets’ titles. In the example, these heuristics improve the ranking of the “Allbus 2014” dataset when analyzing paper one. Figure 5.2 shows an overview of our approach. Two steps labeled with “M” means they need human interactions. “M1” is about the preparation of lists of special features and “M2” is about making final decisions between candidates suggested by our approach.

5.4 Step 4: Exposing the Results to the User, and Interactive Disambiguation

Our application supports two workflows through which an expert user can choose the best matches for the datasets cited by a paper from a set of candidates identified automatically. The sizes of these sets have been chosen according to the observations we made during the evaluation of the automated step, as explained in Chapter 7.

One workflow works per reference: for each reference, five titles of candidate datasets are suggested to the user. While this workflow best supports the user in getting every reference right, it can be tiring; each paper in our corpus contains 45 dataset references on average, but these *references* only belong to an average number of three distinct *datasets*.

The second alternative workflow takes advantage of this observation. It works per characteristic feature and suggests six titles of candidate datasets to the user for each feature (which may be common to multiple individual references in the paper).

Finally, an RDF Graph will be exported as an output, which contains information about links identified between papers and datasets. To enable even further analysis of the links identified between papers and datasets, we export an RDF graph containing all candidate datasets identified in the latter workflow for each paper.

For each candidate dataset, we represent the essential metadata of the dataset in RDF: DOI and title.

RDF (Resource Description Framework) is a standard for storing and exchanging data. It encodes structured information through using graphs. Different serialization formats are used for translating data structures into a format that can be stored such as RDF/XML, JSON-LD, and RDFa. JSON format provides a compressed structure for storing JavaScript data. JSON-LD is an improved JSON which includes an RDF context and many programming languages support JSON.

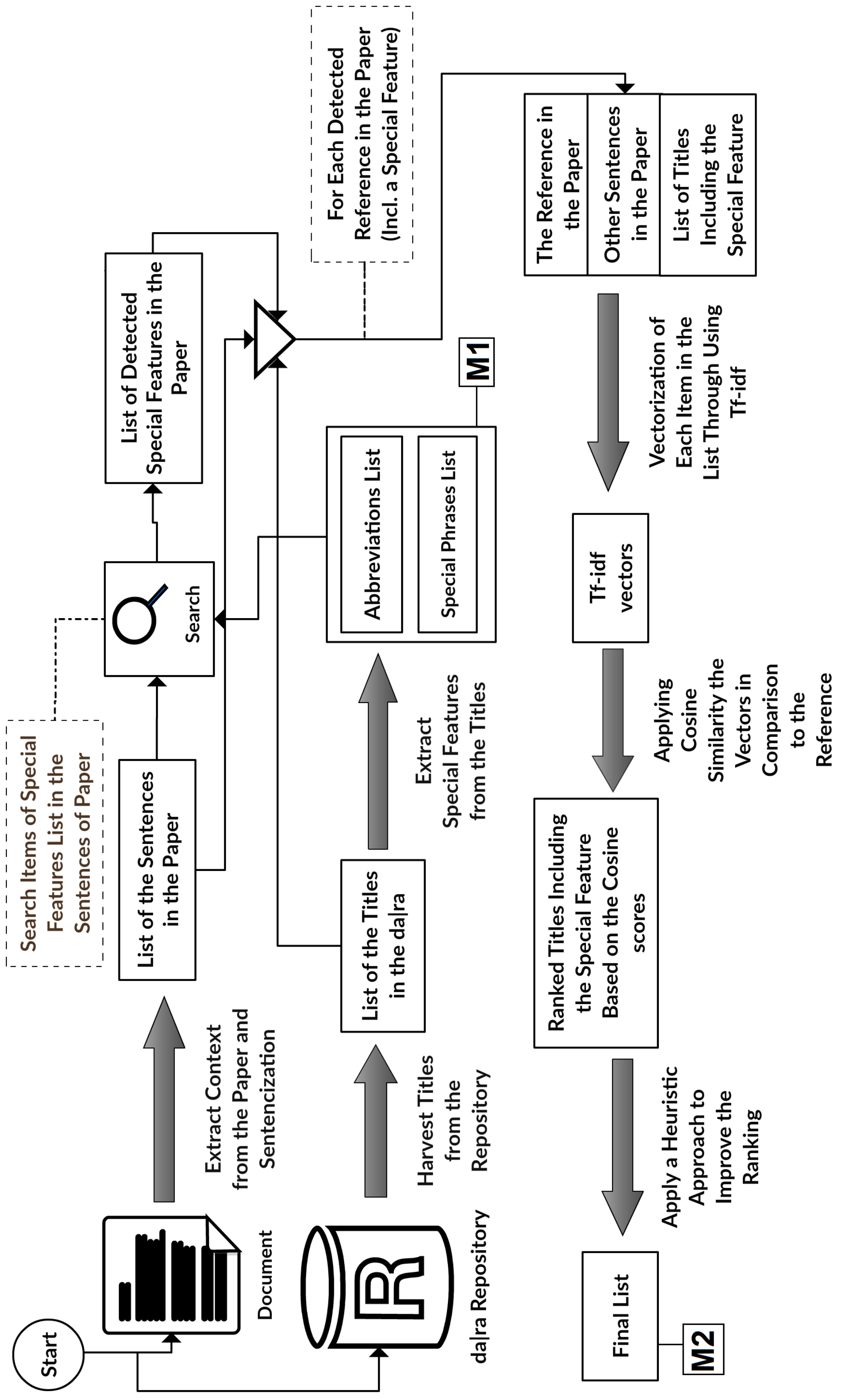

Start
Document
Extract Context from the Paper and Sentencization
List of the Sentences in the Paper
Search Items of Special Features List in the Sentences of Paper
Search
List of Detected Special Features in the Paper
da|ra Repository
Harvest Titles from the Repository
List of the Titles in the da|ra
Extract Special Features from the Titles
Abbreviations List
Special Phrases List
M1
For Each Detected Reference in the Paper (Incl. a Special Feature)
The Reference in the Paper
Other Sentences in the Paper
List of Titles Including the Special Feature
Vectorization of Each Item in the List Through Using Tf-idf
Tf-idf vectors
Applying Cosine Similarity the Vectors in Comparison to the Reference
Ranked Titles Including the Special Feature Based on the Cosine scores
Apply a Heuristic Approach to Improve the Ranking
Final List
M2

Figure 5.2: An overview of the approach.

Chapter 6

Implementation

6.1 Structural and Behavioral Diagrams

One of the best ways to clarify details about an application is utilizing UML diagrams. These diagrams help users and other developers to realize the application from both structural and behavioral aspects [31]. The structural diagrams are responsible for showing the static aspects of the system, such as classes and packages, while behavioral diagrams consider the dynamic aspect of a system. Some instances of behavioral diagrams are use case and activity diagrams [32]. We drew different diagrams to thoroughly cover all aspects.

6.1.1 Use Case Diagram

A set of available services from system services performed by actors are assumed as use cases of the system. A use case diagram demonstrates the relationships among actors, and use cases of a system where an actor is a user who interacts with the functionalities of the system [31]. It is possible that a group of use cases satisfies a goal.

For example, a user should import a document and then apply our approach to the document to detect dataset references in the document. This means we need to follow at least two use cases in order to detect references in a paper.

Actors are usually depicted by “stick man” symbol and are labelled by the name of the actor. An association relationship shows an interaction between an actor and a use case, where the actor is able to link to even more than just a use case. Extending a use case is often dependent and optional. A dashed line displays the relationship with a direction from the extending use case to the extended one. Where this link is used for simplifying the use case, an include relationship is employed to divide a use case into more than one case. Figure 6.1 demonstrates the use case diagram of our system.

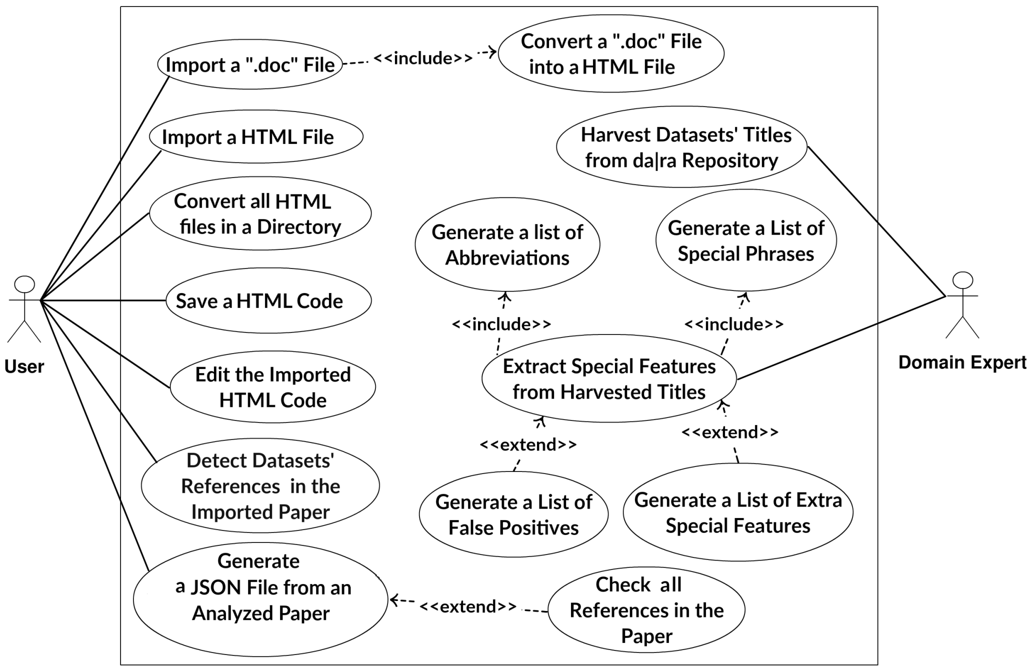

Import a ".doc" File
<<include>>
Convert a ".doc" File into a HTML File
Import a HTML File
Harvest Datasets' Titles from da|ra Repository
Convert all HTML files in a Directory
Generate a list of Abbreviations
Generate a List of Special Phrases
Save a HTML Code
<<include>>
<<include>>
User
Domain Expert
Edit the Imported HTML Code
Extract Special Features from Harvested Titles
<<extend>>
<<extend>>
Detect Datasets' References in the Imported Paper
Generate a List of False Positives
Generate a List of Extra Special Features
Generate a JSON File from an Analyzed Paper
<<extend>>
Check all References in the Paper

FIGURE 6.1: The use case diagram of our application.

Table 6.1 provides some information about our application use cases. The information explains the task complexity and the execution time of each use case. A task complexity is a description about sub tasks of a use case and the number of clicks required to complete the use case. These times in the table are obtained through using Intel i5–5200U (CPU), 4 GB (RAM), and Windows 7 (OS) in the client side.

Use Case	**Description of sub tasks**	**Execution Time**
Import a HTML file	(1). Execute the related function (by two clicks). (2). Select a HTML file (min. by two clicks).	Approx. 4 seconds for a paper with 35 pages (The time depends on the user interaction).

Use Case	**Description of sub tasks**	**Execution Time**
Import a Microsoft Word file	(1). Execute the related function (by two clicks). (2). Select a ".doc" file (min. by two clicks).	Approx. 10 seconds for a paper with 35 pages (The time depends on the user interaction).
Convert all HTML files in a directory	(1). Execute the related function (by two clicks). (2). Select a directory (min. by two clicks).	Approx. 128 seconds for a directory with 24 papers (On the average, each paper includes 25 pages).
Save an HTML document	(1). Execute the related function (by two clicks).	Approx. 4 seconds for a paper with 35 pages. (The time depends on the user interaction).
Harvest datasets' titles from da|ra repository	(1). Execute the related function (by two clicks). Note: The task generates two lists of special features (abbreviations and special phrases) from the harvested titles.	Approx. 7 hours and 15 minutes (The time depends on the API of da|ra and the hardware configuration of the client system).
Extract special features from the harvested titles	(1). Update the list of false positives. (Optional) (2). Update the list of extra true special features. (Optional) (3). Execute the related function (by two clicks).	Approx. 118 seconds (Without concerning optional sub tasks).
Detect datasets' references in the imported paper	(1). Execute the related function (by two clicks). Note: Firstly, lists of special features should be generated and a paper should be imported to be analyzed.	Approx. 40 seconds for a paper with 45 datasets' references on the average.

Use Case	**Description of sub tasks**	**Execution Time**
Edit the HTML document	(1). Call its related form (by two clicks). (2). Edit the HTML document of the imported paper. (3). Decide to apply or discard the changes (by one click).	The time depends on the user interaction.
Check each detected dataset reference in a paper	(1). Call the related form (by two clicks). (2). Select an item among detected special features in the imported paper (by one click). (3). Check each title related to the selected feature(min. by one click). Note: Firstly, a paper should be imported.	The time depends on the user interaction.
Generate a JSON file for a paper	(1). Call the related form (by two clicks). (2). Select an item among analyzed papers (by one click). (3). Select correct items on the list of suggested datasets' titles for the paper. (4). Export a JSON File of selected items (by one click). Note: Firstly, a paper should be imported and analyzed by our approach.	It depends on the user interaction.

TABLE 6.1: Our application use cases with their complexities and execution times.

6.1.2 Activity Diagram

An activity diagram displays the sequence of employing different functionalities of a system, and it is often necessary to have more than one diagram to cover a system completely. An activity depicts a behavior that is formed by some individual actions. Actions are displayed by rounded rectangles in the diagram with a label inside [31].

This action edge is illustrated by a solid line, which possesses a direction from the source into the target action. A start node depicts when the activity is initialized, and a filled black circle symbolizes the node [32]. An end node defines where the activity flow is terminated, and the node is depicted by two nested circles with the inner circle colored black.

A diamond symbolizes a condition element, and it is responsible for illustrating a condition in a system's activity flow [31]. A fork is an item in the diagram that handles parallel activities and is depicted by a bar, which has one edge as an input and more than one output. In contrast to fork, a join node is an element in the diagram used to synchronize some different activities. The item in an activity diagram has multiple input activity edges and one single output edge. A bar also depicts the element, and it is able to merge with the fork node [32]. A note element

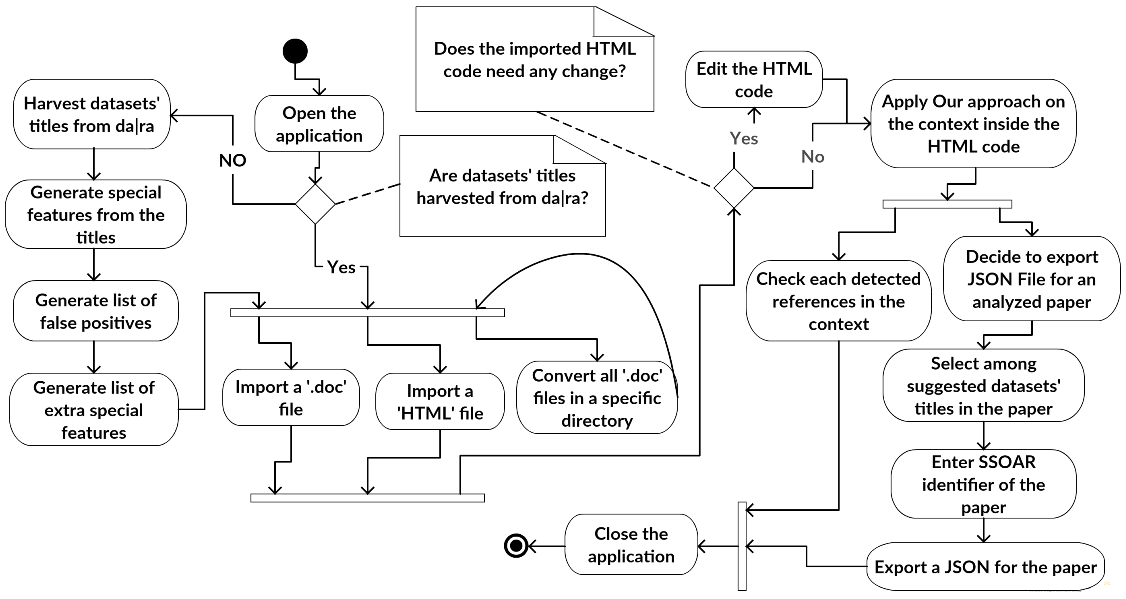

Does the imported HTML code need any change?
Edit the HTML code
Harvest datasets' titles from da|ra
Open the application
Apply Our approach on the context inside the HTML code
NO
Yes
No
Are datasets' titles harvested from da|ra?
Generate special features from the titles
Yes
Check each detected references in the context
Decide to export JSON File for an analyzed paper
Generate list of false positives
Select among suggested datasets' titles in the paper
Generate list of extra special features
Import a '.doc' file
Import a 'HTML' file
Convert all '.doc' files in a specific directory
Enter SSOAR identifier of the paper
Close the application
Export a JSON for the paper

FIGURE 6.2: The activity diagram of our application.

is symbolized by a rectangle whose top right corner is bended inwards. It contains further information about another element linked to the note. Figure 6.2 displays the activity diagram of our application. We drew the diagram to cover our system entirely, but some details are still missing, since, as we mentioned, it is hard or sometimes impossible to explain the flow of activities in a system by just employing an activity diagram. Since we have implemented a Windows application it is possible to move freely among actions or to repeat an action.

6.1.3 Package Diagram

A package diagram demonstrates the architecture of our application and how it is designed in different system packages and their dependencies. A package assigns a namespace to group related objects, and the objects usually support the handling of a specific service or a group of similar functionalities. This diagram introduces some new relationships between packages to the standard UML relationships. The "import" is one of the relationships and it shows a direction from an importing package into the imported package. A dashed line without any label symbolizes the relationship.

We have implemented modules of our application in five categories. One of the packages is responsible for handling ".doc" and HTML files. It provides functionalities for importing files into our application, as well as for parsing and editing HTML files.

Another package enables users to harvest datasets' titles from the da|ra repository and afterwards, it handles extracting special features from the titles. The next package supports our approach to extract datasets' references in a paper. It is responsible for applying the mixture of the cosine similarity and tf-idf to both datasets' titles in the da|ra repository and sentences in the paper as well.

There is another package for visualizing the results of our approach. The last one is the main package, which includes a module that binds all other packages together and uses its interface to illustrate the functionalities. Figure 6.3 illustrates the package diagram of our application.

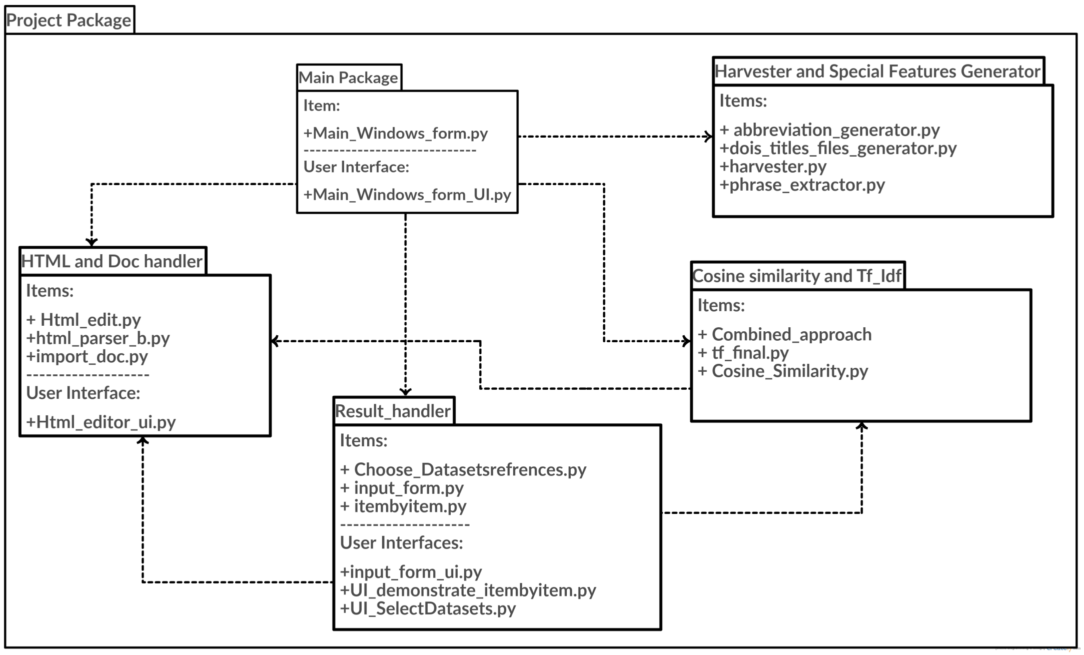


FIGURE 6.3: The package diagram of our application.

6.2 Back-End

We implemented an application for identifying datasets' references in a paper and matching them by corresponding titles in da|ra. We implemented the application by using python (version 3.4). The combination of the high level programming language (Python) and high performance libraries generates a perfect tool for prototyping.

As was mentioned previously, we implement our application in five packages, where each one contains several python files (Figure 6.3). In this section, we explain which libraries are used in our implementation and include a short description of each one.

- **Operator**: The module offers a set of functions to assist Python's built-in operators. For example, It has a function to define which operands and in which order should be used. We used the function in our application to sort items in lists.
- **OS**: It contains some functions to facilitate interacting with operations, such as reading/writing a file, manipulating paths, and creating directories. For example, "listdir(path)" returns the list of all files and directories in the path

given as the input of the function. This library also introduces some related functions to paths' names.

- **Sys**: The module provides interaction with an interpreter. For example, "exc_info()" in the module returns information about the exception being handled.

- **Timeit**: It provides a feature for measuring the execution time of a python code.

- **PyQt**: Qt is a C++ GUI development framework and PyQt is a Python binding for the Qt toolkit, where a sip tool generates the binding. Sip is employed for producing an API for C or C++ libraries to interact by Python classes. PyQt contains a pyuic module, which converts graphical user interfaces designed by qt to Python code.

- **Re**: This library supplies a set of functions for matching regular expression patterns with strings.

- **NLTK (Natural Language Toolkit)** [33]: It facilitates implementing a program to manipulate human language data by supplying an interface for many different linguistic resources. WordNet is one of the sources, and it employs some text processing libraries such as those for classification, tokenization, and stemming.

- **Pyenchant**: It offers a set of bindings for python to interact with the Enchant human language library. The library checks the words and also suggests corrections for words that are spelt incorrectly by utilizing different language packages.

- **Datetime**: The module introduces some functions to work with date and time, such as for parsing or formatting.

- **Math**: It contains a set of the mathematical functions such as "fabs(x) ", which is one of these functions, and returns the absolute value of x.

- **Collections**: The module adds some container data types to the existing types in python. A few of the new data types are "Counter", "Deque", and "OrderedDict". "Defaultdict" is also one of these types, and immediately creates an empty list for a key that does not exist in the object, and then appends the key's value to the list.

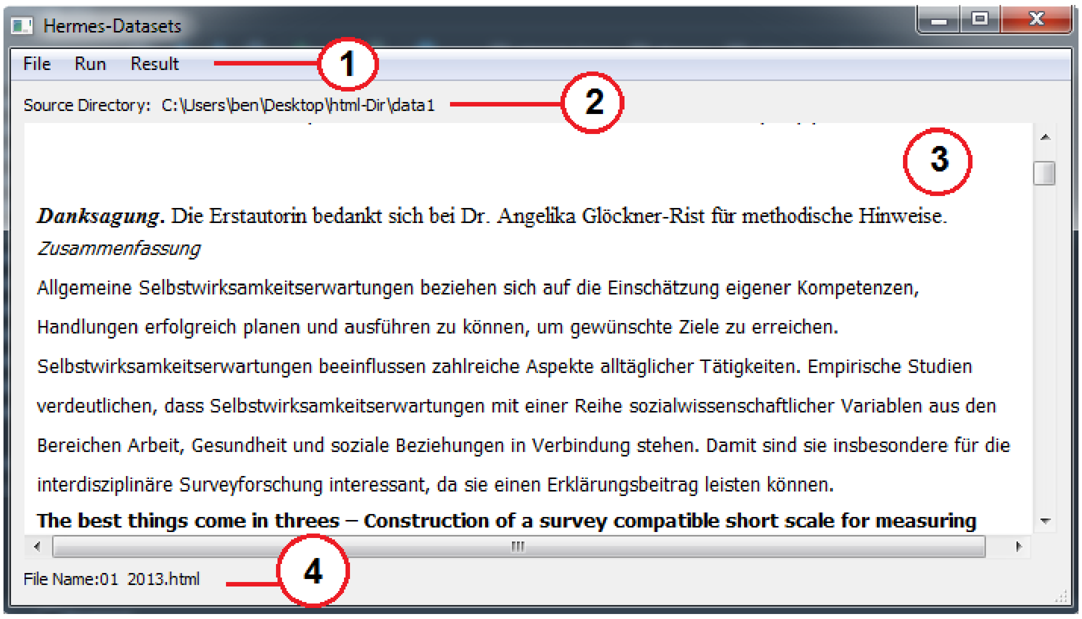
- **Gensim (Generate similar)** [34]: It is a python toolkit, which implements the vector space model. It is also implemented in order to deal with large text collections. Gensim includes implementations of algorithms such as tf-idf and cosine similarity.
- **HTMLParser**: It facilitates parsing HTML and XHTML files.
- **Win32com**: This package offers access from python to Microsoft products such as Excel and Word.
- **Urllib**: It contains some modules for handling URLs. For example, "urllib.request" is one of these modules and facilitates opening and reading URLs.
- **xml.etree.ElementTree**: The module is an API for parsing and manipulating XML data.
- **Metadata parser**: It is implemented in order to extract metadata from web documents.
- **Json**: It provides an API to convert Python objects to JSON format.

6.3 Front-End

The user interface of the prototype is designed by using the Qt designer tool. Since it is a WYSIWYG (what you see is what you get) designer, it helps us to design Windows and dialogs easily. It is very easy to wire the elements of a Qt interface with functions by employing signals and slots, and the properties of these elements can also be simply manipulated by code. The files generated by Qt have a ".ui" extension and these files contain information about the forms and dialogs in XML format. PyQt4 package is a python binding for Qt and it includes a library whose name is "pyuic4". The library facilitates converting files with a ".ui" extension to Python files. Afterwards, these Python files are easy to employ in a python program such as ours.

6.3.1 Prototype

We have designed five graphical user interface forms for our prototype. It is necessary to mention that the forms are unfortunately not designed to be user centered due to our limited time, and we focused therefore mainly on the functionality. The first form is related to the main window. The window will be displayed as the first form when a user runs the application. A screenshot of the form is illustrated in Figure 6.4.



Source Directory: C:\Users\ben\Desktop\html-Dir\data1

Danksagung. Die Erstautorin bedankt sich bei Dr. Angelika Glöckner-Rist für methodische Hinweise.

Zusammenfassung

Allgemeine Selbstwirksamkeitserwartungen beziehen sich auf die Einschätzung eigener Kompetenzen, Handlungen erfolgreich planen und ausführen zu können, um gewünschte Ziele zu erreichen. Selbstwirksamkeitserwartungen beeinflussen zahlreiche Aspekte alltäglicher Tätigkeiten. Empirische Studien verdeutlichen, dass Selbstwirksamkeitserwartungen mit einer Reihe sozialwissenschaftlicher Variablen aus den Bereichen Arbeit, Gesundheit und soziale Beziehungen in Verbindung stehen. Damit sind sie insbesondere für die interdisziplinäre Surveyforschung interessant, da sie einen Erklärungsbeitrag leisten können.

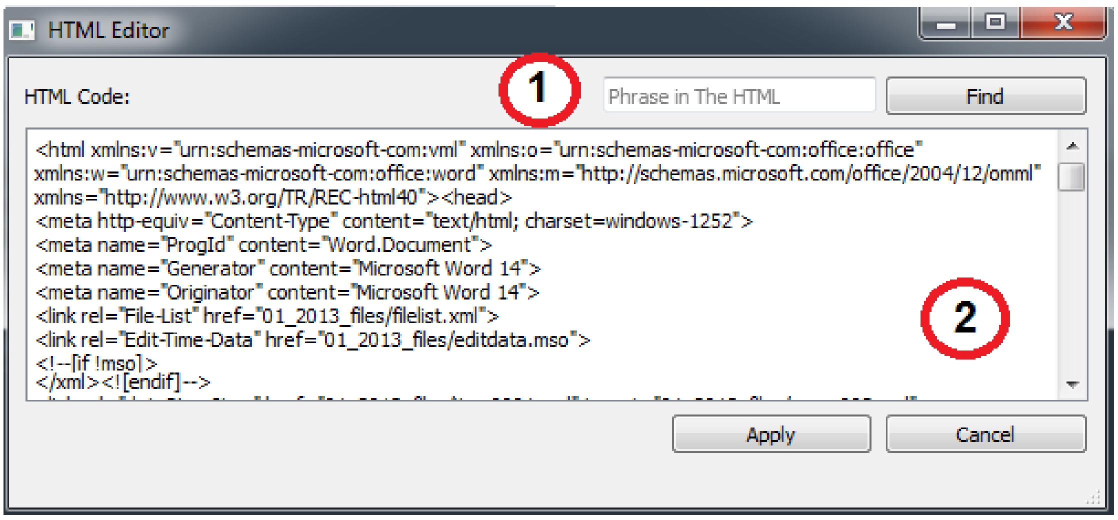
The best things come in threes – Construction of a survey compatible short scale for measuring

File Name:01 2013.html

FIGURE 6.4: The main windows form.

The form includes four main parts: 1. A menu bar, 2. A label that represents the source directory, 3. A section (Qt-WebView module) for displaying an imported paper in the application, and 4. A label that presents the name of the imported file. The menu bar in the form contains three menus ("File", "Run", and "Result"), and each menu wraps some functions. Those functions in the "File" menu focus more on importing files (i.e. papers) in the application. The second menu is "Run", which contains three main functions of the application as well as a caller to another form, which is responsible for handling the manipulation of HTML documents. Those other functions in the menu focus on the harvest of datasets' titles from the da|ra repository, extraction of special features from these titles, and finally, the detection of datasets' references in an imported paper. The "Result" contains two callers to two forms, which are responsible for both the visualization of results and for generating a JSON file from detected datasets in an imported paper.

The application converts a paper (".doc") to HTML format, and afterwards applies further analysis to the paper. Since the conversion is often not perfect, these papers' HTML documents need some editing. The second form is implemented to let users edit the HTML document of an imported paper. In the main form, a double-click on its Qt-WebView enables a user to edit the display of the imported paper directly, such as by adding a text to or removing a text from the paper in the main form. The changes sometimes cannot be applied directly, and the user

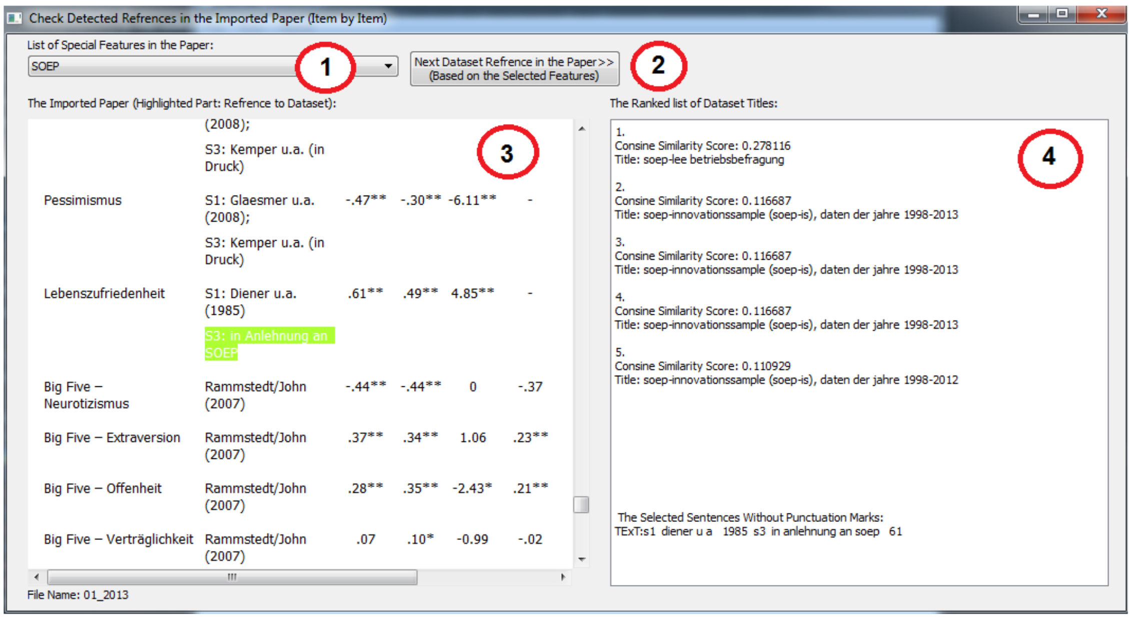


```
<html xmlns:v="urn:schemas-microsoft-com:vml" xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:w="urn:schemas-microsoft-com:office:word" xmlns:m="http://schemas.microsoft.com/office/2004/12/omml"
xmlns="http://www.w3.org/TR/REC-html40"><head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="ProgId" content="Word.Document">
<meta name="Generator" content="Microsoft Word 14">
<meta name="Originator" content="Microsoft Word 14">
<link rel="File-List" href="01_2013_files/filelist.xml">
<link rel="Edit-Time-Data" href="01_2013_files/editdata.mso">
<!--[if !mso]>
</xml><![endif]-->
```

Figure 6.5: The "HTML editor" windows form.

should change HTML document; in this case, the "HTML Editor" form will be useful. The form contains two main parts (Figure 6.5): 1. A text box for inserting a phrase query to search inside the HTML document of the imported paper, and 2. The "plain text box" in the form that displays this code. The "find" button in the "HTML Editor" form is responsible for finding the first matched phrase in the HTML document with the query phrase. Each time the user clicks on the button, the next matched phrase will be detected. The HTML document in the plain text is ready to be manipulated by a user who should finally decide to either apply the changes or to discard them.

The third form is displayed in Figure 6.6, and it is responsible for highlighting datasets' references in the imported paper. It has four main parts: 1. A drop down list that includes the list of detected features in the paper, 2. A button that handles the highlighting of detected references based on a selected feature in the drop down list, 3. A WebView displays the paper with a highlighted reference to a dataset, which includes the selected feature, and 4. A plain text box that contains a ranked list of the top five datasets' titles that are similar to the highlighted reference in the paper. A user should initially select a feature from a list of special features, and


Check Detected Refrences in the Imported Paper (Item by Item)
1
2
3
4

List of Special Features in the Paper:

SOEP

Next Dataset Refrence in the Paper>>
(Based on the Selected Features)

The Imported Paper (Highlighted Part: Refrence to Dataset):

	(2008);				
	S3: Kemper u.a. (in Druck)				
Pessimismus	S1: Glaesmer u.a. (2008);	-.47**	-.30**	-6.11**	-
	S3: Kemper u.a. (in Druck)				
Lebenszufriedenheit	S1: Diener u.a. (1985)	.61**	.49**	4.85**	-
	S3: in Anlehnung an SOEP				
Big Five – Neurotizismus	Rammstedt/John (2007)	-.44**	-.44**	0	-.37
Big Five – Extraversion	Rammstedt/John (2007)	.37**	.34**	1.06	.23**
Big Five – Offenheit	Rammstedt/John (2007)	.28**	.35**	-2.43*	.21**
Big Five – Verträglichkeit	Rammstedt/John (2007)	.07	.10*	-0.99	-.02

File Name: 01_2013

The Ranked list of Dataset Titles:

1.
Consine Similarity Score: 0.278116
Title: soep-lee betriebsbefragung

2.
Consine Similarity Score: 0.116687
Title: soep-innovationssample (soep-is), daten der jahre 1998-2013

3.
Consine Similarity Score: 0.116687
Title: soep-innovationssample (soep-is), daten der jahre 1998-2013

4.
Consine Similarity Score: 0.116687
Title: soep-innovationssample (soep-is), daten der jahre 1998-2013

5.
Consine Similarity Score: 0.110929
Title: soep-innovationssample (soep-is), daten der jahre 1998-2012

The Selected Sentences Without Punctuation Marks:
TExT:s1 diener u a 1985 s3 in anlehnung an soep 61

FIGURE 6.6: The windows form for reviewing datasets' references in an imported paper (Item by item) .

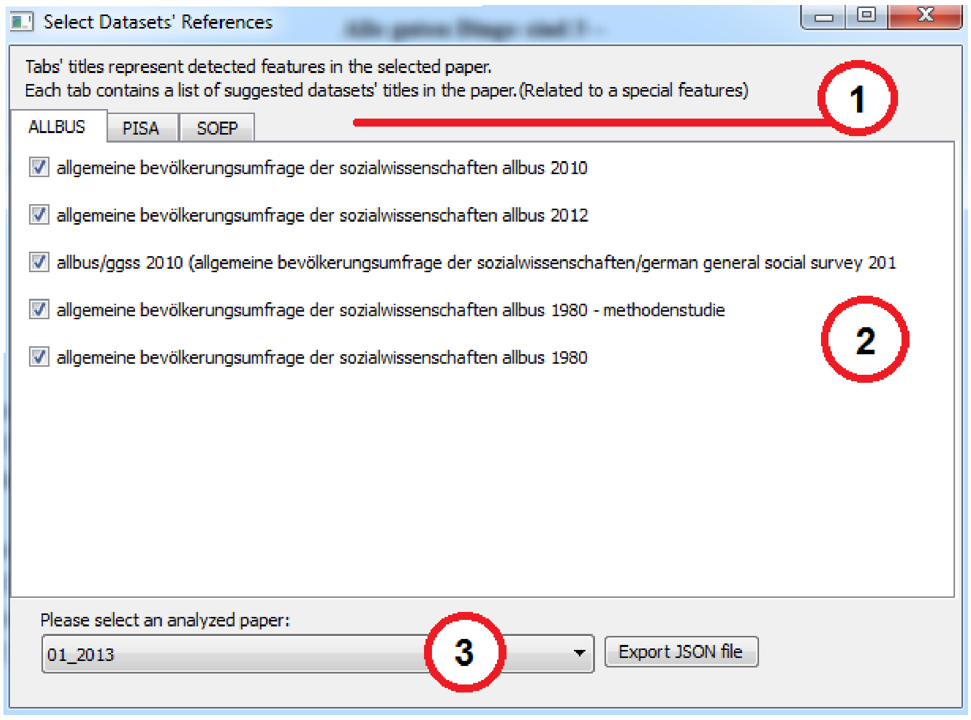
then click on the button to spot the first reference that includes the feature in the imported paper. The next time the user clicks on the button the next reference is highlighted. A list of the top five most similar titles in da|ra appears to the spotted reference in the text.

The highest item on the list is the most similar title, and items on the list become less similar from the top down. Each item on the list displays a dataset's title registered in da|ra, as well as its cosine similarity score. At the bottom of the plain text box, the highlighted reference in the paper is displayed without punctuation marks.

The fourth form provides an interface for exporting a JSON file from the suggested list of datasets' references in an analyzed paper. The list is suggested by our application, and a user can select among items on the list and then decide to generate a JSON file.

The form contains a tab widget that splits elements into some sections, which are then labeled differently to represent different categories. The titles of these tabs refer to detected features in a selected paper. Each tab contains a list of datasets' titles, which contain the specific feature through which the tab is labeled. These titles are suggested candidates of our approach for detected references in the paper.

A paper should first be analyzed, and afterwards it will be added to the drop down


Select Datasets' References

Tabs' titles represent detected features in the selected paper.
Each tab contains a list of suggested datasets' titles in the paper.(Related to a special features)

- ✓ allgemeine bevölkerungsumfrage der sozialwissenschaften allbus 2010
- ✓ allgemeine bevölkerungsumfrage der sozialwissenschaften allbus 2012
- ✓ allbus/ggss 2010 (allgemeine bevölkerungsumfrage der sozialwissenschaften/german general social survey 201
- ✓ allgemeine bevölkerungsumfrage der sozialwissenschaften allbus 1980 - methodenstudie
- ✓ allgemeine bevölkerungsumfrage der sozialwissenschaften allbus 1980

Please select an analyzed paper: 01_2013

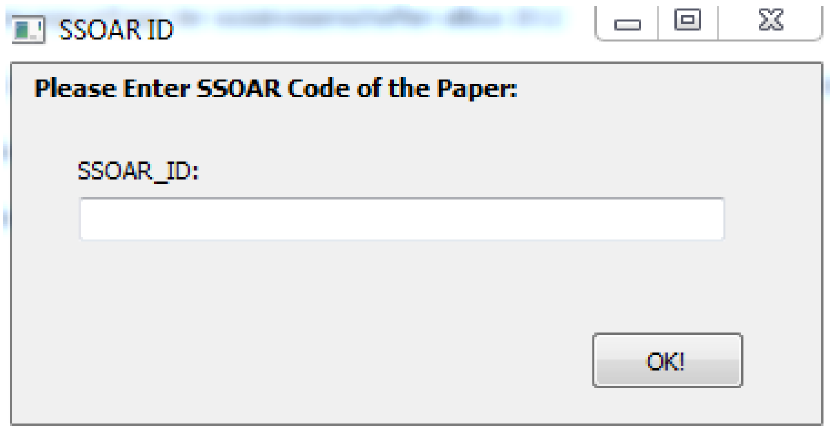
FIGURE 6.7: The windows form for exporting JSON file from datasets' references in an analyzed paper.

list at the bottom of the form that is displayed in Figure 6.7. A user should select a paper from the items on the list, and then the list of suggested datasets titles that are related to the paper will be shown in the tab widget.

The user selects which datasets' titles should be in the exported JSON file and clicks on the button in the form after making their final decision. The button shows the fifth form (Figure 6.8), and the form asks the user to insert the paper's SSOAR Identifier. A URL addresses each paper in the SSOAR repository, and the following example is related to a paper in the repository that represents the format of these URLs:

http://www.ssoar.info/ssoar/handle/document/549

The last section in the URL – which only contains digits – is the paper's SSOAR identifier. The user inserts one SSOAR identifier to specify exactly which paper is being analyzed, and then confirms that choice by clicking on the button in the fifth form. The application harvests the paper's information from the repository and merges it with the information of selected datasets titles. Finally, the information will be stored as a JSON file, which facilitates reusing the information.



Please Enter SSOAR Code of the Paper:

SSOAR_ID:

Figure 6.8: The windows form which asks the user to enter the SSOAR identifier of a paper.

6.4 Data Structure

We can define two general types of data for our application. The first type is related to input data such as mda papers, datasets' titles, and DOIs, which are explained in Chapter 4. The second type is related to output files such as JSON files, which contain application results of our approach. This section is about how our application stored the data. They will be stored in text or JSON files. The largest file in our application is less than 4MB, and is responsible for keeping datasets' information registered in da|ra.

6.4.1 The Input Data

The input data category contains two sub-categories, which are the context of an imported paper and the information of registered datasets in the da|ra repository. The conversion of papers into HTML files is often used by metadata extractor approaches since HTML can provide some more information about the context, such as the position of a specific part of context, font, and size of the part. This information can play an important role in metadata extraction.

In our case, however, since a reference to a dataset may appear anywhere in a paper and does not follow any special patterns, this extra information is not helpful in identifying a dataset reference in a paper. Our application only extracts texts from papers and does not consider the structure of HTML.

The reason we still stick to HTML format and not just a simple text file is that HTML helps us to easily display papers in our application. It also helps us to easily identify the position of a detected reference in a paper.

Each imported paper should first either be converted into HTML format or be directly imported from a HTML source file. Since the file corpus that we possess was in ".doc" format, we implemented the feature of conversion from ".doc" file to "HTML" format. However, many tools and libraries currently exist for converting other formats such as "PDF" into HTML format. For instance, "poppler"[1] is a binding for python, which facilitates converting "PDF" into HTML format. In our application, we used the MS office application itself. Our application calls MS office and asks it to do this conversion. Since ".doc" format is Microsoft's suggestion, the best and the easiest converter is through MS office.

The datasets' information sub-category covers three files. One of the files is responsible for keeping the harvested titles and DOIs of datasets from the da|ra repository. Each row of the file refers to a dataset and the row contains the DOI and the title of the dataset, which are separated by a delimiter. A unique string is defined as the delimiter, and an example of these rows is:

Eurobarometer 79.2 (2013) #this is BEN_DOI# 10.4232/1.11798

In the example, "Eurobarometer 79.2 (2013) " is the title of the dataset. Since we tried to assign a unique string that can never or only rarely be part of a dataset's title, our selected delimiter is "#this is BEN DOI#". Finally, the last part is the dataset's DOI, which in this case is "10.4232/1.11798".

By possessing the file, our application generates two other files that carry titles and DOIs of datasets separately. This means one contains the DOIs and another one includes the titles. Two rows in these two files with a same index refer to a dataset.

6.4.2 The Output Data

The output files can be split into two sub-categories, which are "Special features" and "Result". The first subcategory is "Special features", and it covers four files.

[1]https://poppler.freedesktop.org

Our application is able to extract special features from titles of the datasets. The special features are also categorized into the two groups “abbreviation” and “special phrase”, and these groups are stored in two text files. For example, an “abbreviation” file includes the list of the abbreviations extracted from the dataset titles, and each row refers to an abbreviation.

There are two more files in the subcategory: “False positives” and “Extra features”. While the extracted features may contain some false positive entities, the list may also fail to contain some true candidates from the datasets’ titles. These entities can be added or removed directly from “abbreviation” and “special phrase”, but it has the problem that each time after extraction of special features from the titles, these entities should be manually added or removed. Thereupon a user can add these entities to their related list (i.e. “False positives” and “Extra features”), and going forwards they will be added or removed automatically each time.

The second subcategory is “Result”, which contains three types of files. Each reference in a paper and its list of the top five most similar datasets’ titles to the reference are displayed in the third form (ref. Fig 6.6), while the suggested datasets’ references in the paper are displayed in the fourth form.

Each paper possesses two folders after our algorithm is applied. The first folder contains the files, each of which is related to a reference in the paper and includes its related ranked list of datasets’ titles. These files store each title like the example below:

Value:0.956437
Text:allgemeine bevölkerungsumfrage der sozialwissenschaften allbus 2010

The first line indicates cosine similarity score of the title, and the second line denotes the title of a dataset in lower case. The next entity in the ranked list will be stored immediately after the current entity in the same format.

The second folder contains text files, each of which is related to a detected special feature in the paper, and it also contains associated suggested datasets’ titles to the special feature for the paper. Each title is formatted like the following instance in these files:

1____pisa–i–plus 2003, 2004____0.196164

These files are related to pre-characteristic feature flow (ref. 5.4 and 7.2). Each line refers to a dataset's title in da|ra and contains three parts, with the first part concerning the frequency of the title's repetition in the related special feature category. The title itself is the second part in the line, and, finally, the third part shows the highest similarity score of the title in the category of related special feature. These parts are separated by a unique delimiter, which is formed by four underlined symbols in a row.

The last types of files in the "Result" subcategory are JSON files, which are the container of outputs of the fourth form. A user decides among suggested datasets' titles, and then clicks on the button for generating a JSON file. The file contains some information about the paper and the selected datasets. As mentioned before, the user enters the SSOAR identifier of the paper and then the application collects some of the paper's related metadata from the SSOAR repository. This metadata contains the title, alternative title, author(s), publication date, description, language, publisher, and URN.

Afterwards, this metadata will be merged with the information of the selected datasets, where the DOI and title of each dataset form the information. The following example demonstrates the structure of these JSON files:

```json
{

"@context": "http://schema.org",

"@type": "ScholarlyArticle",

"alternativeHeadline":  "Four short scales for measuring
    the  personality trait of \"justice sensitivity\"",

"author": "Rammstedt, Beatrice",

"citation": [
      {

         "@type": "Dataset",

         "headline": "german general social survey -
          allbus 2000 -   capi-papi",

         "sameAs": [
          "http://dx.doi.org/10.4232/1.10106"
                                     ]

      }

         ],

"datePublished": "2013",

"description": "Published Version",

"headline": "Vier Kurzskalen zur Messung des
   Persoenlichkeitsmerkmals \"Sensibilitaet fur
   Ungerechtigkeit\"",

"inLanguage": "de",

"publisher": "DEU",

"sameAs": "http://nbn-resolving.de/http://dx.doi.org/10.
   12758/mda.2013.015"

}
```

"Schema.org" offers schemas for structured data on the internet, and it currently contains 642 Types and 992 Properties, where the types are organized in a hierarchy and each type includes a set of properties.

"Thing" is the most universal type of item, and it can be more specific types such as Creative Work, Event, Intangible, Organization, Person, and Place. These specific types may be assumed as generic for other types; for example, creative work is generic for types such as Dataset and Scholarly Article. Table 6.2 contains the information of those properties which are used in our output JSON files' structure.

Properties from Creative Work		
Property	Expected Type	Description
headline	Text	Title of the item.
alternativeHeadline	Text	A secondary title of the item.
author	Person or Organization	The author of this content.
citation	Creative Work or Text	A reference to another creative work, such as another publication, dataset, etc.
inLanguage	Text or Language	The language of the content.
datePublished	Date	Date of first publication.
publisher	Person or Organization	The publisher of the item.

Properties from Thing		
Property	Expected Type	Description
description	Text	A short description of the item.
sameAs	URL	A URL to address the items on the web explicitly.

TABLE 6.2: Output JSON files' properties.

Chapter 7

Evaluation

The calculation of evaluation metrics such as precision, recall, and F-measure require ground truth. We therefore selected a test corpus of 15 random papers from the 2013 and 2014 issues of the mda journal – six in English and nine in German.

A trained assessor from the InFoLiS II project at GESIS reviewed all papers one by one and identified all references to datasets. Afterwards, the assessor attempted to discover at least one correct match in da|ra for each detected reference, and the result is a set of lists of datasets per paper. These lists were used as a gold standard to compare with the results of our algorithm in order to examine differences and similarities.

7.1 Evaluation Process and Description

We decided to divide our evaluation into two steps. The first step focuses on identifying dataset references in papers. Here, accuracy depends on the quality of the generated dictionaries of abbreviations and special phrases (the accuracy metrics used in the thesis are explained in 2.4).

Our algorithm searches these characteristic features in the full texts; detection of any of these features may lead to the detection of a dataset reference (see row "Detection" in table 7.1). In this phase, if a characteristic feature is identified both in a paper and in the gold standard, it will be labeled as a true positive. If the feature is in the gold standard but not in our output, it will be labeled as a false negative, or as a false positive in the opposite case.

The second step of the evaluation is about the accuracy of matching detected references in papers with datasets titles in the da|ra registry. This evaluation phase considers only true positives from the previous step. The lists of suggested matches for an item, both from the gold standard and from our output, are compared in this step. Since a dataset may occur on its own or be integrated together with other studies, an item can have more than one true match (e.g. Allbus 2010 in ALLBUScompact 1980–2012). In this step, an item will be labeled as a false negative if none of the suggestions for the item in the gold standard appears in the output of our algorithm. The number of false positives and false negatives are equal in this step, since a missing corresponding match means the possession of false positives. True positives, false positives, and false negatives are counted and then used to compute precision and recall.

The third row in table 7.1 refers to the accuracy of two phases of the algorithm as one unit in order to find how well it works generally, and does not consider one specific section (i.e. identification or matching). In order to satisfy this purpose, we repeated the second phase of evaluation, but this time included all data from the first step and not only the true positives. If an item is identified as false positive in the first section of evaluation, it is labeled as such in the evaluation as well.

7.2 Evaluation Results

The algorithm gains high precision in both the detection and matching phases, which means it has a much smaller number of wrong predictions. It also covers the majority of relevant data, which leads to high recall. The results of evaluations that we calculated are shown in table 7.1.

Our observations in the second evaluation step confirm the choices of set size in the interactive disambiguation workflows. In the per-reference matching workflow (as mentioned in 5.4), a ranked list of dataset titles is generated for each of the 45 dataset references (on average in our corpus) in a paper by employing a combination of cosine similarity and tf-idf.

Phase of Evaluation	Precision	Recall	F-measure
Detection	0.91	0.77	0.84
Matching	0.83	0.83	0.83
Detection+Matching	0.76	0.64	0.7

TABLE 7.1: Results of the Evaluation

Our observation shows that the correct match among da|ra dataset titles for each reference detected is in the top five items of the ranked list generated by combining cosine similarity and tf-idf for that reference. Therefore, we adjusted our implementation to only keep the top five items of each candidate list for further analysis, such as an expert user's interactive selection of *the* right dataset for a reference.

The per-feature matching workflow (as mentioned in 5.4) categorizes references by characteristic features. For example, in a paper that contains exactly three detected characteristic features – "ALLBUS", "PIAAC", and "exit poll" – each dataset reference relates to one of these three features. If we obtain for each such reference the list of top five matches as in the per-reference workflow and group these lists per category, we can count the number of occurrences of each dataset title per category.

Now, looking at the dataset titles per category sorted by ascending number of occurrences, we observed that the correct matches for the datasets' references using a specific characteristic feature were always among the top six items.

Chapter 8

Conclusion and Future Work

8.1 Summary

We have presented an approach for identifying references to datasets in social science papers and pairing each of the references with at least one item in da|ra. Our algorithm works in real time and does not require any training dataset.

Since it contains few manual tasks such as initially cleaning the list of extracted abbreviations and phrases from datasets titles in da|ra, it is a semi-automatic approach. This is also the case because a user should make final decisions between candidates suggested for the datasets cited by the given paper. We have achieved an F-measure of 0.84 for the detection task and an F-measure of 0.83 for finding correct matches for each reference in the gold standard.

8.2 Limitations

There are some issues that impact the results of our approach. Authors sometimes make a reference to a dataset by using the underlying publication of a study without mentioning the dataset title or abbreviation. This kind of citation is one of the challenges, since it is impossible to detect these datasets based on their titles. Although the da|ra registry is large and is growing fast, there are still many datasets that have yet to be registered there. This circumstance will adversely affect the task of detecting dataset references in papers and matching them to corresponding items in da|ra.

After the evaluation, our observation reveals that in our test, the corpus contained 25 different references to datasets, and da|ra could cover only 64 percent of the datasets in our test corpus. The next problem is the abbreviations that are commonly used by a dataset and a none-dataset item. These limitations cause false positives and false negatives, which adversely impact the evaluation of our algorithm.

8.3 Future Works

Future work will focus on improving the accuracy of detecting references to the datasets supported so far, and on extending the coverage to all datasets as well. Accuracy can be improved by better similarity metrics, such as taking synonyms into account. For example, common words can be replaced by their synonym when the title of a dataset contains "survey" but it is cited as "study".

Currently, our approach only considers datasets' titles. Including further dataset metadata can improve our approach significantly. da|ra contains this metadata, and it is possible to extract the references easily through the repository's API.

Other algorithms such as identifying the central dataset(s) on which a paper is based can improve the ranked list generated by similarity metrics. The identification of central dataset(s) is possible after pairing a share of dataset references in a given paper with titles in da|ra; this identification then affects the ranking of rest of the references.

Coverage can be improved by taking further datasets not registered in da|ra into account. One promising further source of datasets is OpenAIRE, the Open Access Infrastructure for Research in Europe, which so far covers more than 16,000 datasets from all domains – including the social sciences – and is growing rapidly thanks to the increasing attention paid to open access publishing in the EU.

The OpenAIRE metadata can be consumed via OAI-PMH, or, in an even more straightforward way, as linked data (cf. Vahdati et al. [35]). Furthermore, we will enable further reuse scenarios by exporting RDF from the per-reference matching workflow using state-of-the-art annotation.

Employing more patterns leads to covering more datasets, which makes it possible to extend the algorithm's coverage by finding more patterns in datasets titles.

Using a combination of the titles and other metadata can also extend the coverage. We observed that using only a similarity algorithm for linking sentences in a given paper to the datasets' titles in da|ra is not enough, and it is necessary to find some patterns from these titles beforehand. Further research may investigate whether employing a combination of datasets' metadata makes it possible to use only a similarity algorithm.

In a mid-term perspective, solutions for identifying dataset references in papers already published could be made redundant by a wider adoption of standards for properly citing datasets while authoring papers and corresponding tool support for authors.

Bibliography

[1] Behnam Ghavimi, Philipp Mayr, Sahar Vahdati, and Christoph Lange. Identifying and improving dataset references in social sciences full-texts. *International Conference on Electronic Publishing (ELPUB)*, Göttingen, Germany, 7-9 June 2016. URL http://arxiv.org/abs/1603.01774.

[2] Daniel Hienert, Frank Sawitzki, and Philipp Mayr. Digital library research in action: Supporting information retrieval in sowiport. *D-Lib Magazine*, 21 (3/4), 2015. doi: 10.1045/march2015-hienert.

[3] Brigitte Mathiak and Katarina Boland. Challenges in matching dataset citation strings to datasets in social science. *D-Lib Magazine*, 21(1/2), 2015. doi: 10.1045/january2015-mathiak.

[4] Sunita Sarawagi. Information extraction. *Foundations and Trends in Information Retrieval in Databases*, 1(3):261–377, 2007. doi: 10.1561/1900000003.

[5] Meiyu Lu, Srinivas Bangalore, Graham Cormode, Marios Hadjieleftheriou, and Divesh Srivastava. A dataset search engine for the research document corpus. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1237–1240. IEEE, 2012. doi: 10.1109/ICDE.2012.80.

[6] Micah Altman and Gary King. A proposed standard for the scholarly citation of quantitative data. *D-Lib Magazine*, 13(3/4), 2007. doi: 10.1045/march2007-altman.

[7] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007. doi: 10.1075/li.30.1.03nad.

[8] SJ Pepler and K O'Neil. Preservation intent and collection identifiers: Claddier project report ii, 2008. URL http://purl.org/net/epubs/work/43640.

[9] Allen H Renear, Simone Sacchi, and Karen M Wickett. Definitions of dataset in the scientific and technical literature. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, 2010.

[10] Najla Semple. DCC briefing paper: Digital repositories. 2006. URL `https://www.era.lib.ed.ac.uk/handle/1842/3372`.

[11] Gerard Salton and Christopher Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[12] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[13] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML 97 Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann, 1997.

[14] David M W Powers. *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation*, volume 2. Journal of Machine Learning Technologies, 2011.

[15] Bart Lamiroy and Tao Sun. *Computing Precision and Recall with Missing or Uncertain Ground Truth.* GREC 11. Springer-Verlag, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-36824-0_15.

[16] Sungjick Lee and Han-joon Kim. News keyword extraction for topic tracking. In *Networked Computing and Advanced Information Management, 2008 (NCM 08)*, volume 2, pages 554–559. IEEE, 2008.

[17] Peter D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL 01, pages 491–502. Springer-Verlag, 2001. URL `http://dl.acm.org/citation.cfm?id=645328.650004`.

[18] Ayush Singhal and Jaideep Srivastava. Data extract: Mining context from the web for dataset extraction. *International Journal of Machine Learning and Computing*, 3(2), 2013. doi: 10.7763/IJMLC.2013.V3.306.

[19] Christoph Schaefer, Daniel Hienert, and Thomas Gottron. Normalized relevance distance - a stable metric for computing semantic relatedness over reference corpora. In *European Conference on Artificial Intelligence (ECAI)*, volume 263, pages 789–794, 2014. doi: 10.3233/978-1-61499-419-0-789.

[20] Mehran Sahami and Timothy D. Heilman. *A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets.* WWW 06. ACM, 2006. doi: 10.1145/1135777.1135834.

[21] Muhammad Tanvir Afzal, Hermann Maurer, Wolf-Tilo Balke, and Narayanan Kulathuramaiyer. *Rule Based Autonomous Citation Mining with Tierl.* Journal of Digital Information Management 8(3) 196-204, 2010.

[22] Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. Keyword extraction using support vector machine. In *Proceedings of the 7th International Conference on Advances in Web-Age Information Management*, WAIM 06, pages 85–96. Springer-Verlag, 2006. doi: 10.1007/11775300_8.

[23] Hui Han, C. L. Giles, E. Manavoglu, Hongyuan Zha, Zhenyue Zhang, and E. A. Fox. *Automatic Document Metadata Extraction Using Support Vector Machines.* ACM/IEEE 2003 Joint Conference on Digital Libraries, 2003. doi: 10.1109/JCDL.2003.1204842.

[24] Jasmeen Kaur and Vishal Gupta. Effective approaches for extraction of keywords. *IJCSI International Journal of Computer Science*, 7(6):144–148, 2010.

[25] Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo Wang. Automatic keyword extraction from documents using conditional random fields. *Computational and Information Systems*, 4(3), 2008.

[26] Bin-Ge Cui and Xin Chen. An improved hidden markov model for literature metadata extraction. In *6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China*, 2010. doi: 10.1007/978-3-642-14922-1_26.

[27] Francis Kubala, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. Named entity extraction from speech. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 287–292, 1998.

[28] Simone Marinai. Metadata extraction from PDF papers for digital library ingest. In *Document Analysis and Recognition, 2009. ICDAR 09. 10th International Conference on*, pages 251–255. IEEE, 2009. doi: 10.1109/ICDAR.2009.232.

[29] R. Kern, K. Jack, M. Hristakeva, and M. Granitzer. *Teambeam-Meta-Data Extraction from Scientific Literature.* D-Lib Magazine 18(7/8), 1, 2012. doi: 10.1045/july2012-kern.

[30] Katarina Boland, Dominique Ritze, Kai Eckert, and Brigitte Mathiak. Identifying references to datasets in publications. In *Proceedings of the Second International Conference on Theory and Practice of Digital Libraries (TDPL 2012)*, pages 150–161. Springer, 2012.

[31] Martin Fowler. *UML distilled: a brief guide to the standard object modeling language.* Addison-Wesley Professional, 2004.

[32] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The (2Nd Edition).* Pearson Higher Education, 2004. ISBN 0321245628.

[33] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python.* O'Reilly Media, Inc., 2009.

[34] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010. ELRA. URL http://is.muni.cz/publication/884893/en.

[35] Sahar Vahdati, Farah Karim, Jyun-Yao Huang, and Christoph Lange. Mapping large scale research metadata to linked data: A performance comparison of HBase, CSV and XML. In *Metadata and Semantics Research*, Communications in Computer and Information Science. Springer, 2015. doi: 10.1007/978-3-319-24129-6_23.

Appendix A – Gold Standard

Table 1 contains the information of papers in our test corpus. An assessor reviewed all papers in test corpus and identified all references to datasets in the papers (more details in Chapter 7).

Index	**SSOAR_Id**	**URN or DOI**	**Datasets' References in the Paper**
1	37421	http://dx.doi.org/10.12758 /mda.2013.014	Pisa 2000, Allbus 2010, SOEP 2009
2	42893	http://dx.doi.org/10.12758 /mda.2014.005	PIAAC 2011–2012, ESS 2012, Pisa, ALLBUS 2008
3	38448	http://dx.doi.org/10.12758 /mda.2013.016	SOEP, Panel study of income, Biritish Household Panel Survey, Schweizer Umweltsurvey 2007, Mikrozensus 2005
4	37420	http://dx.doi.org/10.12758 /mda.2013.013	soep 2009, soep 2005, ALLBUS 2010, International Social Survey Programme(ISSP)
5	35915	http://dx.doi.org/10.12758 /mda.2013.004	ALLBUS 2010 (doi inside paper), SHARE 2010, ESS 2010, EVS 2008, European Quality of life Survey(EQLS) 2011, National Adult Literacy Survey, ALLBUS 1994–2008

Index	**SSOAR_Id**	**URN or DOI**	**Datasets' References in the Paper**
6	42884	http://dx.doi.org/10.12758 /mda.2014.006	PIAAC 2011–2012(doi inside paper), International Adult Literacy Survey, German National Education Panel Study (NEPS), Adult Literacy and Lifeskills (ALL)
7	42886	http://dx.doi.org/10.12758 /mda.2014.007	ALLBUS 2012, PIAAC 2011–2012, ESS, IALS 1994–1998, ALL 2002–2008, SHARE 2005
8	38445	http://dx.doi.org/10.12758 /mda.2013.018	ALLBUS 2012
9	37419	http://dx.doi.org/10.12758 /mda.2013.012	Soep, ALLBUS
10	42887	http://dx.doi.org/10.12758 /mda.2014.008	PIAAC 2011–2012,SHARE 2011
11	38444	http://dx.doi.org/10.12758 /mda.2013.019	ALLBUS 2008, ALLBUS 1994
12	37417	http://nbn-resolving.de /urn:nbn:de:0168-ssoar-374171	SOEP
13	42891	http://dx.doi.org/10.12758 /mda.2014.011	PIAAC 2011–2012
14	37422	http://dx.doi.org/10.12758 /mda.2013.015	ESS 2008, ALLBUS 2000, SOEP 2009

Index	**SSOAR_Id**	**URN or DOI**	**Datasets' References in the Paper**
15	38443	http://dx.doi.org/10.12758 /mda.2013.020	SOEP, Transitions from Education to Employment(TREE 2008), PISA 2000, LV-Panel 71 Frühe Karrieren und Familiengründung: Lebensverläufe der Geburtskohorte 1971 in Ost – und Westdeutschland, British Household Panel Survey, TREE 2010

Table 1: The information of papers in our test corpus

Appendix B – Datasets References in Our Test Corpus

Table 2 contains information about a selection of identified datasets in our test corpus and their DOIs. Datasets in da|ra often have a DOI but in some cases, they may only be addressed by a url. We included only datasets whose DOIs were available in da|ra. We mentioned a big cumulative version for each dataset whose many different versions had been detected in our test corpus.

Dataset	**DOI**
Pisa 2000	http://dx.doi.org/10.5159/IQB_PISA_2000_v1
SOEP	http://dx.doi.org/10.5684/soep.v29
ALLBUS	http://dx.doi.org/10.4232/1.11952
SHARE 2010	http://dx.doi.org/10.6103/SHARE.w3.100
SHARE 2005	http://dx.doi.org/10.3886/ICPSR22160.v2
ISSP	http://dx.doi.org/10.4232/1.12312
EVS 2008	http://dx.doi.org/10.4232/1.11004
Detroit Area Study	http://dx.doi.org/10.3886/ICPSR02880.v2
PIAAC	http://dx.doi.org/10.4232/1.11865
NEPS	http://dx.doi.org/10.5157/NEPS:SC1:1.0.0
ALL 2002–2008	http://dx.doi.org/10.4232/1.11952

Table 2: A selection of detected datasets in our test corpus with their DOIs

Appendix C – Source Code

The source code of the implementation of the application presented in this thesis are uploaded in the git repository with the following URL:

https://github.com/EIS-Bonn/Theses/tree/master/2015/Behnam-Ghavimi