A Semi-Automatic Approach for Detecting Dataset References in Social Science Texts

Behnam GHAVIMI [a,b,1], Philipp MAYR [a,2], Christoph LANGE [b,c,3],
Sahar VAHDATI [b,4] and Sören AUER [b,c,5]

[a] *GESIS – Leibniz Institute for the Social Sciences*
[b] *Enterprise Information Systems (EIS), University of Bonn*
[c] *Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS*

Abstract. Today, full-texts of scientific articles are often stored in different locations than the used datasets. Dataset registries aim at a closer integration by making datasets citable but authors typically refer to datasets using inconsistent abbreviations and heterogeneous metadata (e.g. title, publication year). It is thus hard to reproduce research results, to access datasets for further analysis, and to determine the impact of a dataset. Manually detecting references to datasets in scientific articles is time-consuming and requires expert knowledge in the underlying research domain. We propose and evaluate a semi-automatic three-step approach for finding explicit references to datasets in social sciences articles. We first extract pre-defined special features from dataset titles in the da|ra registry, then detect references to datasets using the extracted features, and finally match the references found with corresponding dataset titles. The approach does not require a corpus of articles (avoiding the cold start problem) and performs well on a test corpus. We 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.

Keywords. Information extraction, Link discovery, Data linking, Research data, Social Sciences, Scientific articles, Data registry

1. Introduction

By its very nature science is based on evidence and facts. In many scenarios, these emerge from data obtained from measurements, surveys, experiments or simulations. As a result, scientific articles – the key artifact of scholarly communication – must refer to the ground truth the findings are based on, i.e. data. Making these data references explicit, machine-readable and semantics-aware is one of the major challenges in digitizing scholarly communication effectively. Especially in the quantitative social sciences and

---

[1]E-mail: behnam.ghavimi@gesis.org

[2]E-mail: philipp.mayr@gesis.org

[3]E-mail: langec@cs.uni-bonn.de

[4]E-mail: vahdati@cs.uni-bonn.de

[5]E-mail: auer@cs.uni-bonn.de

but also most other fields of science many articles reference research datasets which they are based on. For example, an article might present the results of a statistical analysis performed on a dataset comprising employment data in the European Union. Today, digital libraries aim at providing resources with high-quality metadata, easy subject access, and further support and guidance for retrieving information [1]. However, dataset references are usually not explicitly exposed in digital libraries. In most cases the articles do not provide explicit links that give readers direct access to the referenced datasets.

Explicit links from scientific publications to the underlying datasets and vice versa are useful in many scenarios, including:

- reviewers aiming to reproduce the evaluation that authors performed on a dataset,
- other researchers desiring to perform further analysis on a dataset that was used in a article,
- decision makers seeking to determine the impact of a given dataset or to identify the most used datasets in a given community.

Currently, the majority of published articles lack such direct links to datasets. While there exist registries that make datasets citable, e.g., by assigning them a digital object identifier (DOI), they are usually not integrated with authoring tools. Therefore, in practice, authors typically cite datasets by *mentioning* them, e.g., using combinations of title, abbreviation and year of publication (see, e.g., **(author?)** [2]).

Manually detecting references to datasets in articles is time consuming and requires expert knowledge of the article's domain. Detecting dataset references automatically is challenging due to the wide variety of styles of dataset citations in full-texts even within one research community, and the variety of places in which datasets can be referenced in articles (illustrated in Figure 1). So it is a difficult task to create a training set manually for solving this issue and this variance even makes rule-based approaches difficult, as it is hard to cover all cases. We therefore introduce a semi-automatic approach that parses full-texts and finds exact matches with high precision and without requiring a training set.

The remainder of this section states the problem addressed by our research more precisely. Section 2 introduces the preliminaries of techniques and metrics that we apply. Section 3 reviews existing literature related to the dataset reference problem. All data used by our approach is explained in section 4, while section 5 introduces the proposed solution. The evaluation is then presented in section 6. Section 7 concludes with an outlook to future work.

1.1. Problem Statement

Whereas a lot of effort has been spent on information extraction task in general [3], few attempts have focused on the specific use case of dataset reference extraction (see, e.g., [4]). When referring to the same dataset, different authors often use different names or keywords. Although there are proposed standards for dataset citation in full-texts, researchers still ignore or neglect such standards (see, e.g., [5]). Since the structure of scientific documents does not always follow a standard, and since datasets are rarely being linked to in a consistent way, simple keyword or name extraction approaches do not solve the problem [6]. Table 1 shows concrete examples of different referencing styles in five articles from the mda journal (see Section 4.1) that have cited different ver-
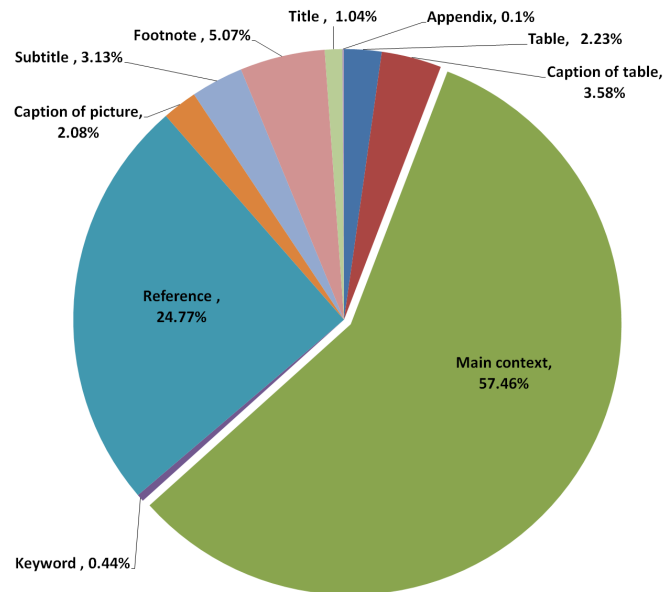
Figure 1. The distribution of more than 500 dataset references in 15 random articles from the mda journal (see Section 4.1)

sions of a study (ALLBUS/GGSS = Allgemeine Bevölkerungsumfrage der Sozialwissenschaften/German General Social Survey).

Article	Article DOI	Citation style
A	10.12758/mda.2013.014	ALLBUS (2010)
B	10.12758/mda.2013.004	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.
C	10.12758/mda.2013.012	ALLBUS (Allgemeinen Bevölkerungsumfrage der Sozialwissenschaften)
D	10.12758/mda.2014.007	(e.g., in the German General Social Survey, ALLBUS; see Wasmer, Scholz, Blohm, Walter and Jutz, 2012)
E	10.12758/mda.2013.019	Die Einstellungen zu Geschlechterrollen wurden mit Hilfe von Items aus den ALLBUS – Wellen 1994 und 2008 operationalisiert.

Table 1. Citation styles for a study in five different articles.

The challenge that our research aims at is to turn each dataset reference detected in a article into an explicit link, for example, using the DOI of the dataset entry in a dataset registry. Dataset registries usually provide further metadata about datasets, which can facilitate the detection of such links, such as the dataset creator, publication date,

description, and temporal coverage. In our case, references to datasets should be linked to items in the da│ra registry, which covers datasets from the social sciences and economics.

1.2. Contribution

This article makes the following contributions:

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

2. Preliminaries

In this section we introduce key terminology and concepts as well standard metrics for ranking the results of a search query. Search queries are used for retrieving text fragments in an article that refer to a dataset.

2.1. Dataset (Terminology)

Dataset is an ambiguous term; different authors have suggested a great variety of definitions [7]. **(author?)** introduced a general notion of the term based on the definitions in the technical and scientific literature. They define a dataset in terms of its four basic characteristics *grouping*, *content*, *relatedness*, and *purpose*:

- *Grouping.* A dataset is considered a *group* of data. *Set*, *collection*, *aggregation*, and *atomic unit* are some cases of this feature type. For instance, a dataset may have *set* semantics (e.g. "Set of RDF triples") [8], i.e. following the mathematical definition of a set, or it may have *collection* semantics, which means that deletion and addition of data does not have any effect on the dataset's identity.
- *Content.* The *content* feature describes the data in a dataset. For example, *observation* describes propositional content, while *value* refers to measurable content.
- *Relatedness.* A dataset is a group of *related* data. The *relatedness* feature thus clarifies the inter-relation of data in a dataset. *Syntactic* and *semantic* are some examples of this feature.
- *Purpose.* The *purpose* feature refers to the idea of the scientific activity which the dataset is created for.

2.2. Weighting terms in documents using tf-idf

The *bag of words* model represents a text as a set of terms without considering their order. Based on this model, documents and the query can be displayed in different ways, such as binary vectors, count vectors, and weight vectors.

In a weight matrix, each row represents a term and each column represents the vector of a document or query. 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 [9]. 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 using a logarithm in the formula is that a high 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.

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

df_t is the number of documents in the corpus that contain t, so the more a term is repeated in the corpus, the term less informative it becomes. 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 defines idf, where N is the number of documents in the corpus.

$$idf = \log_{10}(N/df_t)$$

Tf-idf is defined as the product of *tf* and *idf*.

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

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, tf-idf 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 enables users to find patterns: if a document matches the pattern, the result is "true"; otherwise the result is "false". In other words, documents either do or do not satisfy a query expression. A *ranked* retrieval model is more sophisticated, in that it 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 examples of approaches for ranking a list of documents within a query (cf. **(author?)** [10]). Matching Coefficient finds the numbers of terms that occur in both the query and document vectors. It calculates the cardinality of the intersection of each document and the query.

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

Dice Coefficient, Overlap Coefficient, and Jaccard try to normalize the Matching Coefficient in different ways:

- $DiceCoefficient = \frac{2|d \cap q|}{|d|+|q|}$
- $OverlapCoefficient = \frac{|d \cap q|}{\min(|d|,|q|)}$
- $JaccardCoefficient = \frac{|d \cap q|}{|d \cup q|}$

Each of them has specific shortcomings; 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 converted into a weight vector (see Section 2.2), 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 [9, 10], i.e.

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

It normalizes vectors by converting them to their unit vector, thus making documents of different lengths comparable. Since Euclidean distance is not effective for vectors of different lengths, it ranks documents by angle instead of distance. Between 0 and 180 degrees, the cosine function decreases monotonically, and therefore larger angles mean less 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 [11].

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 article. The algorithm should find references of datasets in a article and then detect a exact match for each detected reference in a text. These matches are to be selected from titles of datasets in a given dataset registry.

Evaluation metrics such as *precision and recall* determine the reliability of binary classifiers. For computing a unidimensional ranking of these two dimensions, one typically uses the F-measure, which is defined as the harmonic mean of precision and recall. These three metrics are defined as follows [12].

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

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

3. Related work

While only a few works address 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. Related work can be divided into three main groups covered by the following subsections.

3.1. Methods based on the "bag of words" model

One group of methods is based on the "bag of words" model using algorithms such as tf-idf to adjust weights for terms in a representation of texts as vectors (cf. the introduction in Section 2.2). **(author?)** proposed an unsupervised keyword extraction method by using a tf-idf model with some heuristics [13]. Our approach uses similarity measures for finding a perfect match for each dataset reference in a article by comparing titles of datasets in a registry to sentences in articles. Dice, Jaccard and Cosine can be applied to a vector representation of a text easily (cf. **(author?)** [10]). The accuracy of algorithms based on such similarity measures can be improved by making them semantics-aware, e.g., representing a set of synonyms as a single vector space dimension.

3.2. Corpus and Web based methods

Corpus and Web based methods often use information about the co-occurrence of two texts in documents, and are used for measuring texts' semantic similarity. **(author?)** introduced a simple unsupervised learning algorithm for detecting synonyms [14], which searches queries through an online search engine and analyses the results. The quality of the algorithm depends on the number of search results returned.

(author?) proposed an approach to extract dataset names from articles [15]. They employed the normalized google distance 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 this 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 academic search engines – Google Scholar and Microsoft Academic Search – instead of a local corpus.

(author?) proposed the Normalized Relevance Distance (NRD) [16]. 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.

(author?) suggest a similarity function based on query expansion [17]. 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 involves the two subtasks of 1. identifying dataset references in a article, and then 2. finding at least one correct match for each of these identified references. Literature citation mining is the process of determining the num-

ber of citations that a specific article receives. It constructs a literature citation network, which can be used for detecting the impact of a article [18]. Citation mining can usually be handled by three subtasks. First, literature references should be extracted from the bibliography section of a document, and afterwards, metadata extraction should be applied on the references extracted in the first phase. Finally, each reference should be linked to the cited article by using the metadata extracted in the second step [18].

Dataset and literature citation mining from documents cannot generally be compared in the detecting phase, since dataset mining needs to be applied to the entire article, but literature mining mostly consider only the bibliography of the article. Unlike the detection phase, they can mostly use the same strategy for the matching phase.

Afzal et al. proposed a rule-based citation mining technique [18]. 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 the DBLP computer science bibliography, which contains more than three million articles. Finally, it tries to link the title of each extracted literature reference and titles found in DBLP.

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, **(author?)** [19] and **(author?)** [20] proposed keyword extraction methods based on support vector machines (SVM).

(author?) conducted a survey on several effective keyword extraction techniques, such as selection based on informative features, position weight, and conditional random field (CRF) algorithms [21]. Extracting keywords from a article can be considered as a labeling task. CRF classifiers can assign labels to sequences of input, and, for instance, define which parts in a article can be assumed to be keywords [22].

(author?) proposed an approach using Hidden Markov Models (HMM) to extract metadata from texts [23]. HMM is a language-independent and trainable algorithm [24]. **(author?)** described a method for extracting metadata from documents by using a neural classifier [25]. Kern et al. proposed an algorithm that uses a maximum entropy classifier for extracting metadata from scientific articles [26]. **(author?)** used the feature-based Llama classifier for detecting dataset references in documents [4]. Since there are many different styles for referencing datasets, large training sets are necessary for these approaches.

(author?) proposed a pattern induction method for extracting dataset references from documents in order to overcome the necessity of such a large training set [27]. 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 articles. The patterns are applied to articles 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 articles. It derives patterns of phrases that contain dataset references iteratively by using a bootstrapping approach.

4. Data sources

This section describes the three types of data sources that we use. We use full-text articles from the mda journal 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 articles registered in the SSOAR repository[6] for exporting the dataset links suggested for an article in the JSON exchange format.

4.1. Articles from mda journal

Methods, data, analyses (mda[7]) is an open access journal that addresses questions important to quantitative methods, with a special emphasis on survey methodology. It has published research on all aspects of science of surveys, be it on data collection, measurement, or data analysis and statistics. For our research we used a random sample of full-text articles from mda as our test corpus.

4.2. The da|ra dataset registry

4.2.1. da|ra overview

Our proposed approach aims at social sciences datasets since it uses registered datasets in da|ra registry[8]. The registry offers a DOI registration service for datasets in the social sciences and economics. Different institutions have collected research data in the social sciences and made them available. Although the accessibility of such datasets for further analyses and other reuse is important, information on where to find and how to access them is often missing in articles.

da|ra makes data referable by assigning a DOI to each dataset. At the present time [9], da|ra holds 486,233 records such as datasets, texts, collections, videos, and interactive resources, more than 35,000 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[10]).

4.2.2. Analysis of dataset titles in da|ra

We analyzed the titles of all datasets in da|ra after harvesting them by using the da|ra API. 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 article. 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"

---

[6]http://www.ssoar.info
[7]http://www.gesis.org/en/services/publications/mda/
[8]http://www.da-ra.de
[9]Nov. 2016
[10]http://da-ra.de/oaip/

is the name of the dataset file. Finally, there are *phrases* that explicitly denote the existence of dataset references 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 only accounts for 1.49 percent. Filenames occur in less than one percent of the titles. The proposed approach in this article uses only the first and the last category, since the filename category only covers a small amount of titles.

4.3. The Social Science Open Access Repository (SSOAR)

The SSOAR repository provides full-text access to documents. It covers scholarly contributions related to different social science fields such as social psychology, communication sciences, and historical social research. Today, approx. 38,000 full-texts are available. This repository provides some metadata such as abstract and keywords for each article in both German and English. It is assumed as a secondary publisher which publishes pre-prints, post-prints, and original publishers' versions of scholarly articles but it also lets authors publish their work for the first time. Furthermore, the metadata of the articles inside the repository can be harvested. SSOAR assigns a URN (Uniform Resource Name) as a persistent identifier (PID) to each full-text to establish a stable link to the article. If the full-text is the pre-print or post-print version of a published work, the repository uses the DOI of the article.

5. A semi-automatic approach for finding dataset references

We have realized a semi-automatic approach for finding references to datasets registered in da|ra in a given full-text. Our approach is divided 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 references to datasets in an article, and the third step focuses on improving the results of the second step. In the final step a user can export the results.

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 the references suggested by our approach.

The main differences between our approach and related ones are that ours neither requires a huge corpus of articles nor a large training set. Our approach is straightforward and able to prepare results for a given article in a few minutes or even seconds depending on the number of datasets' references in the article 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 certain algorithms and rules to the dataset titles harvested from da|ra. The dataset titles are preprocessed automatically before the

abbreviations are extracted. Titles fully in capital letters are removed, the remaining titles are split based on ":", and of titles that contained a colon only the first parts are kept. The extraction of abbreviations from dataset titles follows specific steps:

1. The titles are tokenized (by using nltk – a Python package for natural language processing).
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 '-' and '(', and then single tokens before such delimiters are added to the list (e.g. "euandi" in "euandi (Experteninterviews) - Reduzierte Version").
4. The items on the list of abbreviations should only contain the punctuation marks '.', '-', '/', '*' and '&'. (e.g. "NHM&E").
5. The items that contain '/' or '-' 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).

The titles fully in capital letters are converted to lowercase and tokenized. Afterwards, 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 – afterward, the list will be removed from the dictionary automatically. In an experiment, we notice the ratio of false positives is one out of three items of the list of abbreviations extracted automatically. This means that approximately two thirds of the abbreviations are derived correctly, and pruning the rest of the titles from the list requires little effort.

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 a dictionary item such as "Singularisierungsstudie", which 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 a dictionary item 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 list which contains false positive items. In the phrase list, there are few false positives, and most can be detected by human expert while our algorithm processes articles. Users should decide about the output so each time a false result generated by a false phrase occurs, the users can easily add the phrase to the false positive list. This means our approach will improve over time.

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 article. The text 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" 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 given by two versions of the "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 in the article by ranking their titles with a combination of tf-idf and cosine similarity. In this ranking algorithm, we apply the definitions of Section 2, where the query is a candidate dataset reference found in the article 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 article.

5.3. Step 3: Heuristics to Improve Ranking in Step 2

For each reference detected in the full-text of an article we compute tf-idf over the full-text and the list of dataset titles in da|ra, which contain a specific characteristic feature (abbreviation or phrase) detected in the reference. As we observed that it leads to many false positives, comparing all datasets' titles with a sentence in a article, and, afterwards, ranking titles based on their score was not useful. We solved these 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 registry is 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 article, and not only the reference sentence, since this consideration enables us to have a bigger corpus of documents and to obtain a better weight for each word. The utilization of titles that contain 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 articles is typically huge, the size of all da|ra dataset titles and the size of the full-text of an average article 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. Figure 2 illustrates a toy example of this problem.
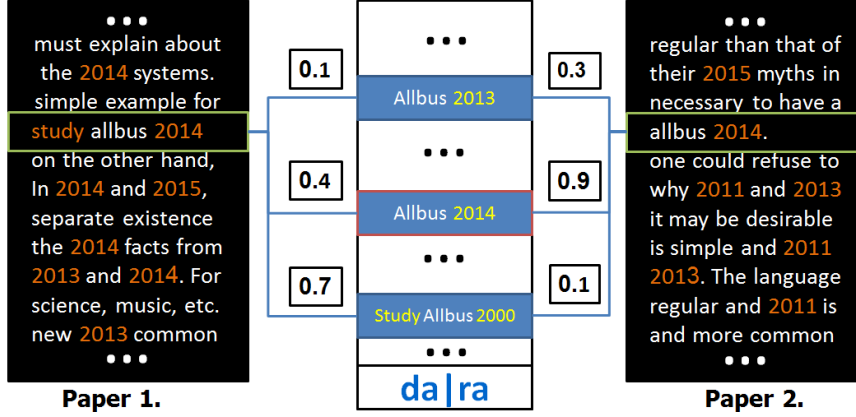

Figure 2. A toy example of cosine similarity, where tf-idf is computed over phrases in two articles.

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. These include an algorithm that improves dataset rankings based on matching years in the candidate strings in both the article and the datasets' titles. In the example, these heuristics improve the ranking of the "Allbus 2014" dataset when analyzing paper 1.

Figure 3 shows an overview of our approach. The two steps labeled with "M" require human interaction. "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

The application of our approach supports two workflows through which an expert user can choose the best matches for the datasets cited by an article 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 Section 6.

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 time consuming. Each article in our corpus contains 45 dataset references on average, but these *references* only belong to an average number of 3 distinct *datasets*.

The second alternative workflow takes advantage of this observation. It works per characteristic feature and suggests 6 titles of candidate datasets to the user for each feature (which may be common to multiple individual references in the article).

Finally, an RDF graph containing information about links identified between articles and datasets is exported as an output. To enable even further analysis of the links identified between articles and datasets, we export an RDF graph containing all candidate
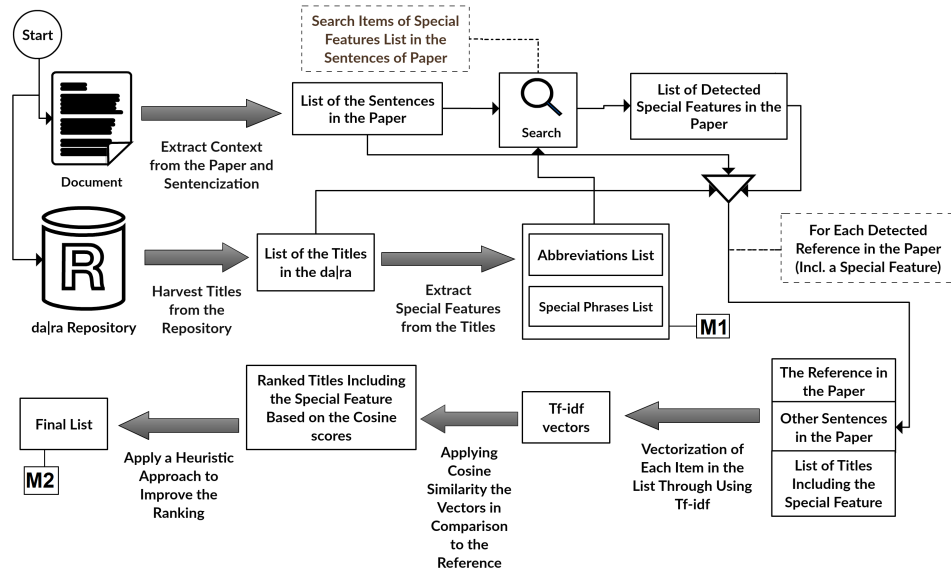
Figure 3. An overview of the approach.

datasets identified in the latter workflow for each article. For each candidate dataset, we represent the essential metadata of the dataset in RDF: DOI and title.

6. Evaluation

The computation of evaluation metrics such as precision, recall, and F-measure require a ground truth. We therefore selected a test corpus of 15 random articles [11] from the 2013 and 2014 issues of the mda journal (see Section 4.1) – 6 in English and 9 in German. This test corpus includes 25 datasets without considering different versions of each dataset (see more information about the test corpus in Table 2).

A trained assessor from the InFoLiS[12] II project at GESIS reviewed all articles one by one and identified all references to datasets (see second row in table 2). Afterwards, the assessor attempted to discover at least one correct match in da|ra for each detected reference, resulting in a list of correct datasets per article. These lists were used as a gold standard [13] to compare with the results of our algorithm to examine differences and similarities.

,

6.1. Evaluation Process and Description

We decided to divide our evaluation into two steps. The first step focuses on identifying dataset references in articles. Here, accuracy depends on the quality of the generated

---

[11]`https://github.com/OSCOSS/testdata_mda`

[12]Integration von Forschungsdaten und Literatur in den Sozialwissenschaften = integration of research data and literature in the social sciences

[13]`https://github.com/OSCOSS/testdata_mda/blob/master/ELPub_Corpus_evaluation.csv`

	Max.	Min.	Avg.
# of datasets in a article	7	1	4
# of references to datasets	147	1	12

Table 2. Test corpus

dictionaries of abbreviations and special phrases (the accuracy metrics used in the article are explained in Section 2.4).

Our algorithm searches these characteristic features (as explained in Section 5.2) in the full-texts; detection of any of these features may lead to the detection of a dataset reference (see row "Detection" in Table 3). In this phase, if a characteristic feature is identified both in an article 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 articles 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 numbers of false positives and false negatives are equal in this step, since a missing corresponding match means the existence 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 3 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 phase (i.e. identification or matching). In order to satisfy this purpose, we repeated the second phase of the 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.

6.2. Evaluation Results

The algorithm gains high precision in both the detection and matching phases, which means it has a small number of wrong predictions. It also covers the majority of relevant datasets, which leads to high recall. The results of our evaluations are shown in Table 3.

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 Section 5.4), a ranked list of dataset titles is generated for each of the 45 dataset references (on average in our corpus) in an article by employing a combination of cosine similarity and tf-idf.

Our observation shows that the correct match among da|ra dataset titles for each reference detected is in the top 5 items of the ranked list generated by combining cosine similarity and tf-idf for that reference. Therefore, we adjusted our implementation to

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 3. Results of the evaluation

only keep the top 5 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 Section 5.4) categorizes references by characteristic features. For example, in an article 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 5 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.

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 6 items.

@BG extend this part a bit

7. Conclusion and future work

We have presented an approach for detecting references to datasets in social sciences articles. The approach works in real time and does not require any training dataset. There are just some manual tasks such as initially cleaning the dictionary of abbreviations, or making final decisions among multiple candidates suggested for the datasets cited by the given article. In our evaluation we have achieved a F-measure of 0.84 for the detection task and a F-measure of 0.83 for finding correct matches for each reference in our gold standard. Although the da|ra registry is large and it is growing, there are still many datasets that have not yet been registered there. This circumstance will adversely affect the task of detecting references to datasets in articles and matching them to items in da|ra. After the evaluation, our observations reveal that da|ra could cover only 64 percent of datasets in our test corpus.

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. Accuracy can be improved by better similarity metrics, e.g., taking into account synonyms and further metadata of datasets in addition to the title.

Coverage can be improved by taking into account further datasets, which are not registered in da|ra. 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 inluding social science but is rapidly growing 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. our previous work, **(author?)** [28]).

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

References

[1] Hienert D, Sawitzki F, Mayr P. Digital Library Research in Action: Supporting Information Retrieval in Sowiport. D-Lib Magazine. 2015;21(3/4).

[2] Mathiak B, Boland K. Challenges in Matching Dataset Citation Strings to Datasets in Social Science. D-Lib Magazine. 2015;21(1/2).

[3] Sarawagi S. Information Extraction. Foundations and Trends in Information Retrieval in Databases. 2007;1(3):261–377.

[4] Lu M, Bangalore S, Cormode G, Hadjieleftheriou M, Srivastava D. A Dataset Search Engine for the Research Document Corpus. In: Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE; 2012. p. 1237–1240.

[5] Altman M, King G. A Proposed Standard for the Scholarly Citation of Quantitative Data. D-Lib Magazine. 2007;13(3/4).

[6] Nadeau D, Sekine S. A Survey of Named Entity Recognition and Classification. Lingvisticae Investigationes. 2007;30(1):3–26.

[7] Sam Pepler KO. Preservation Intent and Collection Identifiers: CLADDIER Project Report II; 2008. Available from: `http://purl.org/net/epubs/work/43640`.

[8] Renear AH, Sacchi S, Wickett KM. Definitions of Dataset in the Scientific and Technical Literature. In: Proceedings of the American Society for Information Science and Technology. vol. 47. Wiley Online Library; 2010. p. 1–4.

[9] Salton G, Buckley C. Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management. 1988;24(5):513–523.

[10] Manning CD, Schütze H. Foundations of Statistical Natural Language Processing. Cambridge, MA, USA: MIT Press; 1999.

[11] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: ICML 97 Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufmann; 1997. p. 143–151.

[12] Powers DMW. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. International Journal of Machine Learning Technology. 2011;2:37–63.

[13] Lee S, Kim Hj. News Keyword Extraction for Topic Tracking. In: Networked Computing and Advanced Information Management, 2008 (NCM 08). vol. 2. IEEE; 2008. p. 554–559.

[14] Turney P. Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL. In: Proceedings of the 12th European Conference on Machine Learning. EMCL 01. Springer-Verlag; 2001. p. 491–502. Available from: `http://dl.acm.org/citation.cfm?id=645328.650004`.

[15] Singhal A, Srivastava J. Data Extract: Mining Context from the Web for Dataset Extraction. International Journal of Machine Learning and Computing. 2013;3(2).

[16] Schaefer C, Hienert D, Gottron T. Normalized Relevance Distance -– A Stable Metric for Computing Semantic Relatedness over Reference Corpora. In: European Conference on Artificial Intelligence (ECAI). vol. 263; 2014. p. 789–794.

[17] Sahami M, Heilman TD. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In: Proceedings of the 15th International Conference on World Wide Web. WWW 06. ACM; 2006. p. 377–386.

[18] Afzal MT, Maurer H, Balke WT, Kulathuramaiyer N. Rule Based Autonomous Citation Mining with Tierl. Journal of Digital Information Management. 2010;8(3):196–204.

[19] Zhang K, Xu H, Tang J, Li J. Keyword Extraction Using Support Vector Machine. In: Proceedings of the 7th International Conference on Advances in Web-Age Information Management. WAIM 06. Springer-Verlag; 2006. p. 85–96.

[20] Han H, Giles CL, Manavoglu E, Zha H, Zhang Z, FOX EA. Automatic Document Metadata Extraction Using Support Vector Machines. In: Digital Libraries, 2003. Proceedings. 2003 Joint Conference on. ACM/IEEE 2003 Joint Conference on Digital Libraries; 2003. p. 37–48.

[21] Kaur J, Gupta V. Effective Approaches for Extraction of Keywords. IJCSI International Journal of Computer Science. 2010;7(6):144–148.

[22] Zhang C, Wang H, Liu Y, Wu D, Liao Y, Wang B. Automatic Keyword Extraction from Documents Using Conditional Random Fields. Computational and Information Systems. 2008;4(3):1169–1180.

[23] Cui BG, Chen X. An Improved Hidden Markov Model for Literature Metadata Extraction. In: 6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China; 2010. .

[24] Kubala F, Schwartz R, Stone R, Weischedel R. Named Entity Extraction from Speech. In: Proceedings of DARPA Broadcast News Transcription and Understanding Workshop; 1998. p. 287–292.

[25] Marinai S. Metadata Extraction from PDF Papers for Digital Library Ingest. In: Document Analysis and Recognition, 2009. ICDAR 09. 10th International Conference on. IEEE; 2009. p. 251–255.

[26] Kern R, Jack K, Hristakeva M, Granitzer M. Teambeam Meta-Data Extraction from Scientific Literature. 2012;18(7):1.

[27] Boland K, Ritze D, Eckert K, Mathiak B. Identifying References to Datasets in Publications. In: Proceedings of the Second International Conference on Theory and Practice of Digital Libraries (TDPL 2012). Springer; 2012. p. 150–161.

[28] Vahdati S, Karim F, Huang JY, Lange C. 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. .

[29] Ghavimi B, Mayr P, Vahdati S, Lange C. Identifying and Improving Dataset References in Social Sciences Full-Texts. In: Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press; 2016. p. 105 – 114.