Ulm University - Faculty of Medicine
Ulm University of Applied Sciences - Faculty of Computer Science

Developing an Automated Bibliometric Analysis System for Finding Rare Disease Experts

Dissertation to Obtain the Doctoral Degree of Human Biology (Dr.biol.hum.) of the Medical Faculty of Ulm University

Presented by Andreas Pflugrad
Born in Villingen-Schwenningen

2016

Acting Dean: Prof. Dr. Thomas Wirth

1st Reviewer: Prof. Dr. Jochen Bernauer

2nd Reviewer: Prof. Dr. Hans Kestler

Day of Graduation: 02.12.2016

This work is dedicated to Hildegard and Wolfgang Pflugrad

Contents

Abbreviations

AC Agglomerative Coefficient

ACS Acrocallosal syndrome

API Application Programming Interface

CDA Congenital dyserythropoietic anaemia

CRUD Create, Read, Update, Delete

DB Database

DBLP Digital Bibliography & Library Project

DBMS Database Management System

DBSCAN Density-Based Spatial Clustering of Applications with Noise

EDM Entity Data Model

EF Entity Framework

ETL Extract, Transform, Load

GUI Graphical User Interface

HypoPP Hypokalaemic Periodic Paralysis

IR Information Retrieval

ISSN International Standard Serial Number

L2E LINQ to Entities

LINQ Language Integrated Query

MeDRA Medical Dictionary for Regulatory Activities

MeSH Medical Subject Headings

MVC Model-View-Controller

NASA National Aeronautics and Space Administration

NCBI National Center for Biotechnology Information

NEMO Normalization Engine for Matching Organizations

NER Named Entity Recognition

NIH National Institutes of Health

NLM U.S. National Library of Medicine

NORD National Organization for Rare Disorders

ORM Object-Relational Mapping

PMID PubMed unique Identifier

RegEx Regular Expression

SOAP (originally: Simple Object Access Protocol)

SQL Structured Query Language

TF-IDF Term Frequency - Inverse Document Frequency

TOS Talend Open Studio

UID Unique Identifier

UN/LOCODE United Nations Code for Trade and Transport Locations

URL Uniform Resource Locator

XML Extensible Markup Language

1 Introduction

Patients suffering from a rare disease often face a condition that is life-threatening or chronically debilitating [26]. These diseases, which often already affect children or adolescents, can, to a great extent, not be cured and are difficult to treat. Rare diseases pose a significant challenge to not only the patients, their families and immediate health care providers [85], but to health care authorities in general due to what can be called the rare diseases paradox:

> *"Rare diseases are rare, but rare disease patients are numerous"* [10]

A disease is considered rare by European definition if no more than five in 10.000 people are affected [79]. Due to the large number of rare diseases, estimations indicate that 3-6% of the population may be affected, pertaining to anywhere between 2,4 million and 5 million patients in Germany and between 13,5 and 25 million patients all across Europe [26, 85]. When adding the numbers from the USA with additional 30 million patients [15], one can imagine that rare diseases affect a huge number of people worldwide. The number of described rare diseases is steadily increasing. While reports from 2007 mention about 4.700 diseases [26], the number increased to around 6.000 in 2012 [61] and as of 2015, a large European rare diseases database comprises 6.368 singular diseases, syndromes and anomalies. This number extends to over 9.000 when considering all groups of diseases or phenomes and their subtypes [2, p.18]. In the USA, where a disorder is considered rare if it affects no more than 200.000 people, the Genetic and Rare Diseases Information Center of the National Institutes of Health (NIH) also lists ca. 7.000 diseases [9]. Thus, rare diseases make up one quarter of all known disease categories worldwide [1].

The rarity of the individual diseases entails a number of problems and challenges. Patients often face an odyssey of many years, visiting countless doctors and clinics before being correctly diagnosed in the first place. Even then, for the vast majority of diseases adequate medical treatment options or effective causal therapies have not been researched yet. Guidelines, if available at all, are based on little experience from few exemplary cases [84]. The disorders are often very complex, affecting multiple body systems. Patients therefore require interdisciplinary specialised care, which, however, is not always available close to their personal residences. Many rare diseases are only

researched by a small number of researchers, yet, due to their complex nature, interdisciplinary research is required as well [85]. As a consequence, patient information and patient empowerment is regarded a necessity, especially for rare disease patients according to Aymé et al. [25]. Yet, they appear to have been neglected factors in healthcare [32] and have only recently grown in attention.

In essence, it can be stated that the availability of information on rare diseases is a key element in improving the situation of both patients and health care providers. One important part of this is information on rare disease experts i.e. the few specialists for each disease who have experience in diagnosing and treating rare conditions. Not only are these experts valuable for providing special patient care, they are essential to interdisciplinary research endeavours and can also serve as competent contact persons for reviewing and evaluating rare diseases information. Thus, finding experts is important for multiple interest groups. Patients and their relatives are looking for people who can help them with diagnosis or treatment. Health care providers are looking for experts to refer patients to if they themselves cannot help. Research experts may be looking for partners to join efforts and rare disease information curators are looking for experts to validate their data.

1.1 Rare diseases information sources

A lot has been done in recent years to improve the availability of information and various information pathways such as online portals have been brought into life. Among the different types are general portals which provide comprehensive information on a broad variety of rare disease topics. The most prominent of these portals in Europe is arguably *Orphanet* [10], which operates a large database of rare diseases information. These include reviewed descriptions, epidemiological data and an according classification as well as various directories regarding expert centres, diagnostic tests, patient organisations and orphan drugs i.e. drugs that are specifically developed for the treatment of rare diseases. The US equivalent to Orphanet is *NORD*, the National Organization for Rare Disorders, which has a similarly broad information spectrum, yet has a significantly smaller database of about 1.300 diseases [16]. For Germany, the *Allianz Chronischer Seltener Erkrankungen (ACHSE)* provides patient-oriented disease descriptions, database references and patient reports [12].

Other portals primarily advocate patient empowerment and self-help. Two examples of such portals are *EURORDIS* [13] and their *RareConnect* initiative [17]. The former is an alliance of patient organisations from 63 countries dedicated to raising rare diseases awareness and building a pan-European community. The latter aims at directly interconnecting patients by providing communication possibilities such as discussion groups for sharing patient stories and experience reports but also for sharing information on medical experts and other helpful resources.

A few portals are specifically dedicated to people searching for experts or expert clinics. In Germany, the *se-atlas* project [18] offers a cartographic tool for searching medical institutions and self-help groups which care for rare disease patients, based on data from Orphanet as well as self-registration. *Expertscape [14]* is a portal for searching medical experts on certain diseases based on their scientific publication activity. The portal retrieves its data from PubMed [20], a worldwide database of biomedical literature. Although Expertscape is not specifically targeted at rare diseases, it can be utilised to find rare disease experts to some extent. Also, Orphanet's database includes expert clinics and reference centres from up to 35 European countries. While these existing information offers already provide some possibility for finding expert contact persons or institutions, a number of issues can be identified.

Most of the aforementioned portals rely on manual surveys, third-party recommendations or self-registration when it comes to gathering information about who and where rare disease experts are. At Orphanet, an encyclopedia team is responsible for maintaining data on rare diseases, their classification and all connected information including expert centres by manually surveying, annotating and validating documented sources such as scientific literature [2, p.13]. Additionally, expert advice may be received e.g. from questionnaires sent to clinics which, however, yield very sparse responses [67]. Data from Orphanet about expert-institutions have also been used in kick-starting the *se-atlas* project, which is now supplemented with self-registration data i.e. institutions can manually enter their expertise regarding rare diseases. This process can, however, prove quite difficult for institutions with expertise on a bigger number of disorders. Expertise can be specified on each level within the Orphanet hierarchy from very broad disease categories down to very specific disease entities. Placing entries for every specific disease entity may become very tedious and some disease entities might be missed in the process. However, placing the institution's expertise at a

broad category level may introduce an additional factor of inaccuracy or even bias as institutions might want to be recognised at a high level, yet their expertise might not actually cover all specific disorder entities of this category.

Further issues with manual surveys and self-registration are completeness and currency of expert-information which is also stated by Liu et al. [54]. With the sheer number of rare diseases, it is unlikely that manual processes can identify all relevant experts for every disease and with the low response rate for self-registration efforts there are likely to be many experts who have not yet been identified. The same problems affect the tracking of changes in an institution's expertise which would need to be actively communicated by the institution towards the information portals or otherwise have to be detected during the next manual update. Another point brought up by Liu et al. is that it is difficult to describe the competence of experts in a way that allows users to assess and compare their expertise.

Automated screening methods mitigate some of these problems by being able to continuously update their information on a large scale. Therefore, *Expertscape* can be considered a valuable step in this direction, it however suffers from a lack of specificity when it comes to rare diseases. Looking for diseases in *Expertscape* is based on the *Medical Subject Headings* (MeSH) vocabulary which works well in the context of common diseases but does not include the complete pool of rare disease terms.

In summary, it can be stated that current approaches for gathering and maintaining information about rare diseases experts may be improved in their completeness, up-to-dateness and specificity by an automated expert finder system which is specifically targeted at rare diseases.

1.2 Expert finding and bibliometric analysis

1.2.1 Approaches to expert finding

Stankovic et al. [76] define expert finder systems as

> *"Information Retrieval (IR) systems which identify candidate experts and rank them with respect to their estimated expertise on a given topic, using available evidence (e.g. documents about/of candidates, social networks of candidates, activities of candidates in real world and online)."*

This definition includes two parts: expert identification i.e. finding and locating people who have expertise in a given topic and expert selection or expert recommendation i.e. ranking available experts against each other and presenting those who are most appropriate to a certain query. McDonald and Ackerman [57] see these two parts as separate tasks. They further distinguish expertise from experts whereby expertise is considered a range of procedural or topical knowledge, skills and experience whereas the expert is an individual with possibly different aspects of expertise across various areas.

Research into computer-assisted expert finding can be traced back well into to the 1990s where McDonald and Ackerman conducted a field study of how people in a software company find colleagues with specific expertise for solving problems. They also outline a number of expert finding systems which had been developed by that time. Among those, there were already some systems which use content analysis and filtering techniques for different purposes such as matchmaking, finding people of similar interests and finding research experts via co-authorship relations. However, the authors conclude that the majority of expert finding in reality is done via social interaction and so-called expert-concierges i.e. people who know the experts within an organisation from experience [57]. Craswell et al. [35] created a system which uses the documents published on the intranet of an organisation to identify experts in different areas. Alani et al. [22] created a network of people, documents, conferences and projects within an enclosed academic computer-science domain. They performed an ontology-based analysis of this network to identify communities of practice rather than single experts. Crowder et al. [36] deployed software agents which scan common resources such as technical report repositories, e-mails and phone books to identify and recommend experts within a distributed engineering organisation. While their system is aimed at accelerating the connection making process between people, they do not see the approach as a replacement of social interaction for expert finding.

Becerra-Fernandez [27] presents a 2006 contemporary review, comparing "expertise locator systems" used in different large companies and organisations such as Hewlett Packard, Microsoft and NASA for project staffing, knowledge sharing and consultation. These systems are mostly based on manual assessments either by the experts themselves or their peers. Other systems are mentioned which rely on analysing technical documents, corporate communications and discussion groups. One featured system

uses a funded research database to find experts within a university network in Florida. Ehrlich et al. [41] incorporate social network analysis into their system for finding experts within an enterprise.

Several expert finding systems base their approach on collections of authored documents e.g. scientific publication databases. Tang et al. [78] created *ArnetMiner*, an information retrieval tool which uses the *Digital Bibliography & Library Project* (DBLP) computer science literature database along with documents crawled from the open web for expertise-oriented search. Deng et al. [39] also use DBLP as data source and include Google Scholar for data supplementation. Afzal et al. [21] perform expert discovery within a specific computer science journal while Liu et al. [54] build their expert repository from two regional publication databases. Expertscape, which has been presented in section 1.1, also falls into this category.

Following the emergence of the Web 2.0, where due to blogs, forums and wikis, people were no longer just mainly information consumers but active information producers, a number of research projects were examining the usage of these new data sources for expert finding. These include Chua [33], Zhang et al.[90], Demartini [38] and Amitay et al. [24]. More recent research looked at question answering communities such as Omidvar et al. [58] and Zhao et al. [91]. Along with this development, expert finding shifted away from mainly looking for expertise in an enterprise or organisational setting towards a much broader i.e. worldwide scope. Another step towards this is the utilisation of the Semantic Web. Aleman-Meza et al. [23] proposed the combination of multiple Resource Description Framework vocabularies for the use in expert finding. Both Stankovic et al. [76] and Latif et al. [52] researched the use of Linked Open Data for finding and recommending experts. More recent research has continued this trend by applying a variety of web mining techniques for expert finding. These include Wu et al. [89], Guan et al. [44] as well as Vergne and Susi [82].

In summary, expert finding systems have grown from enterprise-individual solutions to vast data mining efforts. Their data sources range from internal corporate communication, scientific literature databases, forums and question answering communities up to crawling the entire world wide web as well as using the semantic web. Besides identifying experts, a variety of mathematical models are applied for expert ranking and recommendation. Of these different approaches, the bibliometric approach i.e. identifying experts based on their publication activity has been further elucidated for

this work.

1.2.2 Bibliometric analyses and expert identification

The underlying principle of identifying experts from literature is the premise that if a person has authored scientific publications on a certain topic, this person might be considered an expert on that topic. This premise is stated e.g. by Stankovic et al. [76] as well as Tang et al. [78]. According to Tang et al. [78, p.6], the number of publications could be seen as an indicator of expertise. However, as is stated by Mattern [56], no one single indicator is able to accurately estimate the quality of publications or researchers. Instead, literature data provides a number of additional information that have to be considered with respect to expertise. Among these are the type of publications where authoring a review or guideline might be indicative of a high degree of expertise or the recency of publications where newer research may be of particular interest. Finally, experts are not only to be identified but also contacted. Therefore, information ranging from an author's location i.e. country, city and institution to direct contact possibilities like e-mail addresses constitute potentially useful data. This kind of data might be obtained from the so-called *Affiliation* entry present in many publications.

One additional factor to how an author's expertise is perceived could be the position in which he or she appears in the author list. Wren et al. [88] present a survey of the perceived contribution of authors to a paper based on their position. Especially the first and last positions may be of interest as they appear to receive the most credit. However, Wren et al. also conclude that the actual contribution can greatly differ.

Drawbacks of current bibliometric approaches Many of the presented bibliometric expert finding approaches are situated outside of the biomedical domain. The ones which address biomedical publications, e.g. via PubMed, also suffer from a number of drawbacks. *PubReMiner* [49] aggregates results from searches in PubMed e.g. for a specific disorder into frequency tables showing various meta-information and their interconnection including authors, journals, keywords and countries. These information can be used to gain an overview of potential experts on a specific subject as is also stated by Slater [74]. However, the number of results, that can be processed, as well as the detail to which the information is broken down is restricted and there is no

possibility for an automatic access of the tool, rendering it unusable beyond individual manual checks.

Expertscape [14] for the most part uses the same approach as PubReMiner but within the specific context of finding experts for diseases. As such, it offers additional information including the institutions that the authors have been affiliated with, their position in the respective publications and provided suggestions for getting yet more information from a custom Google search. Besides the issue of not including a lot of rare diseases due to using the MeSH vocabulary, the results that can be obtained from Expertscape also suffer from author name ambiguity.

1.2.3 Author name disambiguation

When using literature data to find experts, it is important to bring together all publications of the same person. However, since author names are only available as raw text and publication databases hardly provide a unique author identifier, two major issues arise that can skew person-centric data: (1) With multiple different authors sharing a common name, disambiguation is required in order to not merge all their publications into one bag and treat the authors as a single person, thus falsely overstating their publication activity. While this is, in many cases, hard enough to do for a human annotator, it poses a significant challenge to perform disambiguation in an automated manner on a large-scale dataset. (2) Additionally, there may be cases where one author's name is spelled differently across multiple publications, e.g. due to typographical errors, special characters or double-surnames. In these cases, their publications need to be merged together despite different spelling in order to not falsely split publication data and thus understating an author's publication activity.

Approaches to name disambiguation A lot of research has gone into name disambiguation, yet a universally usable automated approach appears to be still unavailable, with many attempts involving very specific conditions, manual interaction or individual adjustments. However, most disambiguation processes can be summarised into three steps which are illustrated by Huang et al. [47]: name blocking, similarity calculation and clustering. In the first step, only those names that require disambiguation are grouped into blocks to avoid comparing completely incompatible names in the first place. Grouping is mostly based on exact matches or some kind of name similarity

measure. Subsequent steps only take place in each of these blocks which greatly reduces the computational complexity. The second step involves a pairwise computation of either similarity or distance. This computation is based on certain features that can be derived from the publication data. Finally, the similarities or distances are used to cluster the instances of each block, whereby the entries i.e. publications within each resulting cluster are considered to belong to a single person. While this appears to be the most common pattern, many variations in each step are recorded in the literature.

Han et al. [45] introduced an approach to disambiguate author names from the DBLP library using spectral clustering on three publication features: co-author names, paper titles and publication venues i.e. journals. Huang et al. [47] use a supervised approach to similarity calculation and performed clustering using the DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) algorithm. Later, with Song et al. [75], they use unsupervised similarities and agglomerative clustering, also factoring in the topic distribution of each author. In their work, they state that selecting the right features to base the similarity computation on is far more important than the choice of the clustering algorithm.

Torvik and Smalheiser [80] explicitly perform disambiguation on PubMed data based on a previously developed model [81]. After a name-blocking step that accounts for exact matches between last names and the first name initial, their pairwise similarity profiles are based on 10 factors. Among these are commonalities between first names, title words, MeSH words, co-authors, affiliations and language. Their work also uses probability-based agglomerative clustering and applies some more corrective measures afterwards. A special characteristic is the availability of its results until 2009 for research purposes.

A broader review of various disambiguation efforts up to 2010 is provided by Elliot [42]. There are later notable approaches such as Shu et al. [73]. Wang et al. [83] include some manual user interaction for verifying or correcting their disambiguation results as they regard automatic name disambiguation to be insufficient. Tang et al. [77] propose a unified probabilistic disambiguation framework which also underlies their *ArnetMiner* system. Liu et al. [53] put special emphasis on the varying requirements for name similarity matching across different e.g. Asian name cultures.

Lastly, with Liu et al. [55], the National Center for Biotechnology Information (NCBI) presented a PubMed-specific in-house author name disambiguation process

which, in its basic methodology, resembles the approach by Torvik and Smalheiser [80]. They use supervised machine learning based on large training data sets for similarity computation based on a total of 9 differently weighted PubMed data fields. Even though they claim to have achieved very good results, similar to [80], their system is unfortunately not available for use by third-parties.

1.3 Goals of this work

After examining all approaches to expert finding and their associated prerequisites and issues, it was the goal of this work to address the following questions:

> *Is it possible to conceptualise and implement an expert finder system on the basis of bibliometric analyses that satisfies information demands for rare disease experts and can be used for complementing existing expert registries? Can such a system surmount the drawbacks of similar approaches by being tailored to articles on rare diseases, by automatic in-depth analysis of publication data and by employing author name disambiguation?*

With respect to these questions, a number of more detailed goals can be stated for this work:

Literature search Finding and retrieving the right literature is the foundation of any further analysis. This task requires a comprehensive thesaurus of disease terms to be used for searching literature as well as a search strategy which will, ideally, find all relevant publications on rare diseases.

Literature extraction management The retrieval of publication data needs to take place periodically and on a large scale. Therefore, it was the goal build an appropriate management application which can perform the extraction in a flexible manner.

Data processing The extracted data may inherently contain syntactic and possibly semantic inconsistencies. Therefore, data processing needs to involve the application of standard *ETL (Extract, Transform, Load)* tasks with an emphasis on data cleaning. Since the optimal data structures for initial storage and final representation are different from one another, data transformations need to take place as well.

In-depth analysis In some cases, information that is relevant for expertise estimation might not be directly available from the retrieved literature and instead has to be extracted and integrated from unstructured data or from additional third-party information sources. Therefore, suitable analysis algorithms are required here.

Name disambiguation In order to mitigate the negative effects of author name ambiguity as outlined in section 1.2.3, a disambiguation technique has to be employed. This involves the implementation of multiple steps depending on which of the available approaches is being followed.

Expert profiles For each author, two variants of detailed expert profiles were envisioned which would allow users to assess an author's expertise. The first variant would only include data from publications on a specific disorder since that might be the primary interest of the user. The second variant includes data from all publications of the author thus providing a more comprehensive overview of the author's research, publication activities and expertise. From the expert profile, the user should get a detailed list of all publications which constitute the profile along with a timeline that reflects the author's publication history. Further details should include the list of journals, affiliations and an overview of all keywords associated with the author.

End user client The expert profiles need to be made accessible to end-users. It was intended to provide a web interface that starts with choosing the disease for which the profiles should be displayed. As with the search term thesaurus for retrieving literature data, the Orphanet classification of rare diseases is intended for this purpose, however on a more comprehensive scale, containing all superordinate terms as well as acronyms in all available languages. In order to allow users to navigate through this polyhierarchy, a tree-like display structure was aimed for.

Once the disease is selected, the user is to be presented with a list of authors who published on this disease. These lists should include all necessary information to provide the user with an overview of potential experts and their location. Additionally, users should be able to narrow down the lists according to self-selected criteria. For the latter, appropriate sorting and filtering options were to be implemented.

Evaluation Finally, an evaluation is to be carried out in order to be able to judge whether it is indeed possible to use the system for identifying both new experts as well as those that are already known otherwise and for estimating their expertise. This involves accessing information sources on known experts as well as finding partners for validating newly found potential experts.

2 Materials and Methods

A number of bibliometric approaches have already been outlined in section 1.2.1: Afzal et al. [21], Deng et al. [39], Liu et al. [54] and Tang et al. [78] all base their expert finding mainly on scientific publications whereby their data sources are, for the most part, restricted to computer science literature. Their overall approach, however, can be adopted and the methodology can be divided into four parts: *extraction*, *processing*, *deployment* and *evaluation*.

Data extraction in this work starts with the analysis of *PubMed* as the primary data source and the development of strategies for finding and retrieving rare diseases literature. Subsequent processing steps range from data cleaning tasks to data segmentation and disambiguation. The deployment stage involves the development of an end-user application and the required preliminary data transformations. Finally, the system's performance in finding rare diseases experts is evaluated on the basis of available expert data and compared to existing information sources. The entire process is illustrated in figure 1.

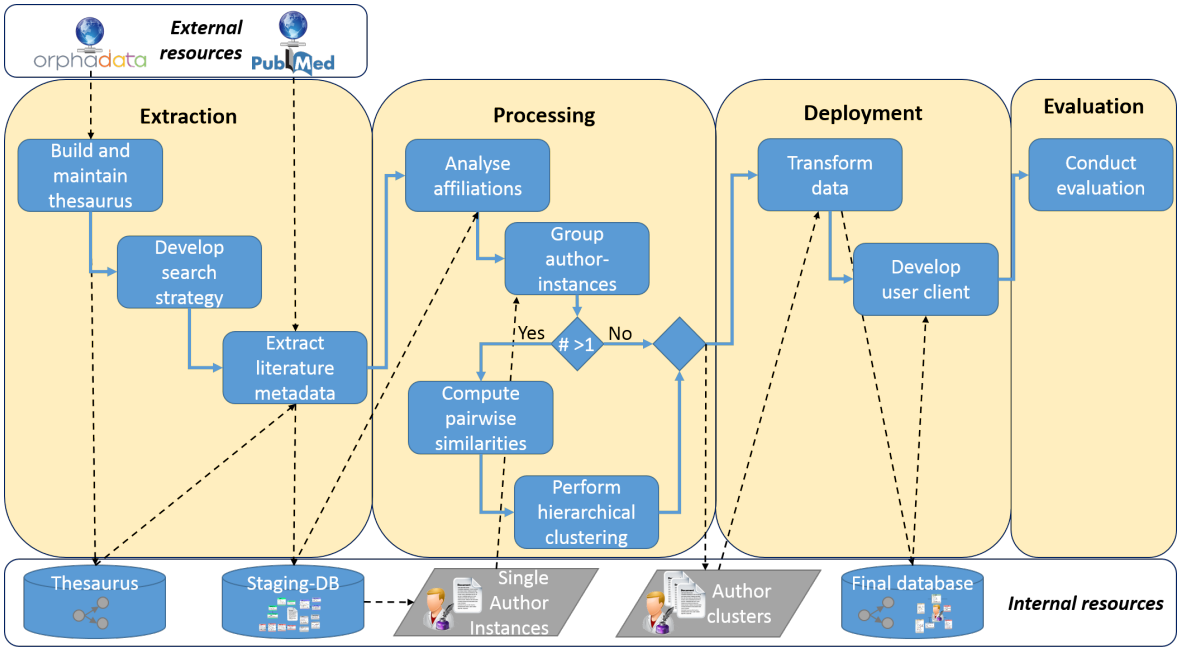

Figure 1: *Diagram of the system's overall workflow.*

2.1 Data extraction from PubMed

In order to base a rare diseases expert finder system on scientific publications, it is necessary to have a comprehensive source for biomedical literature. For this purpose, the *PubMed* [20] database is a primary candidate as it comprises in excess of 26 million citations from biomedical literature. Its sources include MEDLINE, *GeneReviews*, the *Cochrane Database of Systematic Reviews* as well as other online books and life science journals as of May 2016. Given this vast number of literature data, the majority of publications on rare diseases is likely to be found there. Another advantage is the availability of an Application Programming Interface (API) which allows for automated access.

The first step of preparing data extraction was to build a thesaurus of rare disease terms to be used in searching for literature. A search strategy needed to be devised in order to retrieve the right literature from PubMed for each disease i.e. publications which address the disease while keeping the amount of unspecific literature i.e. false positives low. In the final step, literature extraction actually needs to take place using the terms from the thesaurus and the devised search strategy. An appropriate management application to perform these steps as well as a staging database in which the extracted data can be stored needed to be developed.

2.1.1 Building a rare diseases thesaurus

In order to successfully query PubMed for literature on a wide variety of rare diseases, it is necessary to have a controlled vocabulary of disease names that can be utilised for searches. This vocabulary should cover all known rare diseases as well as all commonly used synonyms for each disease. It was decided to use Orphanet's data on rare diseases, which are available via *Orphadata* [3], as the primary base for setting up a rare diseases thesaurus as it should contain most, if not all, diseases which are classified as rare by European standards. It features the *Orpha-number* as a unique identifier and already contains a large amount of disorder terms. In order to build a comprehensive disorder term vocabulary on which PubMed searches can be based, terms from other terminologies were added.

An integrated database was set up starting with importing disease names from Orphadata. Several other terminologies were examined in order to find any comple-

mentary disease names. It was found that MeSH references as well the *Medical Dictionary for Regulatory Activities* (MeDRA) contained additional useful designations. An additional concept "Entered by User" was introduced for entries which may be added later in accordance with rare disease experts. The data in this resulting thesaurus is outlined in section 3.1.1.

2.1.2 PubMed API and article structure

PubMed allows the extraction of literature metadata via the *Entrez Programming Utilities* (*E-utilities*) API [63]. E-utilities comprises a set of server-side tools for querying the NCBI databases including queries for searching and downloading biomedical article data, document summaries or database statistics, retrieving gene records and protein sequences as well as finding cross-links between databases. Access to the Entrez system is realised with specific URL (*Uniform Resource Locator*) requests which can be parametrised to perform the desired query. In the beginning of this project, a SOAP web service endpoint was available and in use for accessing the Entrez system until being revoked in 2015 [64]. Since the entire data collection process took place within the timespan between starting this project and the endpoint being closed down, the successive implementation descriptions will be in reference to the SOAP service. While each of the nine different tools in the Entrez system can be used singularly, a typical usage scenario usually involves multiple tools used in conjunction to form a so-called data pipeline. In order to retrieve article data from PubMed, the basic ESearch → EFetch pipeline is used, where first articles on a specific topic are searched and the resulting articles are then fetched for data extraction.

ESearch ESearch in this case expects a text query, much like when performing a manual search on the PubMed website and returns a list of matching unique identifiers (UID) i.e. the PubMed unique Identifiers (PMID). This list could be extracted and used for retrieving the respective documents, however, this method would lead to additional complexity and an unnecessary increase in data traffic, especially with some of the more common rare diseases whose resulting lists can contain several thousand IDs. Instead, the search result can automatically be stored on the Entrez history server. In this case, two additional parameters, a Web Environment string (WebEnv) and a Query Key are returned by ESearch. Subsequent E-utility calls will use these parameters to access

Listing 1: *Example URL for querying PubMed via ESearch*

```
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=
   pubmed&term=Achondroplasia[Title/Abstract] OR Achondroplasia[
   MeSH Terms] OR Achondroplasia[Other Term] &usehistory=y
```

Listing 2: *ESearch response in XML format*

```
<eSearchResult>
    <Count>2484</Count>
    ...
    <QueryKey>1</QueryKey>
    <WebEnv>
        NCID_1_147815203_165.112.9.37
           _9001_1458321958_731062487_0MetA0_S_MegaStore_F_1
    </WebEnv>
    <IdList>
        <Id>26944588</Id>
        <Id>26899672</Id>
        ...
    </IdList>
    ...
    <QueryTranslation>
        Achondroplasia[Title/Abstract] OR "achondroplasia"[MeSH Terms]
            OR Achondroplasia[Other Term]
    </QueryTranslation>
</eSearchResult>
```

the history server and retrieve the search results from there. A number of optional parameters can be added to the query in order to restrict the search results e.g. to a specific time span. Listing 1 displays an exemplary ESearch query for Achondroplasia in the PubMed database. The term usehistory=y specifies the use of the history server.

The relevant parts of the answer for this query are depicted in Listing 2. *<Count>* states the number of articles that were found, *<IdList>* contains the PMIDs of the articles and *<QueryTranslation>* shows how the query has been handled internally by PubMed.

EFetch EFetch is used to retrieve data records from the NCBI databases. When querying PubMed, it is possible to fetch PMID lists, abstracts or full MEDLINE records. The relevant metadata for this project is contained in the MEDLINE record of each

Listing 3: *Example URL for querying PubMed via EFetch*

```
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi/?db=
   pubmed&query_key=1&WebEnv=NCID_1_147815203_165.112.9.37
   _9001_1458321958_731062487_0MetA0_S_MegaStore_F_1&retmode=xml
```

article. Listing 3 shows the URL request for fetching data records that correspond to the previous ESearch result. When using EFetch within the ESearch → EFetch pipeline, it accepts the WebEnv and Query Key from the ESearch result. Alternatively, a UID list could be provided. As with ESearch there are a number of optional parameters that can be used to further specify the fetch query. Here, retmode=xml has been used in order to receive the response in the *Extensible Markup Language* (XML) format showcased in Listing 4. When using the SOAP-based web service, all parameters are set programmatically and executed the same way.

Article structure The EFetch response is a set of articles where each article comprises a *<MedlineCitation>* element and a *<PubmedData>* element. The latter contains historical information about status changes of the article within the PubMed data system and is of no further relevance. The former contains, among others, the article's unique identifier in the *<PMID>* element as well as an *<Article>* and a *<MeshHeadingList>* element. The *<MedlineCitation>*'s *Status* attribute is used to depict the publication status of the article. The *<Article>* element contains a *<Journal>* element where all relevant journal information including ISSN (International Standard Serial Number), title, abbreviation and publication year are listed. *<Article>* also includes the *<ArticleTitle>* and *<Language>* elements as well as the *<PublicationTypeList>*. Finally, it contains the arguably most important field: the *<AuthorList>*. Each *<Author>* element in the list states the author's *<LastName>*, *<ForeName>* and *<Initials>*. The forename is not always available, especially from older publications because MEDLINE did not record first names before 2002 [80]. In this case, the *<ForeName>* field only contains the initial instead. This complicates the correct assignment of publications to their respective authors, especially if multiple authors share the same name. Since 2014, each *<Author>* element also contains the *<Affiliation>* element, detailing each author's corresponding institution which previously had only

Listing 4: *EFetch response in XML format*

```
<PubmedArticleSet>
    <PubmedArticle>
        <MedlineCitation Status="...">
            <PMID ...>26944588</PMID>
            ...
            <Article ...>
                <Journal>...</Journal>
                <ArticleTitle>...</ArticleTitle>
                ...
                <AuthorList ...>...</AuthorList>
                <Language>...</Language>
                <PublicationTypeList>...</PublicationTypeList>
            </Article>
            ...
            <MeshHeadingList>...</MeshHeadingList>
        </MedlineCitation>
        <PubmedData>...</PubmedData>
        ...
    </PubmedArticle>
</PubmedArticleSet>
```

been available for the first author [11].

Lastly, the *<MeshHeadingList>* contains all MeSH headings that have been attributed to the publication by the National Library of Medicine (NLM). The MeSH headings are split into descriptors and qualifiers. Descriptors are used to categorise the article and can range from age groups, body systems and organs, drugs and medical conditions to geographic locations. Qualifiers can be associated with descriptors to further specify particular aspects of certain subjects, e.g. therapy, diagnosis or drug effects. Descriptors can be designated as major topic to emphasise the focus of a publication. In the same way, one or more qualifiers can be designated as major topic where a descriptor is associated with multiple qualifiers. Even though all of the described data elements have been chosen for extraction, they only comprise a subset of the available data. A more comprehensive overview is given in [19].

2.1.3 Finding a search strategy for PubMed

Searches in PubMed can be varied by the choice of search words, the choice of the fields that these words are searched in as well as how these expressions are logically combined

into a so-called search strategy. Choosing the right strategy can greatly influence the amount and quality of the results that are returned by a PubMed query and this topic has been addressed in numerous research papers. However, most investigations on finding optimal search strategies focus on identifying specific types of publications such as different studies [86], reviews [87], diagnostic tests [40] or adverse effect reports [43]. They rarely touch on disorder-related topics with one exception being Schaafsma et al. [65]. Additionally, these research efforts rely mostly on varying their choice of search terms in order to receive an optimal result, an option which could not be utilised in this project as the search terms are predetermined by the contents of the thesaurus. Finding a valid search strategy has therefore been mostly about choosing the right search fields.

A total of 51 different search fields can be used in a PubMed query and have to be chosen by the user depending on what is to be searched. An overview of all possibilities can be obtained at [30]. With respect to finding publications on a specific topic, similar search fields have been used in the literature, with the *Title* and *Abstract* fields being most prominent. Additionally, MeSH-related fields have been used, e.g. the *Subheadings* field in [40] or [43]. The same and similar fields were examined for obtaining good search results on rare disorder publications in this work. One more possibility of influencing search results is PubMed's automatic term mapping [29]. Usually, search terms sent to PubMed are matched against several translation tables. If e.g. MeSH terms can be identified in the query, the search is automatically expanded to the main MeSH term as well as any direct subordinates which generally leads to a broader search with more results. This can be disallowed by enclosing the search terms in double quotes. In this case, PubMed will search for the text exactly as provided which may lead to very specific but also very few results.

Initial exploratory PubMed searches without a specific strategy suffered from a large amount of obvious false positive results for many disorders. A search strategy had to be found which would mitigate this issue but at the same time would not result in a lack of search results for many other disorders. Therefore, starting from a very strict approach, increasingly less stringent strategies have been applied and examined. The first two of the examined search strategies include the Title (TI) and MeSH Major Topic (MAJR) as a minimal set of search fields. While in the first strategy, the query is carried out as a strict search i.e. automatic term mapping by PubMed is disallowed, the

second strategy constitutes a soft search which allows term mapping. A third strategy adds the Abstract field to be searched together with the Title (TIAB). Finally, the MeSH Major Topic restriction has been opened up to include all MeSH Terms (MH), regardless of whether or not they are marked as major topic as well as the Other Terms (OT) field which contains keywords provided by the authors.

For each possible strategy, search runs have been conducted based on the first 100 Orpha-numbers. In order to choose the strategy to be employed in a full-scale data collection, the amount of returned articles has been compared across all possibilities while sample-based manual checks verified the specificity of the results as far as possible.

The results for the different strategies as well as additionally discovered issues are presented in section 3.1.2. In general, the examination showed that disease names from other languages such as German did for the most part not return any results which is why only English designations were used for searching literature. Due to the issues with acronyms and problematic disorder names as described in section 3.1.2, false positives were excluded by disallowing automatic term mapping where applicable and acronyms of disease names were removed from the search term thesaurus. Finally, in combination with these additional cleaning steps, the least strict strategy was used for literature extraction i.e. terms are searched for as Title/Abstract OR MeSH Terms OR Other Term. An example of this strategy is shown in listing 1 of section 2.1.2.

2.1.4 Staging database

A database was designed that reproduces the relevant parts of the PubMed article structure outlined in section 2.1.2. This database has two intended purposes: storing the extracted article data for further processing and allowing first data analyses in order to provide an early confirmation of the feasibility of the bibliometric approach. Storing the data mainly involves a large number of recurring write operations. Articles are likely to be retrieved from PubMed multiple times, either during subsequent searches for the same disorder term or because the article is found for multiple terms e.g. in the case of synonyms. However, it is not necessary to store data from the same article multiple times since this would unnecessarily inflate storage size, possibly slow down analytic queries and, in the worst case, distort query results. Redundancy is therefore avoided for article-related data such as titles, publication types, MeSH headings and

affiliations.

The database was designed as a relational 3rd normal form schema. As such, it is well suited for writing operations while avoiding unnecessary redundancy. Additionally, all relevant data elements are connected in a way that allows for a wide variety of preliminary analytic queries, although at the expense of performance due to the potentially high number of required join operations. The model is presented in detail in section 3.1.3. No special requirements were identified which would warrant the use of a specific database management system (DBMS). Therefore, the database was created using Microsoft SQL-Server 2012 as it integrated well into the existing development environment.

2.1.5 Extraction application

A *C#* application was built which combines the aforementioned elements into a cohesive extraction process. The application is aimed at the goal of managing the extraction of PubMed metadata as mentioned in section 1.3. As such, it needs to access the thesaurus to retrieve search terms, call the E-Utilities SOAP endpoint to find articles according to the search strategy outlined in section 2.1.3. The application then needs to fetch and extract data from a large number of publications, and store or update the entries in the dedicated sections of the staging database.

Since the extraction application is not supposed to be provided to a wide outside audience, a full-scale requirements analysis was not performed. Still, a few features were outlined which guided the overall development. The application has to allow for the automatic execution of searches in PubMed, including the extraction of retrieved data for any diseases in the thesaurus. "Automatic" in this case means that the process can be manually started but should then run on without requiring manual interference until a specified end e.g. until all available disease terms have been processed. Although performance optimisation has not been a primary goal for the application, options were examined how search runs could be carried out in a way that allows the extraction of a large amount of publication data in a short amount of time within the server-side limits set by PubMed and the E-Utilities API.

The application was built around a central controller which provides multiple interface functions to achieve the necessary flexibility. At its core, a fetching algorithm

with iteratively decreasing article contingents was implemented for the retrieval of a large amount of data from PubMed. In addition to this basic functionality, it became beneficial to enable the curation of the thesaurus to a certain degree from within the application. This involves the possibility of manually adding, editing or deleting disease terms, possibly in cooperation with a rare diseases expert. For this reason, it was decided to implement an administrative graphical user interface (GUI) which provides these possibilities as well as options for controlling and monitoring the extraction process. This administrative user interface has been implemented within the same C# application project using the Windows-Forms library. The detailed architecture and operating principles of the resulting application as well as key figures of the extraction runs are presented in section 3.1.4.

Database access All communication with the staging database is realised through the ADO.NET *Entity Framework* (EF). The EF provides an integrated way of connecting the application to a database in an object-oriented fashion by using an object-based model of the actual database, the so-called Entity Data Model (EDM). The EDM realises Object-Relational Mapping (ORM) i.e. the data model is represented as a collection of classes where each class is mapped to an actual database table. ORM often involves creating the data model first alongside writing the application. The actual database is then created from this data model i.e. forward engineered. Here, a bottom-up approach of creating the database schema first, based on the PubMed article structure and the relevant thesaurus parts, was chosen. Thus, the data model was automatically reverse engineered from the database and then manually adjusted for accuracy.

Using the EDM has the advantage that all interaction with the database can be limited to the classes of the data model. The EF handles connecting to the database, creating and executing all necessary commands for creating, reading, updating or deleting (CRUD) data and parsing any results back into class objects. It is therefore not necessary to write any specific Structured Query Language (SQL) statements for the application. *LINQ to Entities* (L2E) was used for the extraction application since it is the recommended way of using the EF [50, pp.435-466]. With L2E, the traditional SQL functionalities such as filtering, joining or grouping are realised as special operators which are in turn implemented as in-language extension methods. Listing 5 shows

an exemplary L2E query for retrieving a disorder list. The data model is loaded as the so-called *Context* from which information can be queried. In the example, the *Diseases* class is joined with the *SearchTerms* class on matching Orpha-numbers. Additionally, the *Where* operator is used to restrict the results to Orphanet main terms (*Origin == Orphadata*) in English (*Language == "EN"*). Each disorder's Orpha-number and term are projected as contents of the resulting list.

Listing 5: *Exemplary LINQ to Entities query for retrieving a disorder list of Orpha-Numbers and their Orphadata main terms from the thesaurus.*

```
using (PubMed_StagingEntities context = new PubMed_StagingEntities())
{
    var diseases = context.Diseases //(=FROM)
        .Join(  context.SearchTerms
            .Where(t=> t.Origin == "Orphadata" && t.Language == "EN"),
        d => d.OrphaNo, //join criterion 1
        t => t.OrphaNo, //join criterion 2
        (d, t) => new { d.OrphaNo, t.Term}); //(=SELECT)
}
```

Extraction process The entire extraction process, assuming that a full automatic search is to be performed, is summarised as pseudocode in algorithm 1. The algorithm iterates through each term of each disorder, assembles the *searchString* to be used in the *ESearch* query and performs the call against the E-Utilities API. The result is processed according to an iterative decreasing-block-size approach and publication data for each article is either newly stored or updated.

Algorithm 1 performPubMedSearch

```
load configuration;                                          ▷ search fields to be used
load thesaurus;                                              ▷ containing search terms
for all disease ∈ thesaurus do
    for all term ∈ disease do
        term = scan(term);                                   ▷ use quotation marks if necessary*
        searchString = assemble(term);                       ▷ according to configuration
        searchResult = performESearchQuery(searchString);    ▷ via eUtils
        repeat
            fetchBlocks = split(searchResult);               ▷ Initial block size: 500 articles
            for all block ∈ fetchBlocks do
                fetchResult = performEFetchQuery(block);     ▷ via eUtils
                if eFetchQuery was successful then
                    for all article ∈ fetchResult do
                        if article does not exist in DB then
                            extractData(article);            ▷ store in Staging-DB
                        else if publicationStatus has changed then
                            updateData(article);             ▷ update in Staging-DB
                        end if
                    end for
                    remove from fetchBlocks;
                end if
            end for
            decrease block size;                             ▷ subsequent block sizes: 100, 50, 10, 1
        until all blocks processed or loop finished with minimal block size
    end for
end for
▷ * if necessary = if the search term contains problematic expressions such as 'or'
```

Extraction runs Starting in February 2014, a total of three consecutive extraction runs were conducted whereby intermittent updates to the thesaurus i.e. additional disorders and necessary restructures e.g. changes to how affiliations are represented within the PubMed article structure lead to several interruptions. Since the first run had to retrieve all available rare diseases publications from PubMed, it took roughly seven months to complete. Subsequent runs, which only had to capture new articles and update a number of existing ones, proceeded much faster i.e. in a matter of a few weeks each.

2.2 Data processing

After being extracted from PubMed, the raw data needed to be processed and analysed in order to make it usable for creating author profiles. Data processing in this work concerned two major steps: affiliation segmentation and author name disambiguation. Affiliation segmentation and analysis is important for gathering additional information from institutional data which is useful in a number of ways, i.e. from enabling localised expert searches to supporting the disambiguation process. Disambiguation is required in order to alleviate the issues arising from multiple different authors sharing the same name and to allocate all publications to the correct person, allowing for the calculation of aggregate data for each author. The disambiguation process is oriented towards the basic three-step methodology outlined in section 1.2.3, i.e. name grouping, pairwise similarity computation and clustering.

2.2.1 Affiliation analysis

One major step in refining the extracted data is the segmentation and analysis of affiliation strings. Corresponding institutions of authors are submitted to PubMed as one cohesive text, despite containing multiple valuable data items. These items are relevant in a number of ways from using them for disambiguation to providing them as search and filter options to the end user. A common affiliation structure that can be found generally comprises the department, institution, the detailed address, the institution's city, the region, the institution's country and the author's e-mail address. Region pertains to federal states or other administrative areas within a country. Not all of these items are present in every affiliation. An example of this structure, barring regional information, is given in listing 6. In order to make full use of this information, it needs to be extracted and correctly classified. This, however, is complicated by the lack of standardisation. While a large number of affiliations have been found to follow a certain structural pattern, many other differ in the amount and order in which the information occurs, ranging from a single e-mail address to detailed departmental information.

Listing 6: *Example of an affiliation string from PubMed*

```
Department of Physiology, Ulm University, Albert-Einstein-Allee 11, Ulm 89069,
    Germany. frank.lehmann-horn@uni-ulm.de
```

Jonnalagadda and Topham [48] attended to this matter with NEMO (*Normalization Engine for Matching Organizations*) which presented a process to extract organisation names from affiliation strings and map them to canonical names. They used regular expressions to split up such a string into several parts and combined it with multiple dictionaries, named entity recognition (NER) and machine learning techniques in order to iteratively analyse each part. This process also covered the detection of other relevant information parts such as cities or countries and was therefore used as a template for creating a simplified segmentation algorithm for this project, which is primarily based on a number of lookup tables.

Country

CountryCode	CountryName	LocalName	Synonym
CH	Switzerland	Schweiz	Suisse
CI	Côte d'Ivoire	NULL	NULL
CK	Cook Islands	Cook Islands	NULL
CL	Chile	Chile	NULL
CM	Cameroon	Cameroon	NULL
CN	China	Zhong Guo	NULL
CO	Colombia	Colombia	NULL
CR	Costa Rica	Costa Rica	NULL
CU	Cuba	Cuba	NULL
CV	Cape Verde	Cabo Verde	NULL
CW	Curaçao	NULL	NULL
CX	Christmas Island	Christmas Island	NULL
CY	Cyprus	Kíbris	Kypros
CZ	Czech Republic	Ceska Republika	NULL
DE	Germany	Deutschland	NULL

Region

RegionID	CountryCode	RegionName
704	CZ	Moravskoslezský kraj
705	CZ	Olomoucký kraj
706	CZ	Pardubický kraj
707	CZ	Plzenský kraj
708	CZ	Praha, hlavní mesto
709	CZ	Stredoceský kraj
710	CZ	Ústecký kraj
711	CZ	Vysocina
712	CZ	Zlínský kraj
713	DE	Baden-Württemberg
714	DE	Bayern
715	DE	Berlin
716	DE	Brandenburg
717	DE	Bremen
718	DE	Hamburg

City

City_ID	CountryCode	CityName
64572	FR	Yvré-l'Évêque
64573	FR	Yzeure
64574	FR	Zaessingue
64575	FR	Zellenberg
64576	FR	Zetting
64577	FR	Zilia
64578	FR	Zoteux
21585	GB	Abbots Bromley
21586	GB	Abbots Morton
21587	GB	Abbotsbury
21588	GB	Aberaeron
21589	GB	Abercynon
21590	GB	Aberdare
21591	GB	Aberdaron
21592	GB	Aberdour

Figure 2: *Lookup tables for identifying countries, regions and cities in affiliations*

A location-lookup database was created which comprises information about countries, regions and cities. Countries are identified by their *United Nations Code for Trade and Transport Locations* (*UN/LOCODE*). Country names are available in English, the respective local language and in some cases a third synonym which might be encountered in publications as well. Regions are identified by an incremental surrogate key. The *UN/LOCODE* serves as reference to the respective country. The *City* table is built in the same manner with the names being available in the local language. Examples of all three lookup tables are shown in figure 2.

The first step of this simplified algorithm is to detect an e-mail address within the entire affiliation string. This is done by using the regular expression (RegEx) shown in listing 7. If an e-mail can be detected, it is removed form the string. In the second step, the affiliation string is segmented into multiple parts using the RegEx shown in

listing 8. This expression was also used by Jonnalagadda and Topham [48]. A graphical illustration of this segmentation and the different analysis steps are shown in figure 3.

Listing 7: *Regular expression used to detect e-mail addresses in affiliation strings*

```
([a-z0-9_\\.-]+\\@[\\da-z\\.-]+\\.[a-z]{2,6})\\.*
```

Listing 8: *Regular expression used to split PubMed affiliation strings into multiple parts for further analysis; provided by [48]*

```
(?<!(\\.))[^\\,\\;\\:\\@]+((\\,|\\;|\\:|$)(?=(\\p{Z}|[A-Z]|$))|(\\.)
    (?=(\\p{Z}|$)))
```

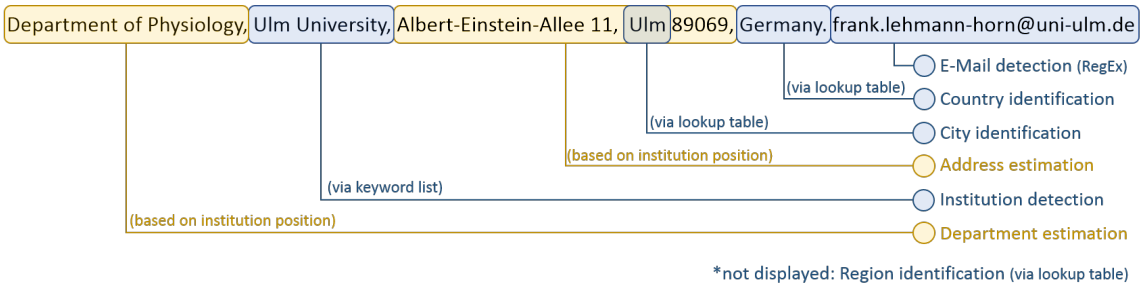

Figure 3: *Illustration of the segmentation and analysis of affiliation strings*

The distinct segments of the string are analysed in reverse order i.e. starting with the last element to first identify the country using the *Country* lookup table. Next, depending on how many elements were contained in the affiliation string, the city and region are identified based on their respective lookup tables. If the city is not available as a stand-alone value, the institution name might still contain the city name e.g. "Ulm University Hospital". In that case, an extended lookup is performed i.e. the phrase is further broken down and its parts checked again for matching city names. If the country is not available from the affiliation, it can be looked up retrospectively using the city name if the city has been unambiguously identified. Each part that can be matched against the lookup tables is removed and the remaining parts are checked for possible institution entries using a keyword list of English, German, French and Spanish words. The institution keyword list was adapted from [48] and extended by additional terms

e.g. “Universitätsklinikum”. Finally, after an institution name could be matched, any remaining segments to the “left” are regarded as the department designation while the remaining segments to the “right” are regarded as parts of the institution’s address. The algorithm can be viewed in detail in algorithm 2.

Algorithm 2 Affiliation-Analysis

```
1: if identify email then                                          ▷ via RegEx
2:     store and remove email;
3: end if
4: segments = split affiliationString;
5: if segments[last]->identify country then                        ▷ Country lookup table
6:     store and remove country;
7:     if segments.size > 2 then
8:         if segments[2nd-last]->identify city then               ▷ City lookup table
9:             store and remove city;
10:            if segments[last]->identify region then             ▷ Region lookup table
11:                store and remove region;
12:            else
13:                if segments[last]->identify city then           ▷ extended lookup*
14:                    store and remove city;
15:                end if
16:            end if
17:        else if segments.size == 2 then
18:            if segments[last]->identify city then               ▷ extended lookup*
19:                store and remove city;
20:            end if
21:        end if
22:    end if
23: else if segments remaining then
24:    if segments[last]->identify city then                       ▷ extended lookup*
25:        store and remove city;
26:        identify and store country;                             ▷ based on city
27:    end if
28: end if
29:        ▷ *”extended lookup”: the segment is split into multiple parts. Each part is
    matched against the lookup table
```

In order to evaluate the effectiveness of this algorithm, 1.000 processed affiliations were randomly picked and the segmentation results manually annotated. For each

segment, it was noted whether information about it was present in the original affiliation and whether the segmentation algorithm correctly detected said information and assigned it to the correct field. Sensitivity, specificity and accuracy were used as performance measures. Sensitivity i in this case is the ratio of all correctly assigned entries i, provided that the information i were indeed available, whereby i depicts a certain type of segment such as a city or a country. As such, the sensitivity formula is shown in equation 1.

$$Sensitivity_i = \frac{CorrectlyAssigned_i}{InformationAvailable_i} \tag{1}$$

Specificity in this context, as shown in equation 2, is the ratio of correctly assigned entries provided that there was actually no information available in the original affiliation. A correct assignment in this case means that the entry is empty and no false information has been assigned to the respective segment.

$$Specificity_i = \frac{CorrectlyAssigned_i}{InformationNotAvailable_i} \tag{2}$$

Depending on whether or not certain information is available, it can also be correct if no value is assigned to the respective data field. Therefore, an additional accuracy metric was introduced which compares the overall number of correctly assigned entries to the overall number of analysed entries i.e. 1.000. Accuracy is formalised in equation 3.

$$Accuracy_i = \frac{CorrectlyAssigned_i}{AllEntries} \tag{3}$$

Lastly, the availability of information items i.e. the ratio of how many of the analysed entries contained a certain type of information in the first place, as is depicted by equation 4, was determined.

$$Availabiliy_i = \frac{InformationAvailable_i}{AllEntries} \tag{4}$$

The results of the evaluation are presented in section 3.2.1. The segmentation algorithm was applied to all affiliations that were extracted from PubMed in preparation of name disambiguation and further data processing.

2.2.2 Naive grouping as a baseline approach

In order to have a baseline grouping of author names to compare the results of the disambiguation process described in section 2.2.4 to, a naive approach which is based on exact name matches was examined. This simple approach, at first, compares last names and then groups all authors who share a common first name. If the first name is not available or only consists of the initials, it is grouped with other instances which only have the initials available. This naive grouping is, however, prone to ambiguity issues which are illustrated in figure 4.

Illustrating different scenarios In order to illustrate the effects of different grouping and disambiguation approaches discussed in this work, figure 4 shows five imaginary publications, i.e. author instances, which were retrieved from PubMed and belong to two actual persons. The correct allocation of the instances to these authors can be seen at the bottom right of the figure where the two publications with an incomplete first name ("M. Baumann"), belong to a different author each. The third publication with an incomplete first name and a typographic error in the last name ("M. Bauman") belongs to the second author. It can be seen that the naive grouping approach b) would falsely aggregate the two publications which belong to different authors but share the exact same combination of initial and last name while all other publications would remain in their own group.

The ideal, i.e. correct, allocation of these publications to the authors could be achieved by a disambiguation process which, during the blocking step, takes into consideration that e.g. misspelled last names could actually belong to similar names as is shown with approach c) of figure 4. Most disambiguation approaches found in the literature do not consider name similarities and would incorrectly place the instance with the misspelled name into its own cluster as is shown with approach a) at the top of figure 4. Therefore, the disambiguation approach examined in this work was set up to consider similarities between last names for grouping.

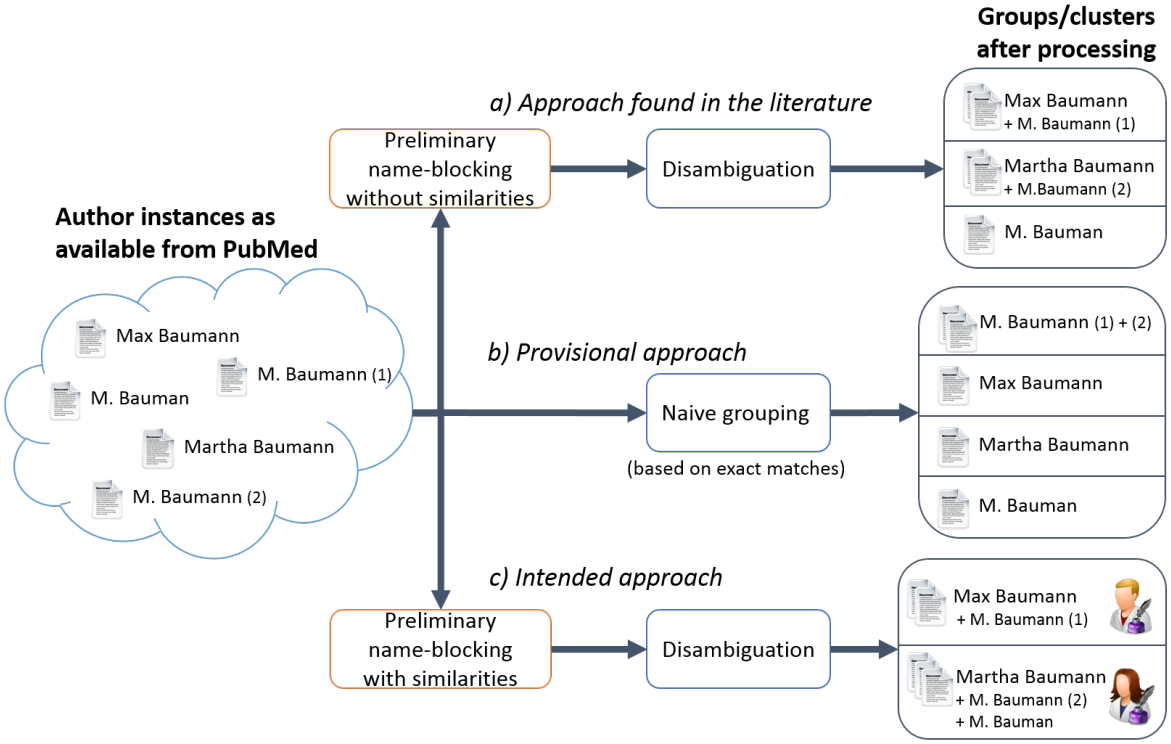

Figure 4: *Illustration of different grouping approaches for single author instances, i.e. publications, from two actual authors. The top approach (a) which is often found in the literature applies a preliminary blocking step based on matching last names and first initials exactly. The subsequent disambiguation step clusters the author instances for each name-block based on additional publication features such as co-authors, journals or affiliations. With this approach, the publication with the misspelled author name ("M. Bauman") is assigned to a different name-block than the other four instances and thus incorrectly clustered. The bottom approach (c) shows the ideal result which was aimed for in this work. This correct allocation of all publications should occur when disambiguation is applied after considering potential similarities between last names during the blocking step which includes the entry with the misspelled last name. In both cases, it is assumed that the disambiguation process itself is free of errors. The naive grouping approach (b) displayed in the centre is based on exact matches of both first and last names. While this grouping is easily achieved, it is prone to producing mostly false results compared to the actual distribution.*

2.2.3 Name similarity allocation

Prior to grouping author names into blocks for disambiguation, it was necessary to find an effective way of measuring similarity between names in order to be able to collate not only identical but sufficiently similar names while keeping independent names apart. For comparing different similarity metrics against each other, a reference dataset of manually annotated name associations was created. Each available metric was applied to the same names and the matching results were compared to the reference data.

Similarity metrics A variety of similarity metrics are eligible for this approach. Bilenko et al. [31] present a categorical overview of string similarity metrics and compare them on multiple data sets for different usage scenarios. One of those is a set of synthetic, census-like text fields including person names, which is comparable to the existing problem of matching author names. In their work, three metrics, *Edit-Distance*, *Jaro* and *TF-IDF* (*Term Frequency - Inverse Document Frequency*), performed very well on the census data set. The *Edit-Distance* metric is a measure of how many edit operations (copy, insert, substitute or delete) need to be performed on a string s1 at minimum until it matches another string s2 so that s1' == s2. The distance metric automatically chooses the best sequence in which the edit operations take place. The most basic implementation of this is the *Levenshtein distance* which uses a simple unit-cost scheme. More complex metrics can associate different costs for each edit operation or provide certain discounts depending on the positions within the strings like the *Needleman-Wunsch* and *Smith-Waterman* distances respectively. Yet, Bilenko et al. found the basic Levenshtein distance to perform best on the census data set [31]. However, Liu et al. [55] describe the edit distance as being problematic for matching e.g. Chinese names.

The *Jaro* metric measures similarity between two strings based on the number and order of matching characters. A mathematical representation of the metric as well as an example are given in [31]. With the *Jaro-Winkler* variant, matches between the initial few characters of two strings are especially weighted and the metrics have been found in the same work to be very suitable for short strings such as personal names. In their census data comparison, the *Jaro metric* ranks second. In another work, Cohen et al. [34] found that a "soft" variant of TF/IDF performed best on another set of census-

like data as well as a set of bibliometric data such as author names, title or venue. TF/IDF, a widely used metric in the information retrieval community is described in detail in [31].

Other metrics that were subject of [31], [34] and [37] such as *Jaccard*, *Monge-Elkan* and *N-Gram* (n=3) and additionally, the *Metaphone* scheme which compares strings by their pronunciation using a phonetic algorithm, were examined for this work.

Creating a reference list of name associations The reference dataset was created by randomly selecting last names from the extracted data and producing one or more preliminary associations to a second name based on the *DoubleMetaphone* phonetic algorithm, which is an extension of the original *Metaphone* metric. Of these associations, 25 names for each letter of the Latin alphabet were again randomly selected in order to reduce the dataset to a size which could be manually processed. Due do multiple associations per name, the final list comprised 651 distinct names with a total of 1.186 associations. An excerpt from this list is shown in table 1.

Table 1: *Excerpt of name associations in the reference dataset. For each name, it was manually annotated whether the associated name is considered to be a possible match (1) or not (0). The differences between names have been categorised with the categories being listed in table 2.*

Name	Association	Possible Match	Category
ABOLTINS	ABOLT	0	7
ABU-HASSAN	ABOU HASSAN	1	6
AGOULNIK	AGULNIK	1	1
AKOPIAN	AGOPIAN	1	3
ALCOVER	ALCOVER GARCÍA	1	8
ALES	ALLS	0	4
CHAFFAI	CHAFFAÏ	1	5
QU	QUA	0	2

For each association on this list, manual annotation was carried out on whether or not a surname was considered to be a variant of the name it had been associated to. Additionally, a reason for accepting or declining each match was noted. These reasons can be grouped into a total of eight categories which mostly refer to single character differences and are listed in table 2. The table also shows how often each category was

encountered in the manually annotated dataset. Only syntactical differences between strings were examined while phonetic differences were disregarded. Of the 1.186 associations, 863 were marked as sufficiently similar, i.e. a positive matching decision was made, while 323 were seen as too different.

Table 2: *Categories of differences between two surnames along with the number of cases in which these differences occurred in the manually annotated dataset. Examples for each category are shown in table 1.*

Category	Description	Matching decision	Cases
1	Single additional or missing character	positive	369
2	Single additional or missing character	negative	14
3	Single edited character	positive	253
4	Single edited character	negative	230
5	Single edited special character	positive	86
6	Multiple additional or missing characters	positive	34
7	Multiple additional or missing characters	negative	79
8	Double surname	positive	121
$\sum$			**1186**

Comparison of different metrics The same names which were used in creating the reference list were now again matched by each available similarity metric using Talend Open Studio (TOS). The *Levenshtein distance* and the *Metaphone* algorithm are a part of the freely available version of *TOS for Data Intrgration.* For the remaining metrics, the *SecondString* Java library was imported. While some metrics automatically make a matching decision, others create a similarity score for each two strings. For the latter, thresholds above which the strings are considered to be a match need to be defined. For the Levenshtein distance, a maximum distance of 1 was used. For each of the probabilistic methods such as Jaro-Winkler, various matching thresholds were examined in order to find the best result.

Normally, with the *1-Edit-Distance*, all names which differ only in a single character would be matched. While this approach was found to already produce good results, the manual annotation showed several cases (230) where a single substituted letter changed the names enough to no longer consider them similar, thus having the highest number of non-matches in table 2. Based on these cases, additional rules for the edit-distance

were derived which reject or accept two names to be matched if the involved characters meet certain conditions. These are shown along with their exceptions in table 3. Using these rules, an *Extended-1-Edit-Distance* was implemented.

Table 3: *Additional rules for an* Extended-1-Edit-Distance. *The rules only affect strings where a single character was substituted. Matching decision declares whether two strings are matched if the primary conditions are met. The decision changes if the exception conditions are met.*

Rule	**Matching decision**	**Exceptions**
Exchange affected the first character of the strings	negative	none
Vowel exchanged for consonant or vice versa	negative	none
Consonant exchanged for different consonant	positive	The exchanged consonant was part of a double consonant (e.g. TT)
Vowel exchanged for different vowel	negative	Vowels are next to each other on keyboard (U-I and I-O) or the exchange affected the last character of the strings

Recall and *Precision* were used to compare each algorithm's performance to the manually annotated reference dataset. Recall in this case can be defined as the proportion of correctly matched names by the algorithm in question ($CorrectMatches_{Metric}$) compared to all matching names of the reference list ($CorrectMatches_{Reference}$) as is shown in equation 5.

$$Recall = \frac{CorrectMatches_{Metric}}{CorrectMatches_{Reference}} \tag{5}$$

Precision can be defined as the proportion of correctly matched names by the algorithm in question ($CorrectMatches_{Metric}$) compared to all name associations made by the algorithm ($AllMatches_{Metric}$).

$$Precision = \frac{CorrectMatches_{Metric}}{AllMatches_{Metric}} \tag{6}$$

Additionally, the F1-score, i.e. the harmonic mean between Recall and Precision,

and the F2-score, which rates recall higher than precision, were used as homogeneous comparison metrics. A higher recall rating was considered due to initial coverage being regarded more important than an optimal avoidance of false matches as the latter can still be resolved at a later stage. The generalised F_β formula is shown in equation 7.

$$F_\beta = (1 + \beta^2)\frac{Precision * Recall}{\beta^2 * Precision + Recall} \tag{7}$$

After evaluating each metric on its own, the most promising ones were examined in combinations of two whereby both union and intersection results were considered. All evaluation scores can be found in section 3.2.2. Finally, the metric combination of *Jaro-Winkler* $\bigcup$ *Extended-1-Edit-Distance* with a matching threshold of 0,87 for the *Jaro-Winkler* metric was chosen as the basis for allocating name similarities.

Allocation process The grouping process is shown in algorithm 3. In order to exclude independent names from being matched, an instance count threshold was introduced. Author names that occurred more often than the threshold value were not considered a variation of any other name and were thus not allocated. The algorithm iterates through all names below the threshold and matches them against those names above the threshold which feature an identical first initial. As an example, the name *A. Pflugard* could be matched and grouped with *A. Pflugrad* but not with *M. Pflugrad*. For both similarity metrics, the best match among all eligible names is determined. For the Jaro-Winkler metric, this is the match with the highest score or, should multiple matches with the same score exist, the one with the highest instance count. For the Extended-1-Edit-Distance where no score is allotted, the instance count is the determining factor for the best match. The final allocation decision is based on the score and instance count of both best matches. If the instance count of the best Jaro-Winkler match is higher than that of the Edit-Distance match or its score is above a certain additional threshold (0,95), the name is allocated to the Jaro-Winkler match. In any other case, the name is allocated to the best Edit-Distance match.

Algorithm 3 Name Similarity Allocation

define *threshold*; ▷ Below which names could be considered typos
for all *nameA* **where** iC $\leq$ *threshold* **do**
  for all *nameB* **where** iC $>$ *threshold* $\cap$ *nameA*(fI) $==$ *nameB*(fI) **do**
    find *matchJW* = JW.match(*nameA*) **where** (score(JW) $\cap$ iC) $==$ **max**;
    find *matchE*1 = E1.match(*nameA*) **where** iC $==$ **max**;
  end for
  if *matchJW*(iC) $>$ *matchE*1(iC) $\cup$ *matchJW*(score) $\geq$ *thresholdJW* **then**
    allocate *nameA* $\mapsto$ *matchJW*;
  else
    allocate *nameA* $\mapsto$ *matchE*1;
  end if
end for

iC: instanceCount i.e. how often a name (fI + surname) appears overall
fI: firstInitial
JW: Jaro-Winkler metric
E1: Extended Levenshtein 1-Edit Distance metric
thresholdJW: threshold above which matchJW is always peferred

2.2.4 Examining author name disambiguation

After grouping author instances into blocks based on name similarities, a disambiguation process was examined with the goal of transforming each name block into one or more distinct persons with their attributed articles. Disambiguation is required for all blocks that contain at least two author instances.

Exemplary works The two existing disambiguation approaches which specifically deal with PubMed author data, namely those by Torvik et al. [80, 81] and Liu et al. [55], served as an example. Both base their similarity computation on various features that are available from the publication data and apply different weights to each feature. After a probabilistic hierarchical clustering on the similarity profiles, they apply additional post-processing steps in order to receive their final author clusters. Since the development of a fully sophisticated disambiguation system is beyond the scope of this work, only the core features were adopted as much as possible. These are weighted similarity profiles and hierarchical clustering. The goal was to implement a

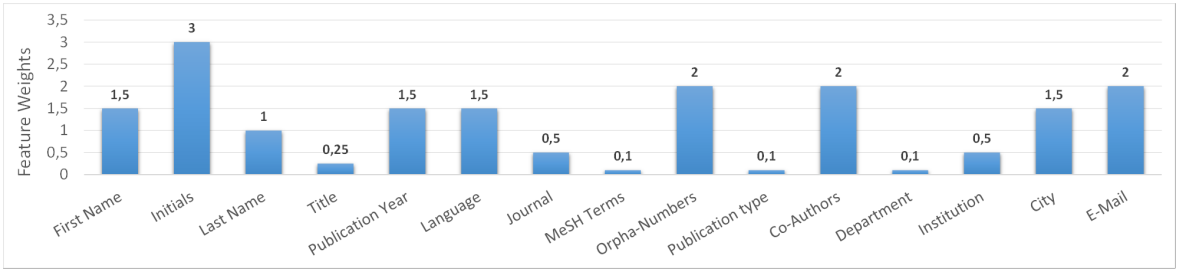

Figure 5: *Features and their respective weights used for disambiguation.*

simplified process which would still be able to achieve a better author distinction than having no disambiguation in place.

Disambiguation approach A total of 18 data fields were considered for computing the pairwise similarities between the instances of each block. These include the first name, initials and last name of the author. Co-authors are represented by their respective name-block ID. The publication title was shortened based on the stop word list provided by PubMed [28]. Further fields are the publication year and language as well as the abbreviated publication venue i.e. journal. The IDs of all MeSH descriptors and qualifiers associated with the publication, publication types and associated Orpha-numbers are also used. Finally, information about departments, institutions, address parts, cities and e-mail addresses is included as far as it is available from the affiliation segmentation presented in section 2.2.1. 15 of those data fields were eventually used for disambiguation with their initial weights being adopted from Liu et al. [55]. These features and their weights are illustrated in figure 5.

In order to perform both similarity computation and clustering within the same framework, the *R* free software environment for statistical computing was used as it provides all of the tools for the necessary steps. For each name-block, the disambiguation database is accessed and all entries are retrieved. Pairwise similarity computation is done via the *daisy* package, which actually computes a dissimilarity matrix. *Daisy* is used in conjunction with the *Gower* metric as it allows for previously specified weights to be taken into consideration. On the basis of the dissimilarity matrix, clustering is performed using the *agnes* (agglomerative nesting) method. *Agnes* performs a bottom-up hierarchical clustering starting at height = *hZero*. Each data point i.e. author instance is first regarded as a single cluster. Each nearest two instances with the

smallest dissimilarity are then joined into a cluster. This process continues until eventually all data points are contained in a single cluster at height = *hMax*. An example of such an hierarchical clustering result, represented in the form of a dendrogram, is shown in figure 6.

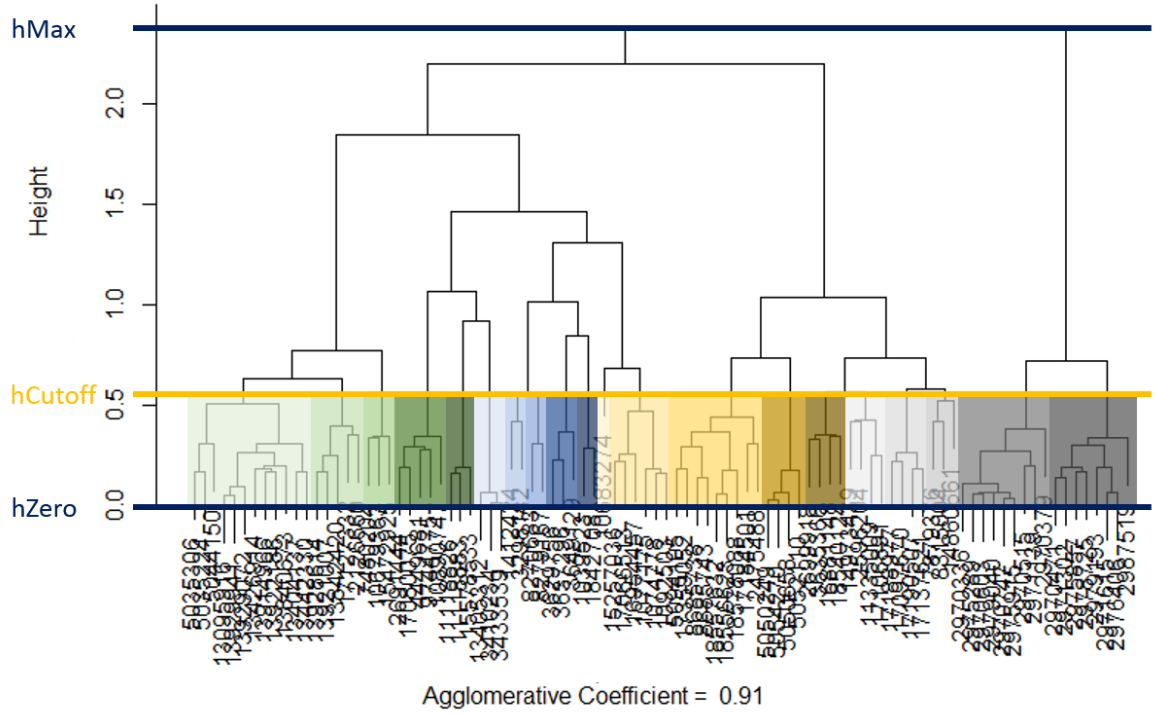

Figure 6: *Annotated example of an author-name-block after using agglomerative nesting. For this particular name-block, the ideal cut-off height to reveal the actual author clusters would be 0.56.*

For the last step in determining which instances actually belong to the same person, it is necessary to estimate a cut-off height *hCutoff* below which each cluster that has been formed thus far constitutes an author whereby all single instances contained in the cluster represent the author's publications. In the exemplary approaches [81, 55], this is determined by a mathematical model based on probabilistic inter-cluster-similarities. Here, a different approach was examined which uses a fully automatic method for *R*. The *Dynamic Tree Cut* package, suggested by Langfelder et al. [51] was originally developed for the use with genomic data. Their algorithm detects clusters in a dendrogram based on the shape of its branches and determines the associated cut-off

height.

A number of parameters can be varied to influence the tree cut method. While most of them were left in their default setting, the *minGap* parameter was of particular interest. With this parameter, the minimal gap between what is recognised by the method as the "core" of a potential cluster and the height at which it is merged into the next bigger cluster can be set. A higher value for *minGap* will therefore result in stricter conditions for a cluster to be detected as such. In order to allow for a fully dynamic mode of operation, a value that is inherent to the dendrogram had to be chosen. Using the overall height of the dendrogram in any way proved difficult as it can vary greatly depending on the number of instances in a name-block. Instead, the *Agglomerative Coefficient* (AC) was examined. This value measures the amount of clustering structure found in the data set i.e. very clearly defined cluster structures will have a high agglomerative coefficient.

Two scenarios for the use of the the AC have thus come into question: the more clearly defined the cluster structure, the more strict should the conditions for cluster detection be set for the dynamic tree cut method. In this case, the *minGap* parameter is set to an AC of different powers i.e. single, squared and cubed, for different tests. The second scenario that has been considered is: the more vague the clustering structure is, the more strict should cluster detection be. For this scenario, the *minGap* parameter is set to $1 - AC$.

Clustering process The clustering result is mainly influenced by two parameters: (1) the choice of features, i.e. data fields, and their respective weights to be included in computing the dissimilarities between instances and (2) the choice of the *minGap* parameter for dynamically finding the cut-off height in the cluster dendrogram. The overall clustering process which was implemented in *R* is shown as pseudo-code in algorithm 4 using the name-blocks that were created by the allocation process described in section 2.2.3.

Algorithm 4 Automatic name disambiguation

```
for all block ∈ nameBlocks do
    retrieve data values d;                          ▷ of all author instances in the block
    specify weights w;                                          ▷ one for each data value
    compute dissimilarity matrix dm = daisy(d(w));
    perform clustering dendro = agnes(dm);
    retrieve agglomerative coefficient ac from dendro;
    determine cut-off height hCutoff = dynamicTreeCut(ac);
    for all cluster c ∈ dendro(hCutoff) do
        assign cluster IDs;                              ▷ based on smallest instance ID
        store c;
    end for
end for
```

Evaluation dataset In order to evaluate the performance of this approach, an extensive testing dataset which would ideally include references to the definitive publications of a large amount of authors was not available. Instead, the *Author-ity* dataset from Torvik and Smalheiser [80] was used. In their work, they achieve very good results on a number of different evaluation datasets comprising, amongst others, an *ISI HighlyCited* dataset, self-citation datasets and a small manually disambiguated dataset. Therefore, the *Author-ity* data has been regarded as a quasi-gold-standard to compare the disambiguation approach at hand against.

Author-ity provides each identified author cluster as a data row containing the author's namespace similar to the name-blocks created by the similarity allocation in section 2.2.3 as well as the author's publications amongst other things. Since *Author-ity* records cover all of MEDLINE, the first step was to trim down the dataset and include only those records which match any of the rare diseases publications retrieved by the extraction process. Matching was done based on the PMIDs and each *Author-ity* record was supplemented with the author-instance-IDs from the staging database that correspond to each publication. Since the clusters which result from the disambiguation process also constitute collections of author-instance-IDs, the datasets become directly comparable.

Evaluation process The disambiguation process was repeatedly carried out with different feature-weights and *minGap* parameters. Each time, the resulting clusters

were compared to the clusters within the *Author-ity* dataset and the overlap calculated. The naive instance grouping as described in section 2.2.2 was carried out for each name-block and served as a baseline comparison.

With the feature weights remaining fixed, several variations of the *minGap* parameter were evaluated and the best setting i.e. the one with the most overlap was chosen for further evaluations. Different variations of the feature weights have been examined subsequently. The results of this evaluation are presented in section 3.2.3.

2.3 Deployment

Following the data processing and analysis steps, the data was transformed into the final structure. This structure would reflect the envisioned expert profiles including aggregate numbers e.g. of publications and allows for a client application to search and retrieve these profiles according to their connected disorders. A suitable user client needed to be developed which provides functions for finding rare diseases as well as browsing, sorting and filtering expert lists. Additionally, detailed single expert profiles needed to be created in order to allow users to estimate each author's expertise, thus fulfilling the goals outlined in section 1.3. The necessary components have also been presented in [72].

2.3.1 Data transformations

The data transformations for use by the client application comprised three steps: additional character cleaning in order to minimise syntactical inconsistencies, data grouping in order to get the relevant aggregate numbers for each author and, finally, structural changes of the data model into a query-optimised schema. Although basic character cleaning was performed when extracting data from PubMed, there were still inconsistencies in a number of entries. Thus, an additional, more thorough cleaning and harmonisation process was applied. This process resolved issues pertaining to inconsistent formatting, special characters outside of the Latin alphabet as well as duplicate or missing entries among others.

In order to create the expert profiles outlined in section 1.3, data from all distinct publications of the same author needed to be aggregated. It was intended to base this aggregation on the author name disambiguation presented in section 2.2.4 for

an accurate summary of each author's data. However, due to time constraints and inconclusive findings regarding the effectiveness of that particular approach, a different solution had to be found. With other approaches being unavailable, the naive baseline approach of grouping authors based on exact name matches as described in section 2.2.2 was used as a preliminary substitute for the system's prototype.

Since data retrieval from PubMed was performed on an article-by-article basis, the model of the staging database as is presented in section 3.1.3 is an article-centric normalised schema. As such, single publications are the central entity with any additional metadata being attached to each article. This kind of model is optimised for writing operations but entails performance restrictions when it comes to querying data. Also, with the intended usage being to not search for articles but for authors, the overall data structure was transformed into a model which is author-centric and optimised for performant querying. This introduced a certain degree of denormalisation.

2.3.2 User client

The final development step involved the implementation of a user client which realised the usage scenarios outlined in section 1.3 and could ultimately be provided to interested end users. The client was designed to be accessible via a web interface. Such an interface was developed using the *Django* web framework, an open source *Python* web framework which supports the rapid development of web applications. *Django* provides a broad variety of back-end libraries in combination with complementary JavaScript libraries for visualisation purposes. Both the data transformation operations of section 2.3.1 as well as the development of the user client were performed by Schwarzkopf [71]. The resulting client application and its underlying data are presented in section 3.3.

2.4 Evaluation

In order to evaluate the feasibility of the system for finding rare disease experts, multiple examinations were performed with the goal to show whether the system is able to:

1. represent authors of rare disease publications in a way that allows for an estimation of their expertise both in regards to a specific disease and overall

2. include author information in a way that allows for users to contact said authors i.e. display correct contact information

3. identify experts on rare diseases who can be confirmed by established expert registries

4. identify new experts on rare diseases who were previously unknown to other registries

Preliminary proof of concept Early evaluation steps towards a proof of concept involved preliminary checks on the staging database. From the available data, sample profiles of authors and disorders were created and compared to each other in order to see whether an author's expertise and the connection to a disorder would become apparent. These early profiles for an author included the number of articles published for each disorder as well as the associated MeSH terms and journals. Based on the counts of each feature, ranked lists were created for comparison. Disorder profiles were created in the same fashion. Their features include the authors who have published on the disorder with their respective publication counts and also the associated MeSH terms and journals. An example of both profiles is provided in figure 23 of section 3.4.

Expert-annotated author lists After completion of the data processing steps, author lists for specific disorders were created and presented to known clinical experts on those disorders. The experts were asked to annotate the lists and mark those authors which they can confirm as experts as well as those which they can confirm not to be experts respectively. Their feedback also included information on whether the author data contained in the lists was correct. The data included author names, the total number of each author's publications as well as the number of publications that were written as first, middle or last author and the number of reviews. Additionally, contact information such as the author's probable institution, city and e-mail address was included if it was available from the respective affiliations. An example of a list of German experts is shown in figure 7.

An additional set of lists was provided to the Orphanet encyclopedia team in Paris who evaluated the lists from their point of view. The set comprised lists on five different rare disorders chosen by the Orphanet team.

FirstName	LastName	First	Middle	Last	Total	Reviews	Latest	Institution	City
...	...	17	16	23	56	18	2014	Ludwig-Maximilians-University	München
...	...	12	32	7	51	4	2015	Campus Virchow-Klinikum	Berlin
...	...	10	26	6	42	9	2014	Ludwig Maximilians University of Munich	München
...	...	6	18	17	41	2	2015	University of Tübingen	Tübingen
...	...	6	26	1	33	2	2015	Medizinische Hochschule Hannover	Hannover
...	...	7	24	2	33	4	2015	University of the Saarland	
...	...	5	22	4	31	0	2015	University of Magdeburg Medical Center	Magdeburg
...	...	4	17	8	29	5	2014	University of Würzburg	Würzburg
...	...	12	10	6	28	7	2015	University of Ulm	Ulm
...	...	5	10	13	28	8	2015	Friedrich-Schiller University Hospital Jena	Jena
...	...	3	14	9	26	5	2015	Ulm University	Ulm

Figure 7: *Annotated author list for a specific disorder. The colours mark confirmed clinical (green) and theoretical (blue) experts while the others remain unconfirmed. The names were intentionally omitted for this work.*

Comparative evaluation based on verified expert-lists Another step included verified lists of experts for several diseases which were provided by Orphanet and the *Centre of Excellence for Rare Diseases Baden-Württemberg*. The diseases which were provided can be divided into major classes pertaining to the number of available expert profiles and whether they are disease groups i.e. with multiple subtypes or single entities. The first class contains diseases without subtypes and more than 5.000 available expert profiles worldwide. In the second class, there are diseases without subtypes for which comparatively few expert profiles (less than 3.000 worldwide) are available. Finally, the last class contains disease groups with subtypes. Each class of this evaluation set contains six diseases. For each disease, five verified experts were listed on average.

Based on these lists, the system was compared to Expertscape and se-atlas. For each disease and expert, it was checked whether the expert could be found via the user-client and each of the two existing registries respectively.

In-depth-evaluation with Orphanet In a last step, author lists similar to those that were presented to clinical experts were sent to Orphanet Germany for evaluation. There, it was checked whether the lists contained any relevant experts which were not yet part of the Orphanet expert registry. Due to time and resource restrictions, feedback could only be provided for a single list. The results of all different evaluation steps are presented in section 3.4.

3 Results

3.1 Data extraction

3.1.1 Rare diseases thesaurus

The initial search term thesaurus which was used for searching rare diseases literature contained a total of 16.075 terms pertaining to 6.781 diseases. On average, three terms were available for each disasese with some diseases having up to 21 different designations. Table 4 shows an excerpt from the thesaurus data with one entry from each category (*Origin*). Most terms originated from the Orphanet vocabulary with 6.781 main terms, i.e. one for each disease, and 8.004 synonyms. In addition, 870 MeSH terms and 316 MeDRA terms have been added along with 104 terms which were added retrospectively via the administrative user interface.

Table 4: *Exemplary search terms from the thesaurus. The Orpha-number is used as a common disease identifier across different terms. Origin depicts the vocabulary from which each term was imported except for the last entry, which was entered directly into the thesaurus via the extraction management application.*

OrphaNo	Origin	Term	Language
2970	Orphadata	Prune belly syndrome	EN
2970	Orphadata_Synonym	Eagle-Barret syndrome	EN
2970	MedDRA_PT	Eagle Barrett syndrome	EN
2970	MeSH_Reference	Urethral obstruction sequence	EN
680	Orphadata	Normokalemic periodic paralysis	EN
680	Entered by User	Normokalaemic periodic paralysis	EN

3.1.2 PubMed search strategy

Figure 8 displays the different amounts of articles returned when each of the four main search strategies, that were outlined in section 2.1.3 is employed.

It can be seen that allowing PubMed to interpret search terms yielded 9.016 additional results (+17%) compared to the Strict *TI;MAJR* strategy. The most significant increase in the number of articles (+74%) was achieved by adding the Abstract search field. Opening up the MeSH Major Topic restriction to include all MeSH Terms and

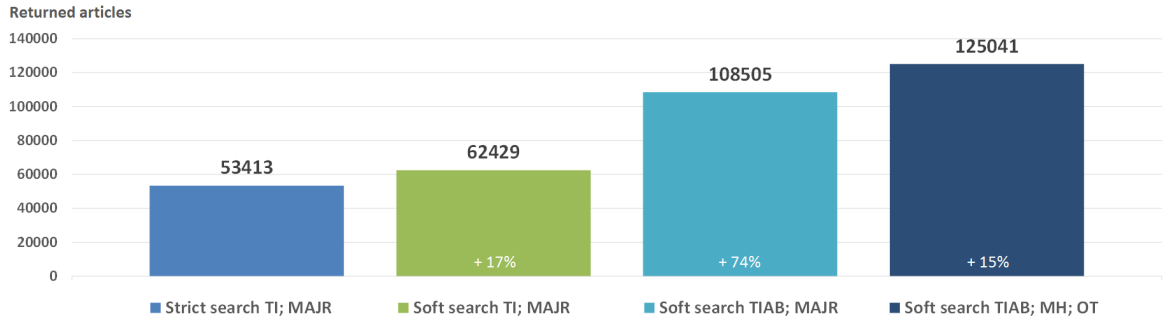

Figure 8: *Comparison of the amount of search results for the diseases of the first 100 Orphanumbers using 4 different strategies. The numbers show how many publications were returned by PubMed in total for each strategy along with the percental increase in results between strategies. "Strict search" disallows term interpretation by PubMed as opposed to "soft search". The search fields are: TI = Title, MAJR = MeSH Major Topic, TIAB = Title/Abstract, MH = MeSH Terms, OT = Other Term*

Other Terms has resulted in another 16.536 (+15%) articles. Looking at individual disorders, some increases were mainly due to enabling the automatic term mapping by PubMed. This was e.g. the case for *Pipecolic acidemia* and *Albers-Schönberg osteopetrosis*. For other disorders such as *Fatal infantile lactic acidosis with methylmalonic aciduria* or *Amoebiasis due to Entamoeba histolytica* none or only very few articles could be found regardless of the applied search strategy. This is likely due to the disorder names not being used in this form in any publication. Results for disorders such as *Acrocallosal syndrome* or *Proximal spinal muscular atrophy* greatly increased after adding the Abstract search field. Further investigations showed that this is caused by acronyms which were used for each of the affected disorders. These can result in a massive number of articles which are not related to the disorder in question. As an example, *Acrocallosal syndrome* when searched with its acronym *ACS* will, amongst others, retrieve publications regarding the *American Cancer Society*, *Acute Coronary Syndrome* or *Autologous Conditioned Serum*.

Further manual analysis identified a severe issue with the automatic term mapping, regarding disorder names which contain the word 'or'. An example of this is *Familial or sporadic hemiplegic migraine*. Without automatic term mapping, a query for this disease returns 86 results (as of April 2016). However, when term mapping is enabled and no match can be found in any of PubMed's translation tables, the term is further

split up and reprocessed [29]. In this particular case, the 'or' is then being interpreted as a logical OR and PubMed will return all publications that include the terms *family* or *familial* regardless of context, leading to 1.065.528 results instead. This example is illustrated in listing 9 with the problematic *OR* being marked. In order to prevent that queries are flooded with false positive results, automatic term mapping needs to be disallowed for these cases.

Listing 9: *PubMed search string for* Familial or sporadic hemiplegic migraine *after automatic term mapping was applied*

```
("family"[MeSH Terms] OR "family"[All Fields] OR "familial"[All Fields
   ]) OR (sporadic[All Fields] AND hemiplegic[All Fields] AND ("
   migraine disorders"[MeSH Terms] OR ("migraine"[All Fields] AND "
   disorders"[All Fields]) OR "migraine disorders"[All Fields] OR "
   migraine"[All Fields]))
```

3.1.3 Staging database

The final model of the staging database is shown in figure 9. It can be divided into 4 sections: *Article_Information*, *Authorship_Information*, *MeSH-Headings* and *Search_Information* with the *Article* table as the central connecting entity. *Article_Information* contains all basic information that is not further split into separate entities due to normalisation. Both *Authorship_Information* and *MeSH-Headings* are further normalised. *Authorship_Information*, which contains all available information about who authored each article, is regarded as a subject of further analysis and processing as described in section 1.2.2. Finally, *Search_Information* connects the data from PubMed with the Orphadata disorder terms. It contains the subset of the thesaurus which is used for searching articles for each disorder and also stores information about when and how corresponding queries were performed.

Article_Information This section gathers all basic publication data around the *Article* table. Information about the journal contains several values (journal title, abbreviation and ISSN) and is therefore kept in a separate table. Articles can be associated with multiple publication types and a single publication status which may change over time. An article's publication year and language are treated as atomic

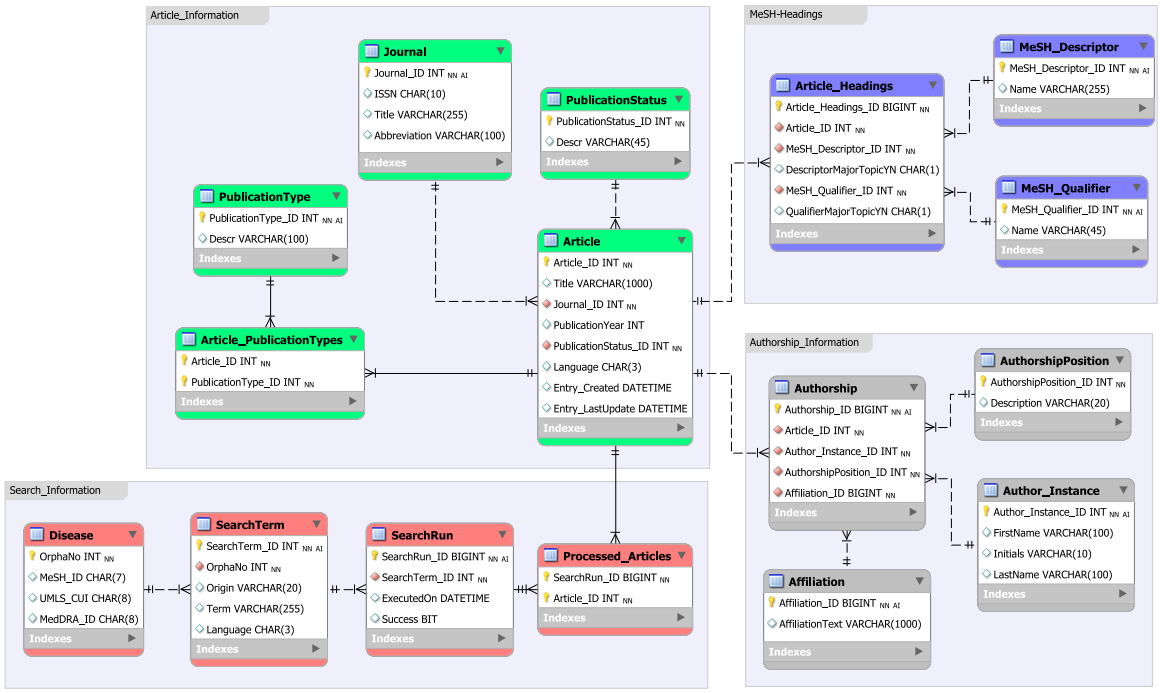

Figure 9: *Relational schema of the staging database*

values and kept within the *Article* table along with its title and PMID. Additionally, the dates when an entry was first created and last updated allow an overview of if and when articles may have been changed since first being captured.

Authorship_Information For each author of an article, an *Authorship* entry is created. It is the central entity of this section and connects the author name (i.e. *Author_Instance*) with the affiliation and position entry. *AuthorshipPosition* serves as a lookup table for the position in which each author of an article is listed in the author list. *AuthorshipPosition* comprises three possibilities: first, last or middle author. The first author position is of particular importance for estimating an author's contribution to a publication. While the last or supervising author position can also hold informational value, the middle authors are often deemed less important as has been outlined in section 1.2.2 or sometimes advised to be completely disregarded [68]. It was therefore decided to not store each middle author's exact position but instead condense this information into the middle author group. The *Affiliation* table contains the complete affiliation text for each author if it is available and a default text otherwise. Exactly

matching affiliation texts are reused i.e. not stored multiple times.

MeSH-Headings The MeSH-Headings section captures the keywords from the *Medical Subject Headings* vocabulary which have been attributed to each publication. These keywords are used as part of each expert profile and are also of potential use for a number of further analysis tasks as will be outlined in section 4.

Search_Information The data from the thesaurus is contained in the *Disease* and *SearchTerm* tables. Disease constitutes a basic overview of which rare disorders are covered by the system with the Orpha-number (attribute *OrphaNo*) serving as the main identifier. The disease terms and thus the main basis for searching articles are provided by the *SearchTerm* table. Here, each relevant term for each disease is stored and classified. *Term* contains the actual text while *Origin* in most cases relates to the database system from which the term was derived. Each term is associated with a specific *Language* and is provided with a surrogate key. The search terms are updated as the thesaurus is being expanded. *SearchRun* depicts a concrete search i.e. extraction run that has been performed for a specific term. This way, all searches can be tracked over time and repeated if necessary. This also allows to see e.g. how long two consecutive searches for the same disorder are apart and could be used to determine when the next update for a specific disorder is due. *ExecutedOn* stores the time of execution of each run while *Success* is used to gauge whether a specific run was successful or whether it was terminated prematurely. Finally, *ProcessedArticles* connects each search run with every article that has been retrieved during that search.

3.1.4 Extraction application

Overall architecture The multi-layer architecture of the management application to govern the data extraction process from PubMed is depicted in figure 10. The layer shown on top of the figure comprises two parts: the GUI for a system-side administrator to control extraction runs and edit the search term thesaurus (*View*) and *External Resources* outside of the immediate system environment, i.e. the E-Utilities API. In the *Business Logic* layer, the core mechanics of the application are implemented. This includes the code behind the administrative user interface, designated *UI code base,* which can be compared to the controller in a model-view-controller (MVC) concept. It

relays user commands to the *ExtractionRunController* via several interface functions and reports the results back to the GUI. It also can interact with the data model to retrieve any information that needs to be presented to the administrator or to store changes to the thesaurus. Lastly, the Persistence layer realises the access to the staging database using the *Entity Framework* as described in section 2.1.5.

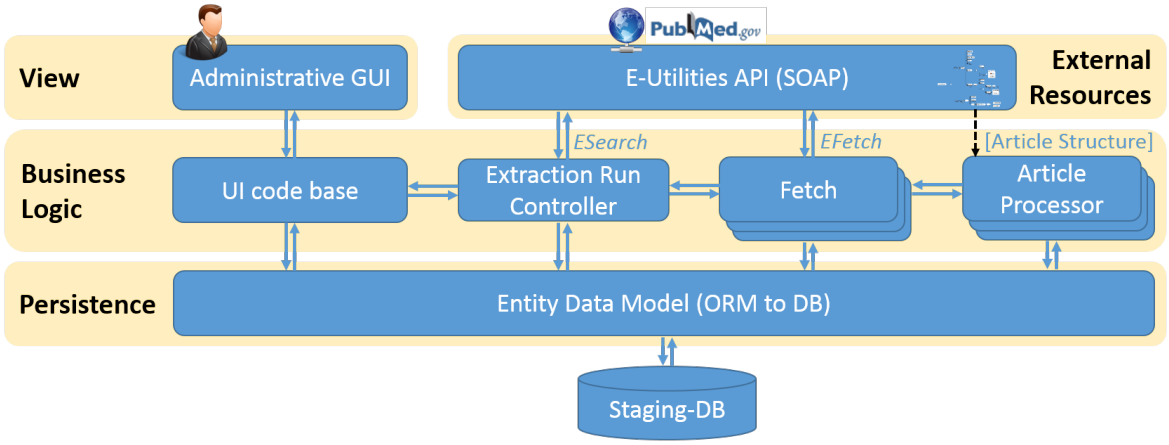

Figure 10: *The overall architecture of the extraction application with its different layers and components. A central* ExtractionRunController*, which can be managed via an administrative user interface governs the process and initiates the communication with the E-Utilities service. Multiple* Fetch *instances are used for loading the article data from PubMed while corresponding* ArticleProcessor *instances extract the relevant data fields according to PubMed's article structure and store the data into the staging-DB.*

The *ExtractionRunController* is the central component of the business logic layer as it governs the extraction process on the highest level. In terms of interaction with other components, it accepts commands from the UI code base and returns possible result or error information. It interacts with the E-Utilities API via the *ESearch* function outlined in section 2.1.2. The results from an *ESearch* request are passed on to one or more *Fetch* instances, depending on the amount of results.

Each *Fetch* component then uses the *EFetch* function to retrieve the article data from PubMed and invokes an *ArticleProcessor* component to extract and store the relevant publication data. The *ArticleProcessor* uses a code-based representation of the XML article structure examined in section 2.1.2 in order to correctly extract and assign specific data fields from the retrieved publication sets.

Interface functions The *ExtractionRunController* class provides three interface functions for starting extraction runs depending on the scope of the search and whether the entire thesaurus or a single term should be processed. The functions and their linkage are illustrated in figure 11. The most general function is to perform a full automatic extraction run for the entire thesaurus (function *performAutoExtractionRun*) which only requires a starting Orpha-number. Using the starting number allows for the extraction run to be resumed at a specific point, i.e. for a specific disease, in case the run had to be stopped, e.g. because of a software error or a server restart for maintenance. The function first queries the *Disease* table of the staging database in order to retrieve all Orpha-numbers that follow the starting number. For each retrieved number, the next, more fine-grained function is called.

The *performExtractionForDisease* function requires the exact Orpha-number for which the extraction run should be performed. As a first step, it retrieves all search terms which are associated with the Orpha-number from the database and are English (attribute *Language*) or have been manually added to the thesaurus (attribute *Origin*). By doing this, the application can confirm that only eligible terms, i.e. those that are likely to bear results when searched for in Pubmed, as outlined in section 2.1.3, are used. The retrieved terms along with the Orpha-number are passed to the next function in line.

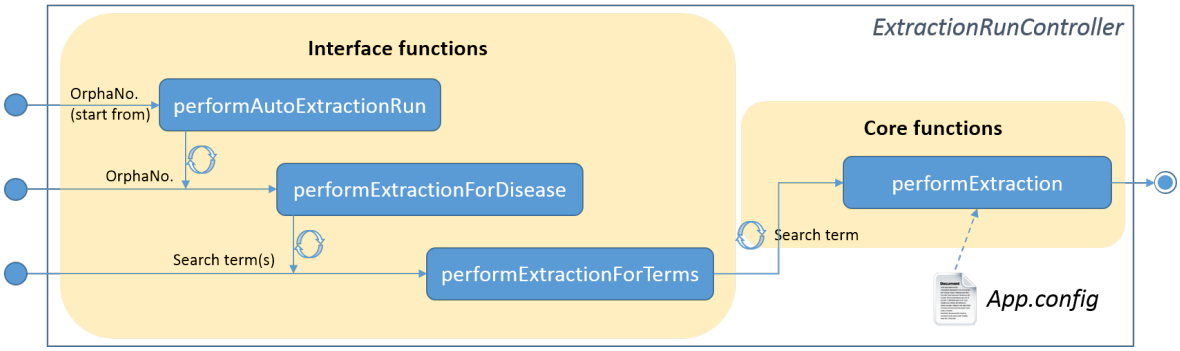

Figure 11: *Three interface functions, built on one another, for starting extraction processes at different levels of granularity as provided by the* ExtractionRunController *class. The core function is always called from the lowest-level method. It dynamically loads the search term configuration from the App.config file.*

At the finest level of granularity, the *performExtractionForTerms* function can be provided with one or more specific terms for a given disease. When used as a stand-

alone function, it proved useful for making ad hoc updates where processing all terms of a disease was not necessary. Examples include recovering from a system error during a run or cases in which it was known that all available publications were covered by a certain term. The function invokes the *performExtraction* core control function which loads the search strategy configuration stored in the *App.config* file, manages the *Fetch* instances and handles the data retrieval as outlined in the following.

Extraction run handling The *performExtraction* function is the core function of the *ExtractionRunController* where the actual search and extraction process takes place regardless of which interface function has been used. It uses the search terms as provided by the interface functions. The first action is to load the search term configuration, i.e. all different search fields that should be used in each PubMed query, as has been laid out in the search strategy devised in section 2.1.3, from the *App.config* file.

For each provided search term, a *SearchRun* entity is created in the staging-DB. An *eSearchRequest* object is provided with all necessary parameters that have been outlined in section 2.1.2 such as db=pubmed, usehistory=y and the search string itself. Initially, the current search term is checked for problematic expressions in order to decide whether to prevent search term interpretation by PubMed as has been explained in section 2.1.3. If the term contains such an expression, it is used within quotation marks. Each single search field that was loaded from the configuration file is appended to the search term with the term being replicated accordingly. All distinct term and search field combinations are connected by OR into a consistent search string. After this parametrisation is complete, the ESearch query is performed and, upon completion, returns a *eSearchResult* object.

The *eSearchResult* provides the WebEnv and Query Key to be used in the subsequent EFetch operation and also contains the number of retrievable results. The results are split into blocks of at most 500 articles each. Attempting to retrieve a higher publication count resulted in an error. The function then iterates through all article blocks and initiates a *Fetch* instance for each block to handle the corresponding *EFetch* call. This is where parallelisation was used. Having a separate *Fetch* instance for each fetch block allows for creating a different thread each thus performing multiple concurrent queries to possibly speed up the extraction process.

Ideally, each fetch block i.e. each *Fetch* instance will have completed successfully. However, several blocks are likely to fail which can be caused by a single article not being retrieved correctly. In this case, the entire block has to be repeated. Additionally, it is not possible to tell which exact article could not be retrieved since the only information pieces available are the starting index and the block size. However, the *ExtractionRunController* does not have information about the index at which a *Fetch* instance failed nor can the possibility be ruled out that multiple articles in one block would fail. Therefore, singling out faulty articles in advance was not possible. Instead, an iterative approach with decreasing block sizes was implemented. Each block that fails to complete is marked for repetition and split up again into smaller blocks of 100, then 50, 10 and finally 1. This process is exemplified in figure 12.

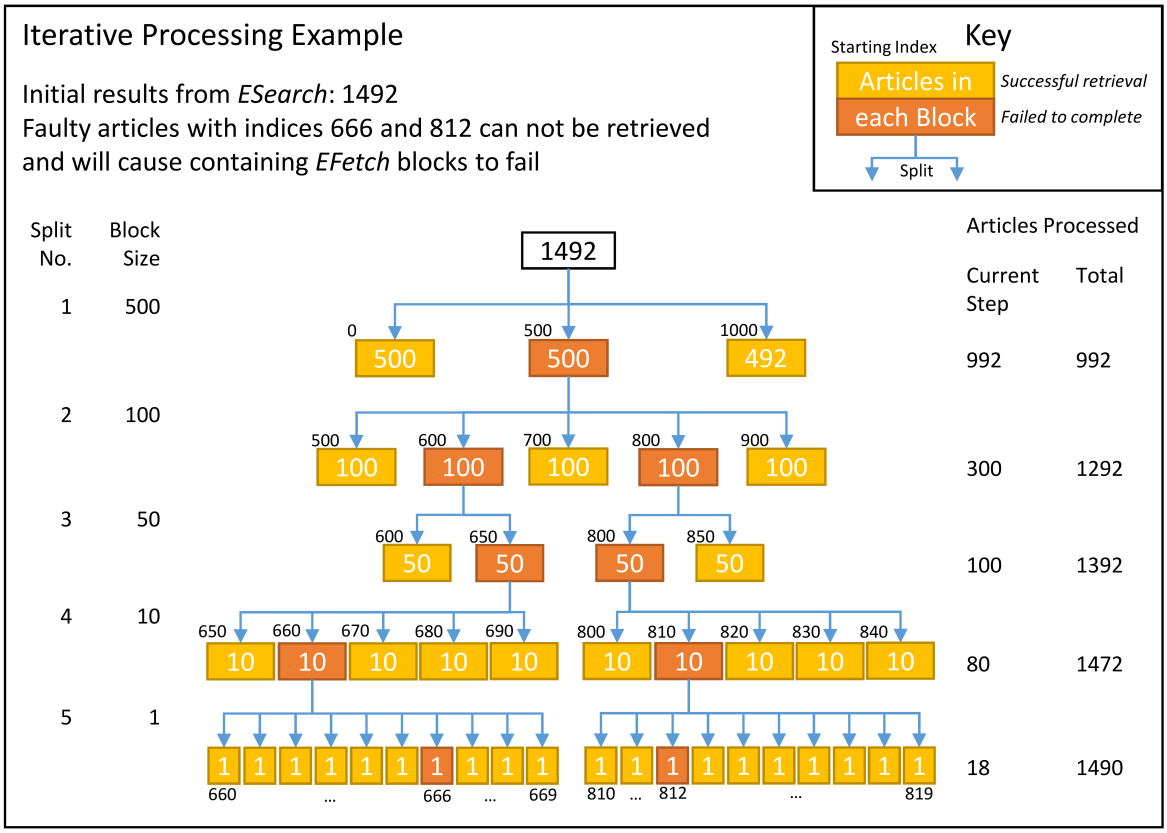

Figure 12: *Example of articles being retrieved using the iterative block-size approach*

Data storage handling Each article is looked up in the staging database on whether it has previously been stored based on the PMID. If that is the case, the currently retrieved article's publication status is compared to that of the previously stored one. A change in the publication status can indicate the availability of new data e.g. the addition of MeSH-headings. In that case, the article is processed as an updatable article. The current data from PubMed is regarded up-to-date and correct and will overwrite any existing data if it should differ. If the status has not changed, the article is only marked as having been found during the current search run in the *ProcessedArticles* table of the staging-DB.

The *ArticleProcessor* handles the step-by-step entry or update of each relevant data field in the staging database. The corresponding article structure to access those fields is provided by the E-Utilities API. Storing data follows an update-or-create principle in order to avoid redundancy. The *ArticleProcessor* first checks for each data item whether it already exists in the database and references it, if applicable. If the data item is not found, it is created first and then referenced. In case of an item not being available from the PubMed article, a default "unknown" entry is loaded from the database and referenced accordingly.

In terms of storing author instances, the last name, fore name, initials and, if available, the affiliation for each entry in the author list are extracted. Corporate authors, i.e. institutions being named as authors instead of persons, are excluded as they have been found to not provide sufficient information about possible contact persons. The *ArticleProcessor* dynamically keeps track of each author's position in the author list in order to correctly assign the first, middle and last author position. The first entry in the list is always designated first author. In publications with two or more authors, the last author position is usually assigned to the last entry in the list. However, if the last entry should be a corporate author, the last author position is attributed to the second-to-last entry i.e. the last real person in the list. Any entries between first and last author are designated as middle authors.

Administrative user interface The interface, which is shown in figure 13, is divided into three major parts: the disease list to the left, the terms list in the middle with interactive elements at the bottom and the protocol area to the right. The disease list is loaded directly from the staging-DB. The list uses the primary Orphanet term

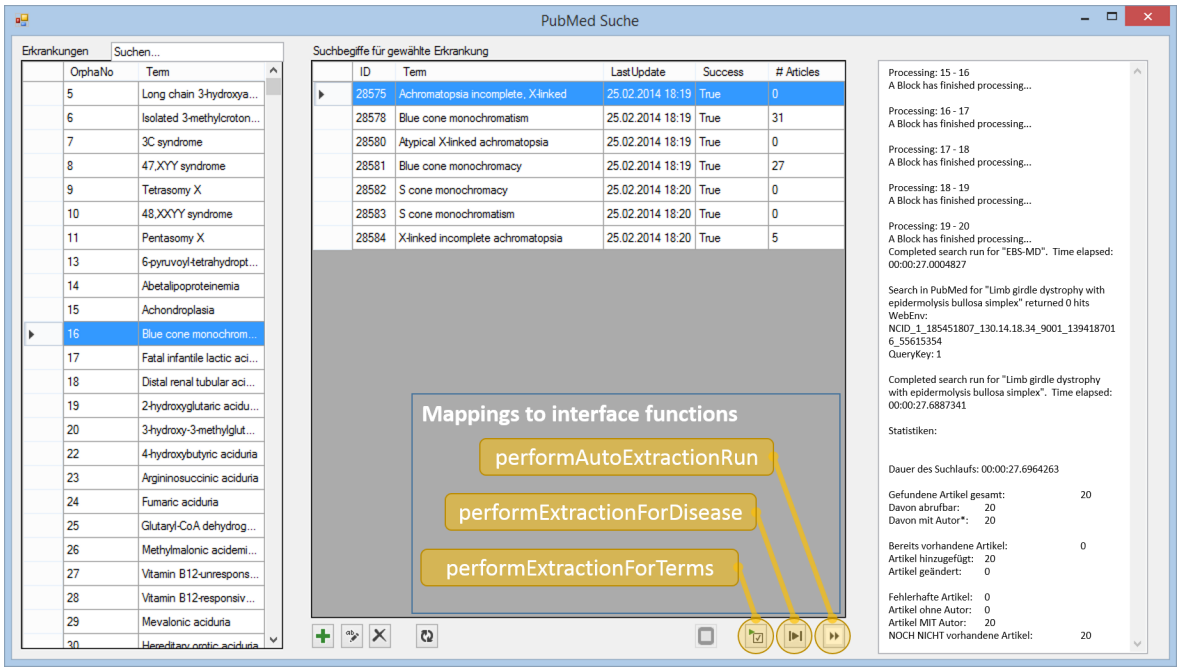

Figure 13: *Graphical user interface for controlling extraction runs and interacting with the search term thesaurus*

to display the disease name. On top of the list, a text field provides basic search functionality to quickly finding a specific disease. Depending on which disorder is selected, the terms list is loaded from the thesaurus.

New terms can be added to the thesaurus and existing terms changed or deleted using the respective buttons on the bottom left. An additional refresh button can be used to load the terms list again in case any changes are not immediately visible. Further to the right, a button to stop an extraction run has been implemented. Regardless of any remaining further search terms that were queued, this button cancels an extraction run after iterative-block processing for the current term is finished. Finally, the three buttons to the right are used to start an extraction run. Each button is mapped to one of the three interface functions of the *ExtractionRunController*. The first one starts a search for those terms which have been selected in the terms list. The second one starts a search for the entire selected disease i.e. all its terms. The third button starts a full automatic extraction run of all diseases starting with the selected disease.

The protocol area to the right serves as an event window that allows the monitoring of an extraction run. It keeps track of events such as the results of *ESearch* requests,

EFetch blocks being processed and extraction runs being completed or cancelled, each with their respective timestamps. At the end of each run, summary statistics that have been contributed by all *Fetch* and *ArticleProcessor* instances and collated by the *ExtractionRunController* are displayed. These contain information about a.o. how many articles were found, retrieved, newly added or changed. Additionally, the completed protocol of an extraction run is saved to a local file.

Extraction results The amount of processed, i.e. retrieved and updated, publications during the extraction runs is shown in figure 14. It can be seen that the vast majority of articles was captured during the first extraction run from February to September 2014. On average, nearly 25.000 articles were processed each day for a total of 3.634.324 publications being stored into the staging-database after processing each available search term from the thesaurus once. From the second extraction run on, where only newly published and updated articles were captured, the numbers drop significantly. Less than 2.000 daily articles and a total of 173.288 were captured during the second run and less than 1.000 per day totalling 37.780 articles in the last run completed in July 2015. More detailed figures are shown in table 5.

Table 5: *Key figures of all conducted extraction runs for retrieving publications from PubMed. The summarised counts for newly captured and updated articles exceed the number of total articles processed as they may include the same article.*

Extraction run	Duration	Average articles per day	Newly captured	Updated
1	7 months	24.723	3.643.883	124.740
2	4 months	1.843	164.586	29.107
3	3 months	712	36.923	1.115
$\sum$			3.845.392	154.962

Overall, 39.150 distinct searches have been successfully performed. For 7.593 of the 16.077 terms in the search term thesaurus, articles could be obtained from PubMed pertaining to 4.208 diseases and including some disease groups. The remaining 8.484 terms, pertaining to 3.374 diseases, did not yield any results. The 3.845.392 publications that have been retrieved in total are of 66 different publication types and involve 17.630 different journals based on their abbreviations, 26.821 different MeSH

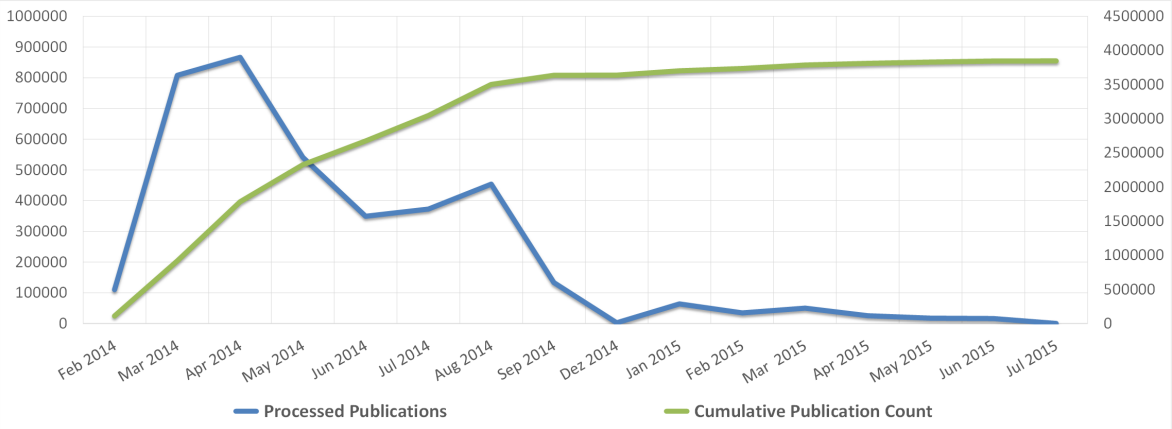

Figure 14: *Monthly and cumulative counts of processed publications over all extraction runs.*

descriptors and all 83 qualifiers. The author instances from all publications add up to 17.034.327 with 1.743.971 distinct affiliations.

3.2 Data processing

3.2.1 Affiliation Analysis

The performance figures of the affiliation segmentation algorithm on 1.000 randomly selected and manually annotated affiliations from PubMed, as presented in section 2.2.1, are shown in table 6.

It can be seen that information on an author's institution is available in most affiliations (99,3%), followed by information on the city (95,2%) which, in some cases,

Table 6: *Performance figures of the Affiliation-Analysis algorithm based on 1000 randomly selected entries.*

	Availability	**Sensitivity**	**Specificity**	**Accuracy**
Department	0,847	0,913	1,000	0,924
Institution	0,993	0,911	0,429	0,908
Address	0,387	0,966	0,644	0,769
City	0,952	0,751	0,813	0,754
Region	0,108	0,380	0,996	0,929
Country	0,866	0,932	0,918	0,930
E-Mail	0,315	0,997	0,997	0,999

could be identified from the institution's name if it had not been noted separately. Other frequent information include the country (86,6%), which could sometimes also be derived from the city, and the department (84,7%). In 38,7% of the affiliations, some kind of address information was available and only 31,5% of all entries contained a valid e-mail address. The least frequent information item was region (10,8%).

The detection rate was mostly very high. Where departmental information was available, it was correctly assigned in 91,3% of cases. Incorrectly assigned information involved cases where the department and institution position were switched in the affiliation string i.e. the institution was named first and the department was falsely identified as the institution. The same effect occurs when an affiliation would only contain a department instead of an institution. In all of the cases where no information on the department was available, the department segment was correctly left empty which leads to a specificity of 100% for the testing data.

Institutions were correctly identified in 91,1% of those cases where an institution was given in the affiliation. Of the seven analysed cases where institutional information was annotated missing because it was either indeed unavailable or unclear, four entries were falsely segmented as an institution which leads to a low specificity (42,9%). These include two entries where only a department was available and the value was assigned to the institution segment. In the other two cases, no "traditional" health care institution was present. These were the Medical Corps of the Israeli Defence Force and the *Healthy Mother/Healthy Child* project of the Egyptian[1] Ministry of Health and Population.

Address information parts, as far as they were available, were mostly correctly assigned (91,1%). The specificity, however, is mediocre (64,4%) as in numerous cases, data has been falsely designated as address information, very often containing institution or city information which had not been identified as such.

City identification has not been optimal with only 75,1% of the available entries having been correctly assigned. Errors include affiliations where cities have been written in a different name than what is available in the respective lookup table or where the city was only part of the institution name and could not be identified as such despite the extended lookup mechanism. In cases where no city was available, various incorrect entries occurred due to parts of country or region names being falsely recognized as a

[1]The country information was not actually provided in the affiliation and had to be researched by the author.

city. Thus, city recognition shows the worst overall accuracy (75,4%).

The identification of regions is rather poor (38,0%). Very often, especially with US states, region names were only available in their abbreviated forms which are not represented in the lookup table. In other cases, the region name has been combined with additional information in the same segment which the algorithm failed to split correctly. The specificity is very high (99,6%) due to the low availability of regional information.

The assignment of countries has worked out well (93,2%) due to a more comprehensive lookup table. In many cases, the country could also be derived from the previously identified city. Still, when country information was provided with unusual spelling, the country could not be recognized. Also, the false identification of some cities led to a false derivation of the country e.g. where England or Republic were recognised as US cities.

Lastly, the e-mail address detection worked exceptionally well with both sensitivity and specificity being at 99,7%. In fact, only a single entry was not identified. However, the address's domain part was incomplete and could actually have been designated as not being a valid address in which case the identification rate of valid e-mail addresses would be 100%.

In summary, it can be said that the algorithm works very well as long as the affiliations do not vary too much from the standard structure and are presented in English or another major European language. In some special cases in which the institution names were kept in their local language, e.g. Finnish or Hungarian, the algorithm does not have the right keywords to detect relevant elements. Other affiliations which could not be processed satisfactorily included ones where the entire author list was part of the affiliation.

3.2.2 Name similarity allocation

Similarity metric comparison Table 7 shows the scores of the different similarity metrics that were evaluated on a reference dataset of 1.186 name allocations as presented in section 2.2.3. For metrics where various different thresholds were examined, only the best result is displayed. It can be seen that the standard Levenshtein 1-Edit-Distance, Metaphone and Jaro-Winkler metrics performed best in both F-Scores while

Table 7: *Performance scores of different similarity metrics for name allocation. For metrics which require a threshold, only the best result is displayed.*

Metric	Threshold	Recall	Precision	F1-Score	F2-Score
Single metrics					
Levenshtein (1-Edit-Distance)	n.a.	0,800	0,727	0,762	0,784
Metaphone	n.a.	0,925	0,673	0,779	0,860
DoubleMetaphone	n.a.	0,831	0,618	0,708	0,777
JaroWinkler	0.87	0,940	0,685	0,792	0,875
Level2JaroWinkler	0.87	0,812	0,627	0,708	0,767
Monge-Elkan	0.85	0,625	0,328	0,431	0,529
Jaccard	0.1	0,258	0,470	0,334	0,284
Soft TFIDF	n.a.	0,713	0,724	0,718	0,715
Levenshtein (Ext.)	n.a.	0,642	0,917	0,755	0,683
Combinations					
Jaro-Winkler $\bigcap$ Levenshtein	0.85	0,773	0,754	0,763	0,769
Jaro-Winkler $\bigcup$ Levenshtein	0.87	0,991	0,652	0,787	0,898
Jaro-Winkler $\bigcap$ Metaphone	0.85	0,902	0,750	0,819	0,867
Jaro-Winkler $\bigcup$ Metaphone	0.9	0,980	0,650	0,782	0,890
Levenshtein $\bigcap$ Metaphone	n.a.	0,744	0,757	0,750	0,747
Levenshtein $\bigcup$ Metaphone	n.a.	0,980	0,657	0,787	0,893
Jaro-Winkler $\bigcup$ Levenshtein (Ext.)	0.87	0,958	0,682	0,797	0,886
Levenshtein (Ext.) $\bigcup$ Metaphone	n.a.	0,971	0,672	0,794	0,892

the extended Levenshtein metric achieved a very high precision (0,917) in single-metric tests.

When looking at the combinations, the results vary with different metrics being in the top three of each F-Score. The highest F1-Scores were achieved by Jaro-Winkler $\bigcap$ Metaphone (0,819), Jaro-Winkler $\bigcup$ Levenshtein (Ext.) (0,797) and Levenshtein (Ext.) $\bigcup$ Metaphone (0,794) which also achieved the third-highest F2-Score (0,892). The highest F2-Scores were achieved by Jaro-Winkler $\bigcup$ Levenshtein (0,898) followed by Levenshtein $\bigcup$ Metaphone (0,893). This comparison constitutes a more detailed analysis of what had also been presented in [59].

Blocking results After finishing the allocation process, a total of 992.629 name blocks were created which contained one or more author instances. There were 167.267 blocks comprising only a single author instance. The maximum number of instances

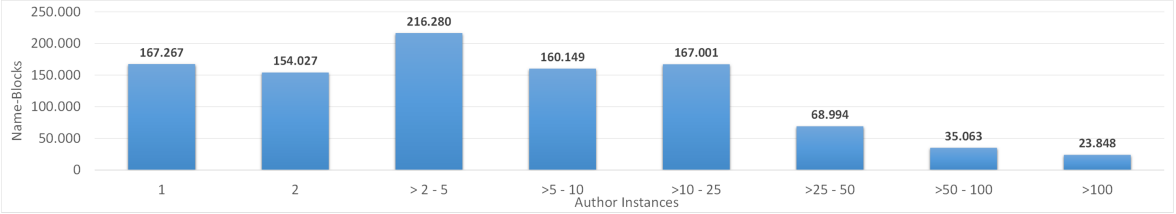

Figure 15: *Author instances per name-block. The X axis shows ranges of author instances contained in a name-block while the Y axis displays how many blocks fall into each range.*

in a single name block was 14.015. Figure 15 shows some more detailed numbers. It can be seen that most name blocks contain up to 25 author instances and that there are comparatively few blocks with more than 100 instances. All blocks which contain two or more author instances, i.e. 825.362, require disambiguation.

3.2.3 Author name disambiguation

The testing dataset for evaluating the disambiguation approach comprised 10.000 name-blocks with a total of 3.677.188 author instances which were disambiguated with varying settings for feature weights and the minGap parameter as outlined in section 2.2.4. Prior to evaluating the disambiguation routine itself, it was examined to what extent the name-blocks, that were created by the similarity allocation process, overlap with the namespaces of the *Author-ity* dataset. The name-blocks match the *Author-ity* namespaces to 99,8996%, meaning that even with the allocation of similar names instead of exact matches, the vast majority of name-blocks is the same as within the *Author-ity* dataset.

The results for varying the *minGap* parameter are shown in table 8. It can be seen that using the Agglomerative Coefficient (AC) according to the first scenario outlined in section 2.2.4, i.e. using stricter conditions for cluster-detection when having more clearly defined cluster structures, results in a poor overlap for approaches 1-3. Using the AC in the opposite way, i.e. using stricter detection conditions the more vague the clustering structure is, resulted in a higher overlap than the baseline approach.

Using the most effective minGap setting, i.e. $1 - AC$, the results for varying the feature weights are shown in table 9. Removing the e-mail feature slightly improves overlap which might be due to the similarity calculation overstating missing e-mails.

Table 8: *Overlap between clusters created by the simplified disambiguation process and the* Author-ity *dataset using different variations of the* minGap *parameter. (AC = Agglomerative Coefficient)*

#	**Variation of *minGap***	**Cluster overlap**
0	Baseline approach	67,3569%
1	minGap = AC	42,6307%
2	minGap = AC^2	41,8486%
3	minGap = AC^3	42,7240%
4	minGap = 1-AC	71,1813%

Table 9: *Overlap between clusters created by the simplified disambiguation process and the* Author-ity *dataset using different feature weights.*

#	**Feature weight variations**	**Details**	**Cluster overlap**
0	Baseline approach	-	67,3569%
1	No changes	Using initial feature weights	71,1813%
2	Remove e-mail feature	E-Mail = 0	71,2295%
3	Higher weighting of last name	Last Name = 2	71,2542%
4	Combining 2 and 3	LastName = 2, E-Mail = 0	71,3062%
5	Remove affiliation features	Department, Institution, City, Country, E-Mail = 0	68,7449%
6	Remove affiliation and MeSH features	Same as above + MeSH = 0	69,0289%

Likewise, putting more weight on the last name feature further improves the overlap by a small amount. The combinations of these two settings remains the best of all examined variations, meaning that the disambiguation process in this form outperforms the baseline approach by only 3,9493%. Thus, its overall effectiveness and benefit remain questionable.

3.3 Deployment

3.3.1 End-user data

After completing the deployment steps presented in section 2.3, the final user-available system is based on a thesaurus of 9.504 distinct diseases and disease groups with a

total of 131.453 terms in seven languages, allowing the system to be used by a diverse user-base. After applying the grouping approach based on exact name matches as described in section 2.2.2, the 17 million single author instances were grouped into 2.999.767 distinct authors for whom profile data was generated.

Figure 16 illustrates the numbers of articles that were captured per disease. It can be seen that for a high number of diseases (1.084) there are ten or less articles available which might indicate that these diseases have only sparsely been researched yet. For another 1.295 diseases total, there are up to 100 articles. Regular numbers then range from a few hundred to up to 5.000 articles while there are only a couple of diseases (271) on which more than 5.000 publications were retrieved. Of these, 15 appear to be very heavily researched and resulted in more than 50.000 publications being returned by PubMed. However, with these very high numbers, the potential of false positives being included has to be considered. There are still 3.374 diseases remaining for which no publications are present in the database. The numbers overall resemble those of a smaller sample presented in [60].

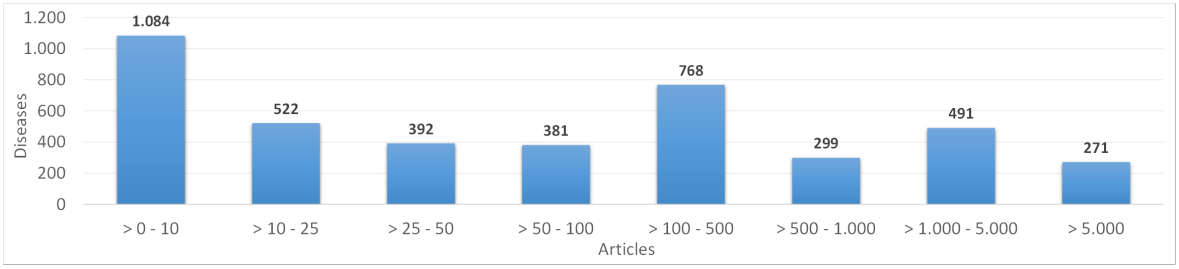

Figure 16: *Retrieved articles per disorders in the final database. The X axis shows different ranges of articles with the number of diseases for which this range of articles is available being displayed on the Y axis.*

The number of authors who publish on single diseases is shown in figure 17. Again, there are a number of diseases with only a few authors, i.e. up to 100 worldwide, publishing on them. For another 1.004, up to 500 authors have published on each disease and 769 diseases have up to 5.000 authors. Lastly, of the 576 diseases with more than 5.000 authors, 58 diseases or disease groups are connected to more than 50.000 authors. This, again, might be partially attributed to false positive results during literature retrieval.

The numbers of articles which authors publish are displayed in figure 18. Most

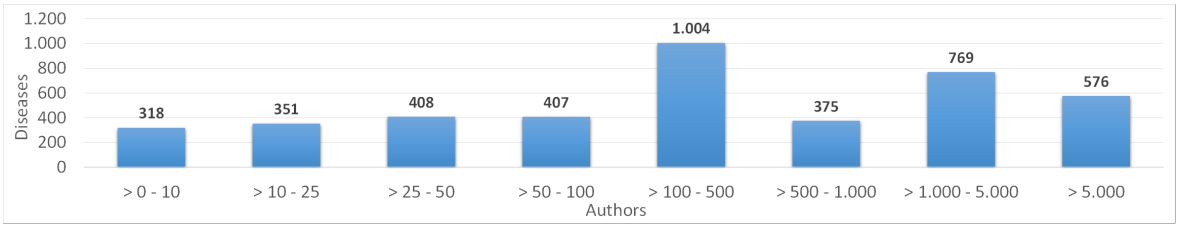

Figure 17: *Numbers of authors publishing on a disease or disease group. The X axis shows, in different numerical ranges, how many authors published on a single disease while the Y axis depicts how many diseases each range is applicable.*

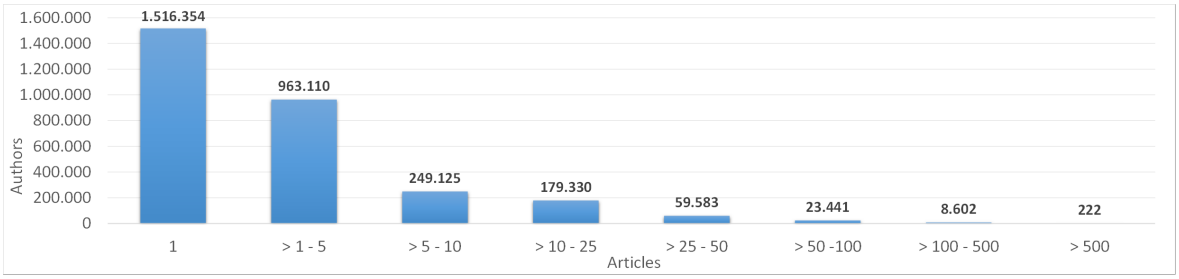

Figure 18: *Numbers of articles published by authors. The X axis shows ranges of publication counts while the Y axis depicts the number of authors whose publication count lies within each range.*

authors (1.516.354) are only recorded in the database with a single publication and another 963.110 authors with up to five articles. Publication counts of up to 500 are quite frequent while there are 222 authors with more than 500. Of those, there are 43 for whom up to 5.000 publications are present. These very high publication counts might be a result of the naive grouping approach which had to be used in the end i.e. the publication counts of a high number of authors with the same name were added up.

Lastly, figure 19 shows the numbers of different diseases or disease groups that single authors publish on. It can be seen that most authors publish on a single or up to five different diseases. Another 642.063 authors total publish on up to 10 and up to 25 diseases respectively with another 76.113 authors being associated with up to 50 diseases. Of the 5.523 authors with more than 100 publications, 34 authors are shown to have published on more than 500 diseases which again might be attributed to name ambiguity issues.

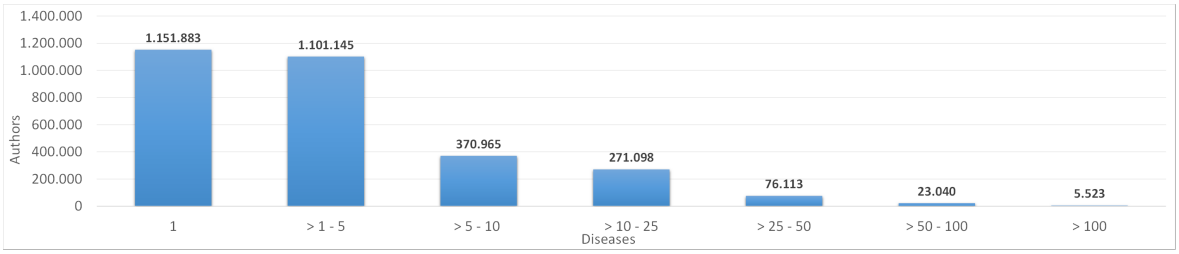

Figure 19: *Numbers of diseases published by authors. The X axis shows the number of diseases which authors publish on. The amount of authors who publish on a given range of diseases is shown on the Y axis.*

3.3.2 End-user client

The web interface for the end-user comprises three major components starting with the disease selection, then showing expert lists as an overview tool and finally providing details in the form of expert profiles.

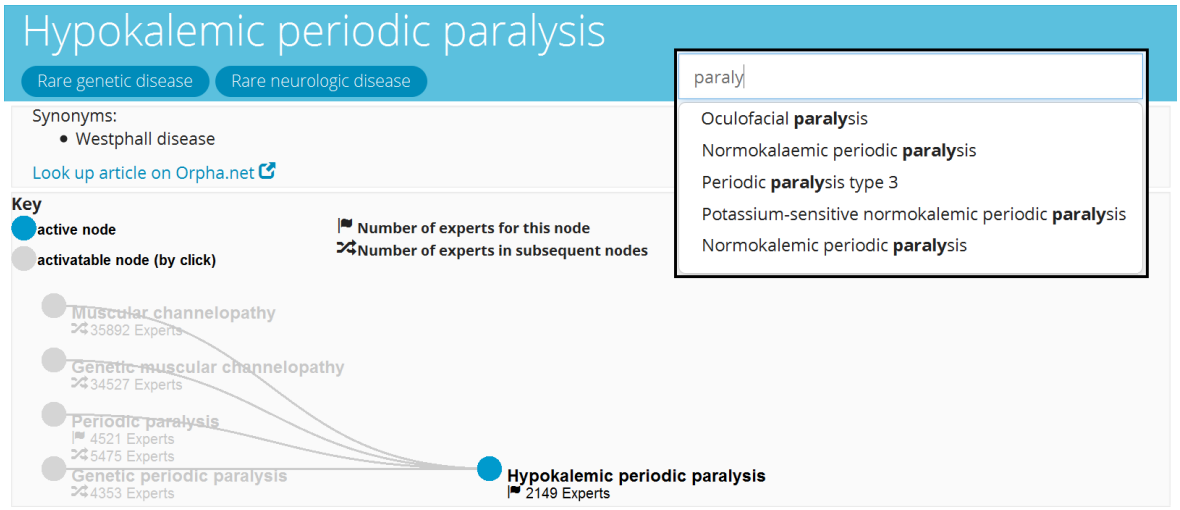

Figure 20: *Hierarchical disorder view and text field for searching disorders (box) in the user client.*

Disease selection The initial step for any user is the selection of the disease for which experts should be displayed. This is done via the text field showcased in the top right of figure 20 which supports word completion for easier handling. After choosing a disease, its position in the Orphanet hierarchy and the involved classifications are displayed which is shown in the remaining part of figure 20. In the case of *Hypokalaemic Periodic*

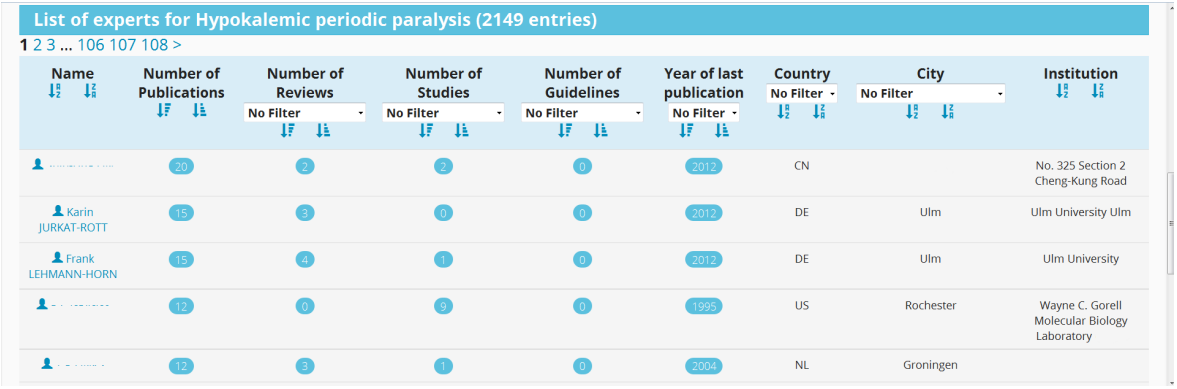

Figure 21: *Expert list for Hypokalaemic Periodic Paralysis in the user client showcasing the data displayed for each author as well as options for filtering and sorting.*

Paralysis (HypoPP), it can be seen that it is part of the Orphanet classifications for rare genetic diseases as well as rare neurologic diseases and has a total of four broader terms as illustrated by the horizontal tree structure. Additionally, the number of available expert profiles are displayed for each node. For those nodes which have narrower terms, the total number of expert profiles that can be found among subordinate nodes is displayed as well. Nodes can be "activated" by the user and the subsequent display will show an expert list for each active node.

Expert lists An example of an expert list for HypoPP is displayed in figure 21. Each row constitutes an author who published on the disease in question and shows the total number of publications, the number of publications of special relevance e.g. reviews, studies or guidelines as well as the author's latest publication year. Basic location information i.e. country, city and institution are shown as far as they are available. The list can be sorted with respect to each column while some columns can additionally be used for filtering. The options for filtering include pre-defined ranges for the number of special publication types, an oldest publication year as well as choosing specific countries or cities in order to allow for users to apply their own criteria as much as possible. Institutions have been excluded from filtering since the current lack of name normalisation for institutions renders this option impractical. A typical filtered list could show experts from Germany with at least one review or more than three studies on a disease whose latest publications are no older than 2015. Clicking on an author name of the expert list leads the user to the detailed author profile.

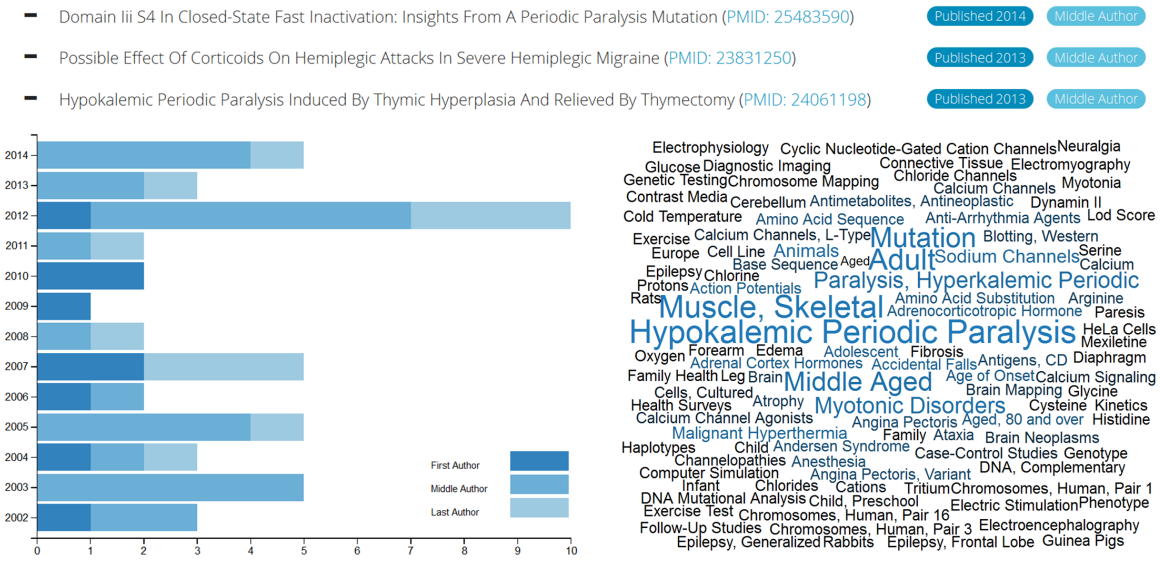

Figure 22: *Different elements of an expert profile in the user client showcasing parts of the publication list (top), the publication timeline (left) as well as a tag cloud of the MeSH terms (right).*

Author profiles The final author profiles are structured into the five sections *Affiliation*, *Journals*, *Articles*, *MeSH Descriptors* and *Authorship*. The Affiliation section provides a list of affiliations that were stated for the author across his or her various publications. From each entry, the year when the affiliation was given is visible and thus allows for a user to see the institutional history of an author. The contact details are provided, as far as they are available from the affiliations, and have undergone the segmentation process presented in section 2.2.1. The *Journals* section comprises a sorted list of the journals in which the author's articles were published. For each journal, the number of articles from the author in question is displayed which is also the primary criterion for sorting the journal list.

In the *Articles* section, the publications on which the profile is based on are listed chronologically. In addition to the article title, a direct link to the publication on the PubMed website is displayed and allows the user to view each publication and all available additional information directly at the source. The publication list in the client also features the publication year and the author's position i.e. as first, middle or last author. An example of such a list is shown at the top part of figure 22. The publication list is additionally illustrated in the *Authorship* section as a timeline with

colour-coded author positions providing a graphical overview of the author's overall publication activity as shown in the left-hand part of figure 22. Finally, the MeSH descriptors of an author are displayed in the identically named section as a tag cloud which emphasises terms hat have frequently been associated with the author. Thus, the user is provided with an overview of the author's main research topics. Such a tag cloud is shown in the right-hand part of figure 22.

As has been outlined in section 1.3, each author profile is available in a disease-specific variant, showing only publication data within the context of the selected disease and a general variant which shows all available information on the author.

3.4 Evaluation

Preliminary proof of concept Figure 23 shows an example of an author profile alongside a matching disorder profile as described in section 2.4. Both were created from the preliminary staging data in order to show the feasibility of the system at an early stage and had also been presented in [60]. The profile of the author shows the three disorders on which the author has published the most articles, which MeSH descriptors and qualifiers have been associated with the author the most across all his publications as well as the journals in which the author has published the most. For the disorder profile of *Congenital dyserythropoietic anaemia* (*CDA*), the authors who have published most articles on it are shown. The remaining data fields are the same as with the author profile, i.e. which MeSH terms are most often associated with the disorder and which journals publish most articles on it.

It can be seen that the author's focus appears to be in the field of haematology as all listed disorders in the profile are haematological disorders with *CDA* being the second most frequently published disorder of the author. The remaining data confirm this appearance as they also contain haematological topics including the MeSH descriptors, of which the disorder in question is second, and journal titles alike.

When looking at the disorder profile, it can be seen that the author is the third most frequently publishing author on the entire disorder. The disorder's third-most frequent MeSH descriptor is *Bone Marrow* with *Bone Marrow Transplantation* being among the author's most frequent research topics. Two out of three of the most frequent MeSH qualifiers that are associated with the disorder are also associated with the

Author Profile - Hermann Heimpel

Top 3 Disorders	#
Chronic myeloid leukemia	44
Congenital dyserythropoietic anemia	29
Acute myeloid leukemia	13

Top 3 MeSH-Descriptors	#
Leukemia, Myelogenous, Chronic, ...	90
Anemia, Dyserythropoietic, Congenital	51
Bone Marrow Transplantation	37

Top 3 MeSH-Qualifiers	#
therapeutic use	115
pathology	95
genetics	75

Top 3 Journals	#
Ann. Hematol.	14
Blood	14
Blut	12

Disorder Profile - CDA

Top 3 Authors	#
... ...	36
... ...	33
Hermann Heimpel	29

	#
Cladribine	727
Erythroblasts	559
Bone Marrow	527

	#
genetics	3732
pathology	2528
blood	2043

	#
Br. J. Haematol.	295
Blood	218
Eur. J. Haematol.	150

Figure 23: *Sample profiles of a specific author and a disorder* - Congenital dyserythropoietic anaemia *(CDA). Matching or similar features in each category of both profiles are highlighted.*

author. Finally, the disorder has mostly been published in haematological journals which partially match the journals displayed for the author.

With this data, the author's expertise and research focus become evident and with the considerable overlap between the author profile's data and the disorder profile, the assumption can be made that the author is an expert on the disorder in question. This particular assertion was confirmed by the author in person [46]. Several other author and disorder profiles have been created and examined. While there are no quantitative results available, the qualitative evaluation showed similar matches between a majority of examined profiles. In a minority of cases, the author profiles would not reflect the author's actual expertise.

Table 10: *Confirmed clinical and theoretical experts from expert-annotated author lists.*

	List 1	**List 2**
Annotated entries	49	19
Confirmed clinical experts	13	10
Confirmed theoretical experts	2	4
Unconfirmed authors	34	5
Entries with errors	5	4

Expert-annotated author lists Of the five author lists that were sent to clinical experts for annotation, only two were returned. Table 10 shows the annotation results. Those authors who the experts saw as appropriate experts with contact to patients are listed as *Confirmed clinical experts*. *Confirmed theoretical experts* denotes authors who were annotated as having expertise on the disorder but won't see patients. These include basic researchers or pathologists amongst others. The term *Unconfirmed authors* includes all entries which were not annotated by the experts but were still listed above the last annotated entry. Finally, erroneous entries include authors who were annotated as not being specific experts on the disorder in question, whose contact details were obsolete or who were no longer considered experts because either their research dates too far back or they were already retired.

Feedback from the Orphanet encyclopedia team has been very positive. On all five lists that were provided to them, the experts they (Orphanet) had expected to appear, were indeed present. This has later been confirmed during a demonstration of an early version of the user client where searches for additional different disorders had the same positive results.

Comparative evaluation based on verified expert-lists The results for comparing the system to two other existing approaches on the basis of verified expert lists are shown in table 11. For each disease, the number of verified experts is shown along with how many of those are German experts. This distinction had to be made since one of the registers that were chosen for comparison (se-atlas) only covers German experts while Exertscape as well as the bibliometric system of this work are designed for worldwide use. Some of the diseases could not be found in the two other registries. The corresponding entries are marked by a '*'.

Table 11: *Evaluation of the system against two other expert finding approaches on the basis of verified expert lists. The sample classes of the diseases pertain to the classes described in section 2.4. Verified experts are authors who should definitely be included in an expert repository for each disease. The values in parentheses show how many of the provided experts were German since se-atlas can inherently only cover those. The remaining columns show, how many of the verified experts could indeed be found with each approach.*

Disease	Verified experts (German)	Experts found in the system	Experts found in se-atlas	Experts found in Expert-scape
Sample class 1				
Amyotrophic lateral sclerosis	4 (3)	4	0	3
Creutzfeldt-Jakob disease	4 (2)	4	2	4
Behçet disease	3 (1)	3	0	2
Nevus of Ito	4 (2)	4	0*	0*
Neuromyelitis optica	4 (1)	1	0	0
Mucoviscidosis	5 (1)	3	1	3
Sample class 2				
Niemann-Pick disease type C	8 (5)	1	2	0
Legius syndrome	5 (4)	0	0*	0*
Branchio-oculo-facial syndrome	5 (2)	0	0	0*
Central core disease	4 (1)	1	0	1
Primary familial polycythemia	6 (2)	1	0	0*
Sporadic pheochromocytoma	5 (1)	1	0	0
Sample class 3				
Osteogenesis imperfecta	5 (1)	3	0	0
Periodic paralysis	5 (1)	2	0	0
Congenital dyserythropoietic anemia	4 (1)	3	1	2
Nemaline myopathy	7 (1)	0	0	0
Cushing syndrome	6 (3)	6	0	4
Beta-thalassemia	4 (1)	2	1	1
$\sum$	88 (33)	39	7	20

* the disease could not be found in the respective register, thus no experts are available.

A total of 88 verified experts for 18 different diseases and disease groups were searched in each system. Of those experts, 39 can be found within the current system pertaining to a success rate of 44%. Using se-atlas, of the 33 possible German experts only seven are featured (21%) with two diseases not being listed in their system at all. Within Expertscape, 20 of 88 experts can be found (23%) with four diseases being unknown to their system.

In-depth-evaluation with Orphanet The author list on which Orphanet Germany was able to provide feedback contained a total of 153 authors who published on the disorder in question and could be located in Germany based on their affiliations. Of those, 39 were clinical experts who were already registered in the Orphanet expert database. Upon applying specific criteria and manually verifying the results, Orphanet was able to identify three new clinical experts from the author list who were previously unknown to them [69].

4 Discussion

Expert finding via bibliometric analysis The overall approach of finding experts from literature data is a technique that has been shown to work in various areas such as company or research environments and has also been employed in a medical expert finder system (*Expertscape*). However, this kind of bibliometric analysis entails two main questions: "Are really all experts identified by this approach?" and "Are the identified experts really experts?". In terms of the first question it can be stated that the bibliometric approach will inherently only identify those - and ideally all - experts who publish on one or more specific rare diseases. The system is, by design, blind to experts without any publication activity and currently also to those whose publication records are not available from PubMed. Given the rarity of the diseases in question and the necessity for researching them, it could be argued that expertise on rare diseases can not go without research, and thus, publication activities. Therefore, the number of true experts who do not publish anything should be negligible. However, there are also points against this reasoning, e.g. very practice-oriented experts such as surgeons who might have been involved in the treatment of rare diseases but were not part of any publications. Thus, a future expansion of the system towards looking to identify experts who do not publish might be beneficial.

The second question pertains to estimating the individual author's expertise. Ideally, a single indicator would be available to gauge a researcher's quality, which, however, is an unrealistic thought according to Friedemann [56], as neither the number of publications, citations or any impact factor are a definitive guarantee for high expertise. Estimating expertise requires looking at all available data by people who, ideally, have profound knowledge of the respective domains themselves. Therefore, instead of using any indices, the end-user client is restricted to displaying the available data and leaves any estimation of expertise to the user. Thus, certainly not every author who is displayed by the system will be an expert. Instead, the number of authors who cannot be considered experts on certain diseases is likely to be rather high, as can be seen from the large amount of authors with a single publication presented in section 3.3.1 and also became evident from using the system thus far.

Similar to this second question, it could additionally be asked whether identified experts are actually experts for the diseases that they have been associated with due

to their publications. This can only be the case if all retrieved publications are indeed about the disease in the context of which they were retrieved. It can be stated in advance to discussing the search strategy that this has not been the case. Therefore, there are experts in the system who are associated with certain diseases via their publications who are not actually experts on these diseases. This may also be due to co-authorships or maybe honorary authorships.

This last train of thought brings fourth a final question: will the identified experts be of any use to those who search for them e.g. a patient? That is, will an expert actually see patients or be able to provide adequate counsel to a peer medical expert? Again, this might not be the case for many authors who are identified as experts by the system and it will come down to a case-by-case basis. It is then up to the user to further explore an author's data using additional search possibilities such as Google. Still, some precautionary measure could be taken here by setting up the author profiles in a way that lets users better assess an author within the system. The MeSH keyword cloud in each profile as well as the author's journals may already give an indication as to whether an author is clinically active or predominantly a basic researcher. Ideally, this distinction could already be determined and presented as part of an author profile.

PubMed as literature data source PubMed with its vast collection of literature data being openly available for queries proved to be an excellent starting point for building this system. While it is feasible to assume that the largest part of articles available on rare diseases has been covered by PubMed, there most likely are more relevant documents from other sources. It had been a principle of this work to utilise only openly available data for creating the information. Yet, in order to achieve an exhaustive and comprehensive corpus of information, sources which are not available in this way might have to be examined and included. This, of course, involves extending the present extraction application to enable the data extraction from new sources as well as extending the data processing steps in order to integrate the new data into the system's information structures.

Additional sources beyond PubMed which have been suggested for examination include other literature databases or search engines e.g. *PSYNDEX [8]* or *Google Scholar* [5], evaluation portals such as *Jameda* [6], or the so-called *White-list [7]* as well as the mandatory quality reports of university hospitals. Retrieving information

about recent or ongoing clinical trials e.g. from [4], might also be helpful. Several of these additional sources are rather local, kept in languages other than English, and may not necessarily be freely accessible. However, if it were possible to obtain information with respect to rare diseases, these sources might still contribute to the overall data coverage.

Rare diseases thesaurus and search strategy Having a comprehensive thesaurus of rare disease terms is an important part for both retrieving literature data in the first place and structuring the system's knowledge representation to the end-user. The choice was made to use the Orphanet classification of rare diseases as it should cover the vast majority of rare disease designations and is actively being maintained and extended. There might, however, still be some diseases which are not yet covered by the Orphanet classification [70] e.g. very recent subtypes of known rare diseases. These would have to be considered as well in order to have an optimal basis for rare diseases information retrieval. The system already provides for these cases with the possibility of manually adding new disease entities via the administrative user interface.

For the most part, only the end nodes of the classification i.e. those with no further subordinate terms have been used in searching for literature. Still, the superordinate terms i.e. disease groups might also be prevalent in publications and for those terms, data should be retrieved as well. This is, however, not trivial as the levels above which the terms are too unspecific to accurately depict expertise vary greatly across all diseases and classification branches. The border between a specific disorder entity and a disease category is rather fuzzy. Deciding which group terms are used and which are not might have to be made individually and manually which would be quite tedious. It also might re-introduce the issue, which has been described in section 1.1, of having "experts" on disease categories which are too general i.e. there are too many subgroups of diseases resulting in a lack of specificity.

Of the different search strategies, i.e. what search fields in PubMed are utilised in combination with the disease terms as outlined in section 2.1.3, the least strict has been chosen. Usually, the accuracy of other, more specific, search goals can be measured against a reference set of papers. Such a set allows for comparative metrics such as sensitivity and specificity. Unfortunately, no sufficient amount of reference literature was available to gauge the accuracy of the searches for this project. Finding a

strategy therefore had to rely on manual, sample-based checks and comparisons of the amount of returned results. Thus, an informed decision on a stricter strategy was not possible. However, most false positives appear to originate from other issues than the choice of search fields. It has been regarded feasible that publications that contain disorder names in their title, abstract, expert-assigned MeSH headings or author-provided keywords are sufficiently close to those disorders so that their authors can be inferred to be experts. Yet, the presence of publications that have been incorrectly associated with certain diseases introduces some doubts about that assumption. In any case, another examination of possible search strategies which is based on sufficient reference literature sets is advisable. Such literature sets should be provided or validated by multiple experts on different rare diseases although latest review articles could already provide a provisional solution.

With 3.374 diseases for which no data could be retrieved, examinations as to what can be done to fill those information gaps are necessary as well. One example for such a disease is *Primary interstitial lung disease specific to childhood due to pulmonary surfactant protein anomalies*. Queries for this term did not yield any results neither with the specified search strategy nor after providing only the disease name without specifying any search fields as of July 2016. PubMed may, however, meanwhile return results for other diseases contrary to earlier extraction runs. Thus, updates from a newly conducted extraction run might reduce the number of diseases for which the system does not have any data.

Extraction application The extraction application had been designed to extract a large amount of literature data and store it for further processing. This worked very well in the beginning of the project but it became increasingly rare that entire blocks of 500 articles could be processed at once. In most extraction runs, the iterative processing algorithm then had to decrease the block-size to one for nearly every block. The cause of this could not be tracked down but it poses the question whether the algorithm should be kept like it is or whether changes should be made to return to the initial extraction efficiency.

However, this issue is of secondary nature as the number of new literature for which data is to be retrieved is significantly smaller than the initial indexing as has been shown in section 3.1.4. It is therefore more important to resume the data retrieval

at all which means adapting the extraction application to the reworked interfaces of the PubMed API since the SOAP interface support was cancelled in 2015. This issue also negatively influences the current usability of the system as the latest publications on rare diseases can not yet be discovered. Current data would otherwise constitute one of the major strengths of the system.

Process automation In terms of automation of the system, a one-click-update routine would have been the ultimate goal, meaning that all extraction and processing steps for updating the system's data could be executed in succession without any further intervention from a human administrator. This is not easily done in the current state of the project as there are still a number of different necessary interventions, e.g. in case of errors, as well as development prerequisites that have to be met first, e.g. the adaptation of the extraction application. However, the overall system architecture resembles a pipes and filters structure with each of the various processing steps being a filter and the different databases acting as pipes. It is therefore quite feasible to develop these components to a point where they can be controlled and executed by a central workflow in order to achieve full automation.

Affiliation analysis The results presented in section 3.2.1 indicate that the segmentation and analysis algorithm works well for the most part. There are some obvious false entries especially where cities do not get detected or another data item is misidentified as a city, often also resulting in a false country identification. The results were also poor where affiliations were present in a language that is not present in the keyword list for institution detection. There is potential for improvement to both the algorithm itself as well as the underlying lookup tables and keyword lists. For further enhancing this processing step, it might be necessary to introduce new elements such as normalising institution names. Institution detection could overall be improved, potentially with the help of semantic web applications which specialise on entity recognition. An advanced analysis could also recognise the type of institution i.e. clinic, laboratory, research institute, to help in further classifying authors.

Name similarity allocation Most of the similarity metrics that are frequently used in matching tasks were compared to each other on the basis of a manually annotated

reference dataset with a broad selection of random names from the database. The results might be improved if another set of annotated data would have been used for additional verification. This, however, was impeded by resource constraints and the results with respect to the overall system are satisfactory.

When looking at the top three comparison results for metric combinations in both F1 and F2 scores, a total of four combinations with scores of up to 0,9 were presented in section 3.2.2. In principle all of these four combinations should be eligible to be used for performing the similarity-based name grouping. Objectively, Levenshtein (Ext.) $\bigcup$ Metaphone may be the primary choice as it is the metric that scores among the top three in both scores (F1 = 0,794 and F2 = 0,892). However, since the initial allocation scheme for the reference list prior to manual annotation was based on the DoubleMetaphone algorithm, this may have introduced some bias i.e. overly well-fitted results for Metaphone. For this reason, different metrics were preferred to those involving Metaphone. Additionally, it is uncertain to what extent recall is really to be regarded higher than precision and so the decision has been made to choose Jaro-Winkler $\bigcup$ Levenshtein (Ext.) for name grouping as it boosts precision while still retaining a very high recall (>0.95).

Examined disambiguation approach The three-step disambiguation method was chosen based on what is often used in the literature and especially within the context of PubMed. Grouping names based on their similarity instead of exact matches has not been done in the exemplary approaches. It was, however, predicted by Torvik and Smalheiser [80] that the difference would be negligible. The results which are presented in section 3.2.3 seem to confirm that prediction as the similarity-based grouping of this work produced almost the exact same name-blocks as are present in the *Authority* data. While name-grouping remains essential to these kinds of disambiguation approaches, performing it on the basis of similarities might not be a necessity.

Developing, employing and exhaustively evaluating a fully sophisticated disambiguation approach would likely have been a project of similar magnitude as this entire work and, instead, a simplified process was created. Using *R* seemed like a good opportunity as it offers all necessary packages for recreating the major disambiguation steps of the exemplary approaches, i.e. weighted dissimilarity computation and hierarchical clustering. Additionally, the *DynamicTreeCut* package presented the prospect of fully

automating the cluster detection without having to find a probabilistic model for the available data. However, there were several issues starting with a lack in performance of the *R* process for bigger namespaces with several thousand author instances, so it is doubtful that the entirety of name-blocks could be efficiently processed. The overall results of the disambiguation to this point in time do not indicate a significant improvement over the simpler grouping approach based on exact name matches. This is especially true with respect to the considerable computational costs of the disambiguation process.

More evaluation steps would have to be done with more variations of feature weights and the use of the dynamic tree-cut parameters in order to find a better setting that would render the disambiguation approach more useful. On the other hand, this could confirm that there might be a systematic error which renders this overall approach unusable. Other evaluation data than the *Author-ity* dataset may also be used for a more comprehensive gold-standard since *Author-ity* itself is the result of a disambiguation approach.

Using the naive grouping scheme as described in section 2.2.2 withholds an important advantage of the system as aggregate data in the expert profiles may underlie considerable inaccuracies due to name ambiguity. However, these issues might be mitigated to some extent by the lower number of researchers for rare diseases compared to more common medical research fields. This could potentially reduce the chance that two publications with identical author names about the same rare disease were actually written by different persons.

Author profiles The constructed author profiles should allow for a detailed estimation of an author's expertise by a user of the system. There are, however, a few issues with their correctness and currency. As the evaluation showed, many profiles are not up to date and the authors may no longer be present at the institution which is stated in the latest available affiliation. This is also due to the current inability of the system to retrieve new literature. In general, information on authors can only be as up-to-date as their latest publication. Another severe currency issue is with authors who are rather recently retired or deceased. These authors may still have relatively current publications but are no longer available for consultation. This is something that the system cannot possibly detect and has to be dealt with via manual research by users.

Additional errors may be introduced by flaws of the affiliation segmentation algorithm which e.g. might not represent the institution name accurately. This, however, is not as big of an issue as the institution can, in most cases, still be well identified by human users. Still, there is generally a lot of room for improving the author profiles in terms of their content by adding new information and refining existing data. Multiple occurrences of the same institution could be avoided by normalising institution names. More current contact information could be gathered by web crawling processes being employed on top of the affiliation segmentation. Though controversial, it has been suggested that a number of indices such as the H-index or impact factor could also be added to the author profiles and expert lists in order to provide a more comprehensive overview. Additionally, the distinction between clinical and theoretical experts should be examined and, if available, added to the profiles. Lastly, a number of functional improvements and additions were suggested, e.g. the possibility to exclude publications from an expert list where the authors are in a middle position of the author list [68], giving more weight to first and last authorships. Such a filter could then be further extended e.g. to also exclude last authorships. There are certainly many more options to explore.

Concerns were raised about direct contact details of experts such as e-mail addresses being openly available to a large number of users. Precautions could be made to restrict the displayed contact information to those of the author's respective institutions if possible.

Data protection questions While it has not yet been addressed within the scope of this thesis, there are a few data protection and data privacy issues to be considered for this project. All data that is used by the system is extracted from either PubMed or Orphanet under free license. Additionally, the data are currently coming from openly available sources, i.e. everybody could conduct the same searches and processing steps to receive the same results as were presented here. Still, the author names constitute personal information and providing them to the outside has to be handled with the necessary care and precaution that is in agreement with data protection and privacy laws.

According to [66], the system is covered by a research privilege during development and testing, i.e. using the data within the context of academic research, is permitted.

Should the system at some point be made available to a broader audience beyond a small circle of domain-specific users, these points might have to be re-examined. Contact information of authors should by then be made available for correction by these authors who themselves, if possible i.e. if their e-mail address is known, should be notified and offered some kind of opt-out possibility. With evolving EU-legislation it may become mandatory to inform every author captured by the system about the intended usage of their data and to make sure that no personal information can be obtained via the system without explicit human interaction. By then, the feasibility of the system in its entirety may have to be reconsidered.

Target user group This again raises the question of who should actually be the target user group of the system. While it has originally been intended to provide access to the system to the public in order to allow each patient, their physicians and caretakers to find and contact the experts they find most suitable for their disease, this goal has not yet been reached. In its current state, using the system requires a certain amount of interpretation when examining its author profiles and possibly estimate their expertise. Therefore, the system's focus and target group have changed and it should for the time being only be provided to a restricted circle of expert users i.e. guides in centres for rare diseases or public expert registries. Ideally, further development will enable the system eventually to be available to the general public.

Evaluation The possibilities to evaluate the system have not been ideal as evaluation opportunities in general were restricted due to lack of reliable source information about definitive experts on various diseases. Four principal evaluation steps were conducted, each targeted at different aspects. However, the results of all steps might suffer from only being conducted on a very small scale. Where external feedback was necessary, the sparse response from experts constituted another impediment. The preliminary proof of concept checks suggested that the system is well able to depict authors and their expertise on different diseases and, vice versa, present the relevant experts for a large number of diseases. The expert-annotated author lists relativise these findings as many experts were indeed to be found on these lists but there was also a large number of authors which were not regarded appropriate experts by the annotators.

However, the comparison to the other registries on the basis of verified expert lists

has shown that the system is quite capable of outperforming those approaches. While the 44% success rate of the system is not as high as has been anticipated, it is twice as high as either of the compared registers. With *Expertscape*, it could often be observed that diseases could not be found due to their reliance on MeSH and the resulting lack of specificity towards rare diseases. The results for *se-atlas,* even if restricted to German experts, were lacking and show how many experts can be left out by a manual registration process. One flaw of this comparison might be that it only looked at if the experts were found, not if their data was correct. It is possible that manually curated registries have more accurate data on experts. However, it can also be argued that knowing who the experts are and then finding their correct contact details is still more beneficial than not knowing about those experts in the first place. Again, this evaluation should be repeated on a larger scale, if possible.

Determining which of the known experts for a disease are also presented by the bibliometric system has been less of an issue than determining whether those authors that were found really are experts and if there are any previously unknown experts among them. For the single disease that has been checked for new experts by Orphanet Germany, three new experts were found in addition to 39 who were already known. These numbers suggest that there is quite a lot of potential in the system for identifying new experts, provided that the resources to actually look for them are available and utilised. These efforts should be reinforced and extended as this probably constitutes the most important benefit of the system.

Conclusion

With respect to the research question posed in section 1.3, it can be stated with some confidence that an expert finder system on the basis of bibliometric analyses as developed in this work is indeed able to satisfy information demands on rare disease experts to a certain degree. Literature data for 64.5% of all rare diseases that are present in the system could be retrieved and expert profiles for nearly three million authors have been created, albeit with a simplistic grouping approach that still suffers from name ambiguity. The system features an interactive diseases selection and various options for filtering and sorting expert lists. The expert profiles allow for the estimation of the author's expertise though it may require some domain knowledge and the

profile data may not always be up-to-date and correct. Notwithstanding the persisting issues, the system can, at this point, already be successfully used by domain experts to identify experts on rare diseases both those who are already known to other expert registries as well as additional ones who were previously unknown to those registries. This assertion is reinforced by the interest, Orphanet has taken in the project [62].

The system is also able to surmount the drawbacks of comparable approaches. Using the Orphanet classification to target rare disease literature, the international orientation of the bibliometric approach and the additional in-depth analysis of the extracted metadata were shown to improve results for identifying experts over other registries. As such, the system can be considered a valuable complement to expert finding efforts in support of rare disease patients.

Outlook

The system has a lot of potential and also need for improvement in order to be usable in a sustained manner and by a wider audience. As has been outlined, further evaluation should be conducted on a larger scale than what has been possible up to this point, especially with respect to how new experts can be identified effectively. The system requires further development and adaptation to new developments concerning the PubMed API and eventually to obtaining data from additional sources. Also, further internationalisation might have to take place since the system features data from publications worldwide. Users who do not want to be restricted to English or a European language could, in the future, be another target group which will necessitate the inclusion of additional languages within the user-client.

Further improvements can be made to the author profiles in terms of data quality and content. Apart from complementing or correcting contact data of experts, categorising the profiles has been a highly requested feature. Two types of information might be added here: firstly, a distinction between clinical experts and those who are mainly involved in basic research and secondly, an estimation of the primary medical discipline in order to enable an easier estimation of an author's expertise. While the former requires more research, examinations on the latter feature have been conducted where it was shown that the summarised MeSH keywords of authors can be used to effectively determine their primary discipline. However, more research and develop-

ment is necessary to include this feature as well. Other improvements are connected to better display, filter and sort options such as selecting experts based on their proximity to a user's location.

Lastly, in order to keep the system's data current, new update methods need to be developed as the current process is more oriented towards processing the entire data pool. With the rather small amount of new publications being added compared to what has already been processed, it is not feasible to perform the processing steps for the entire database again. Instead, different mechanisms for handling new data will be required in the future, if the system should become a sustainable tool for expert finding and quality assessment.

This particular approach of an expert finder system could, in principle, also be applied to a variety of other fields besides rare diseases. Where experts are sought and there is sufficient publication activity to base such a system on, the methods presented in this work could be adopted accordingly. A search term thesaurus with relevant terms from the respective field would have to be set up and the data structures might need to be adjusted to varying publication metadata. However, an effective disambiguation process may become even more important in other fields where the mitigation of name ambiguity due to the rarity of researchers might not be as prevalent as with rare diseases.

5 Summary

Identifying experts on rare diseases is a key element in improving the situation of patients and health care providers alike. Information on rare disease experts is being provided by a number of online portals such as Orphanet, se-atlas and also Expertscape. These are, however, mostly manually maintained and updated which causes issues with the specificity, completeness and currency of their data. In the case of Expertscape, data is being collected via bibliometric analysis from scientific literature, i.e. authors of publications on diseases are presented as potential experts for those diseases. This approach mitigates some of the aforementioned issues, it however introduces a new issue of name ambiguity, whereby publications from multiple authors sharing the same name are not being discerned. In addition, the orientation towards common diseases lets Expertscape miss many rare diseases.

It therefore was the goal of this work to develop an expert finder system on the basis of bibliometric analysis which is specifically oriented towards rare diseases and employs in-depth analysis techniques as well as author name disambiguation. The system should overcome the issues of traditional expert registries and complement the availability of data on rare disease experts.

A thesaurus of rare disease terms was set up using the Orphanet classification of rare diseases. With these terms, PubMed, a biomedical literature database with more than 26 million citations, was searched for publications on rare diseases and the respective metadata extracted. An application which manages the literature extraction in an iterative way as well as a staging database for storing the extracted data were set up. The application is also used in the maintenance of the rare diseases thesaurus.

The extracted data was processed in terms of an in-depth analysis pertaining to segmenting affiliation entries in order to get additional structured information about institutions, cities and countries among others. Further analyses to make full use of e.g. keywords were examined but not fully realised. Author name disambiguation was extensively examined. Grouping name instances to reduce computational complexity prior to performing the disambiguation was based on name similarities instead of exact matches. Multiple similarity metrics were compared against each other and the best suitable matching scheme was employed. A disambiguation approach using pairwise similarity calculation, hierarchical agglomerative clustering and a dynamic

cluster-detection method was implemented. The approach was evaluated against a sophisticated disambiguation approach as well as against a baseline naive grouping based on exact name matches. The disambiguation performance was unconvincing and for the first prototype, the naive grouping scheme was retained including the resulting ambiguity issues.

The overall system was evaluated in different ways, such as having experts annotate author lists manually as well as comparing the system to other expert finding approaches on the basis of verified expert lists. The comparison showed that the system is able to identify more rare disease experts than the other approaches, however, there are numerous false positive entries to be found and data on experts also suffer from a partial lack of correctness and currency.

Still, the bibliometric analysis system is, in conclusion, a successful proof of principle and can already be used in complementing rare disease expert registries by identifying previously unknown experts. Persisting data flaws would need to be sorted out in the future along with further adaptation to changing external conditions such as access possibilities to source databases. More analysis features could be examined and realised in order to enhance the system and provide for its sustainability.

6 References

[1] Nationaler Aktionsplan für Menschen mit Seltenen Erkrankungen: Handlungsfelder, Empfehlungen und Maßnahmenvorschläge

[2] Orphanet - 2015 Activity report: Orphanet Report Series: Reports Collection

[3] Orphadata: Free access data from Orphanet. © INSERM 1997 (Last accessed on 06.07.2016)
Available from `http://www.orphadata.org`

[4] ClinicalTrials.gov: A service of the U.S. National Institutes of Health (Last accessed on 09.07.2016)
Available from `https://www.clinicaltrials.gov/`

[5] Google Scholar (Last accessed on 09.07.2016)
Available from `https://scholar.google.de/`

[6] jameda.de - Deutschlands größte Arztempfehlung (Last accessed on 09.07.2016)
Available from `http://www.jameda.de`

[7] Wegweiser im Gesundheitswesen | Weisse Liste (Last accessed on 09.07.2016)
Available from `https://www.weisse-liste.de/de/`

[8] ZPID PSYNDEX (Last accessed on 09.07.2016)
Available from `https://www.zpid.de/index.php?wahl=PSYNDEX`

[9] Genetic and Rare Diseases Information Center (GARD) | Genetic and Rare Diseases Information Center (GARD) – an NCATS Program (Last accessed on 14.05.2016)
Available from `https://rarediseases.info.nih.gov/gard`

[10] Orphanet: an online rare disease and orphan drug data base. © INSERM 1997 (Last accessed on 16.05.2016)
Available from `http://www.orpha.net.`

[11] U.S. National Library of Medicine (2013) - Author, Corporate Author, and Collaborator Affiliation Display Changes (Last accessed on 19.03.2016)

Available from https://www.nlm.nih.gov/pubs/techbull/nd13/nd13_author_affiliation_display.html

[12] Allianz Chronischer Seltener Erkrankungen (Last accessed on 19.05.2016)
Available from http://www.achse.info/

[13] EURORDIS: Rare Diseases Europe (Last accessed on 19.05.2016)
Available from http://www.eurordis.org/

[14] Expertscape: Your first step to a Second Opinion (Last accessed on 19.05.2016)
Available from http://www.expertscape.com

[15] FindZebra: Help diagnose rare diseases (Last accessed on 19.05.2016)
Available from http://www.findzebra.com/about

[16] NORD: National Organization for Rare Disorders (Last accessed on 19.05.2016)
Available from http://www.rarediseases.org/

[17] RareConnect: Connecting rare disease patients globally (Last accessed on 19.05.2016)
Available from https://www.rareconnect.org

[18] se-atlas: Kartierung von Versorgungseinrichtungen für Menschen mit Seltenen Erkrankungen (Last accessed on 19.05.2016)
Available from https://www.se-atlas.de/

[19] XML Element Descriptions and their Attributes (Last accessed on 20.03.2016)
Available from https://www.nlm.nih.gov/bsd/licensee/elements_descriptions.html

[20] National Center for Biotechnology Information, U.S. National Library of Medicine: PubMed (Last accessed on 21.05.2016)
Available from http://www.ncbi.nlm.nih.gov/pubmed/

[21] Afzal M T, Latif A, Saeed A U, Sturm P, Aslam S, Andrews K, Tochtermann K, Maurer H: *Discovery and Visualization of Expertise in a Scientific Community: Proceedings of the 7th International Conference on Frontiers of Information Technology, Abbottabad, Pakistan, December 16 - 18, 2010.* ACM, New York, NY

[Conference Paper]
Available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.664.7695&rep=rep1&type=pdf

[22] Alani H, Dasmahapatra S, O'Hara K, Shadbolt N: Identifying communities of practice through ontology network analysis. IEEE Intelligent Systems, 18:pp. 18–25 (2003)

[23] Aleman-Meza B, Bojars U, Boley H, Breslin J G, Mochol M, Nixon L J B, Polleres A, Zhdanova Anna V: Combining RDF Vocabularies for Expert Finding. *In:* E Franconi (Ed.) *The semantic web*, vol. 4519 of *SpringerLink: Springer e-Books*. Springer, Berlin [u.a.] (2007)

[24] Amitay E, Carmel D, Golbandi N, Har'El N, Ofek-Koifman S, Yogev S: Finding People and Documents, Using Web 2.0 Data. *In: SIGIR '08*. ACM, New York [Conference Paper] (2008)

[25] Aymé S, Kole A, Groft S: Empowerment of patients: lessons from the rare diseases community. Lancet (London, England), 371:pp. 2048–2051 (2008)

[26] Aymé S, Schmidtke J: Networking for rare diseases: a necessity for Europe. Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz, 50:pp. 1477–1483 (2007)

[27] Becerra-Fernandez I: Searching for Experts on the Web: A Review of Contemporary Expertise Locator Systems. ACM Transactions on Internet Technology, 6:pp. 333–355 (2006)

[28] Bethesda (MD): National Center for Biotechnology Information (US) 2005: PubMed Help [Internet]: [Table, Stopwords] (Last accessed on 07.06.2016)
Available from http://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/

[29] Bethesda (MD): National Center for Biotechnology Information (US) 2005: PubMed Help [Internet]: How PubMed works: automatic term mapping (Last accessed on 28.04.2016)

Available from http://www.ncbi.nlm.nih.gov/books/NBK3827/pdf/Bookshelf_NBK3827.pdf

[30] Bethesda (MD): National Center for Biotechnology Information (US) 2005: PubMed Help [Internet]: Search Field Descriptions and Tags (Last accessed on 28.04.2016)
Available from http://www.ncbi.nlm.nih.gov/books/NBK3827/pdf/Bookshelf_NBK3827.pdf

[31] Bilenko M, Mooney R, Cohen W, Ravikumar P, Fienberg S: Adaptive name matching in information integration. IEEE Intelligent Systems, 18:pp. 16–23 (2003)
Available from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1234765

[32] Brunsmann F, von Gizycki R, Rybalko A, Hildebrandt A G, Rüther K: Patientenselbsthilfe und seltene Erkrankungen. Mitgestaltung der Versorgungsrealität am Beispiel seltener Netzhautdegenerationen. Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz, 50:pp. 1494–1501 (2007)

[33] Chua S: Using Web 2.0 to Locate Expertise. *In: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, pp. 284–287. ACM, New York, NY [Conference Paper] (2007)

[34] Cohen W, Ravikumar P, Fienberg S: A Comparison of String Distance Metrics for Name Matching Tasks. Kdd workshop on data cleaning and object consolidation. [Conference Paper], pp. 73–78 (2003)
Available from https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf

[35] Craswell N, Hawking D, Vercoustre A M, Wilkins P: PANOPTIC Expert - Searching for Experts. Ausweb Poster Proceedings [Conference Paper], 15:pp. 17–21 (2001)

[36] Crowder R: An Agent Based Approach to Finding Expertise in the Engineering Design Environment. 14th International Conference on Engineering Design: Research for Practice - Innovative Products, Processes and Organisations [Conference Paper] (2003)

[37] da Silva R, Stasiu R, Orengo V M, Heuser C A: Measuring quality of similarity functions in approximate data matching. Journal of Informetrics, 1:pp. 35–46 (2007)
Available from http://dx.doi.org/10.1016/j.joi.2006.09.001

[38] Demartini G: Finding Experts Using Wikipedia. 2nd International ExpertFinder Workshop (FEWS2007) [Conference Paper] (2007)

[39] Deng H, King I, Lyu M R: Formal Models for Expert Finding on DBLP Bibliography Data. *In: Eighth IEEE International Conference on Data Mining (ICDM)*, pp. 163–172. Pisa, Italy [Conference Paper] (2008)

[40] Devillé W, Bezemer P D, Bouter L M: Publications on diagnostic test evaluation in family medicine journals: an optimal search strategy. Journal of clinical epidemiology, 53:pp. 65–69 (2000)

[41] Ehrlich K, Lin C Y, Griffiths-Fisher V: Searching for Experts in the Enterprise: Combining Text and Social Network Analysis. *In:* T Gross (Ed.) *Proceedings of the 2007 international ACM conference on Supporting group work*. ACM, New York, NY [Conference Paper] (2007)

[42] Elliot S: Survey of Author Name Disambiguation: 2004 to 2010. Library Philosophy and Practice (e-journal), 473 (2010)

[43] Golder S, McIntosh H M, Duffy S, Glanville J: Developing efficient search strategies to identify reports of adverse effects in medline and embase. Health Information & Libraries Journal, 23:pp. 3–12 (2006)
Available from http://dx.doi.org/10.1111/j.1471-1842.2006.00634.x

[44] Guan Z, Miao G, McLoughlin R, Yan X, Cai D: Co-Occurrence-Based Diffusion for Expert Search on the Web. IEEE Transactions on Knowledge and Data Engineering, 25:pp. 1001–1014 (2013)

[45] Han H, Zha H, Giles C L: Name disambiguation in author citations using a K-way spectral clustering method. *In:* M Marlino, T Sumner, F Shipman (Eds.) *The 5th ACM/IEEE-CS joint conference*, p. 334. Denver, CO, USA [Conference Paper] (2005)

[46] Heimpel H: (Center for Rare Diseases Ulm) Presentation and discussion of preliminary sample expert and disorder profiles (2014 pers. comm.)

[47] Huang J, Ertekin S, Giles C L: Efficient Name Disambiguation for Large-Scale Databases. *In:* J Fürnkranz (Ed.) *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases*, vol. 4213 of *Lecture Notes in Computer Science*, pp. 536–544. Springer, Berlin [u.a.] (2006)

[48] Jonnalagadda S, Topham P: NEMO: Extraction and normalization of organization names from PubMed affiliation strings. Journal of Biomedical Discovery and Collaboration, 5:pp. 50–75 (2010)

[49] Koster J: PubMed PubReMiner: Detailed analysis of PubMed Search results: [Internet] (2004)
Available from `http://hgserver2.amc.nl/cgi-bin/miner/miner2.cgi`

[50] Kühnel A: *Visual C# 2012: Das umfassende Handbuch ; [Spracheinführung, Objektorientierung, Programmiertechniken ; Windows-Programmierung mit der Windows Presentation Foundation ; inkl. LINQ, Task Parallel Library (TPL), ADO.NET und Entity Framework]*, vol. 1997 of *Galileo Press Galileo Computing*. Galileo Press, Bonn, 6., aktualisierte und erw. aufl edn. (2013)

[51] Langfelder P, Zhang B, Horvath S: Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics (Oxford, England), 24:pp. 719–720 (2008)

[52] Latif A, Afzal M T, Tochtermann K: Constructing Experts Profiles from Linked Open Data. *In: ICET 2010*. IEEE Islamabad Section, Islamabad, Pakistan [Conference Paper] (2010)

[53] Liu J, Lei K H, Liu J Y, Wang C, Han J: Ranking-based name matching for author disambiguation in bibliographic data. *In: Proceedings of the 2013 KDD Cup Workshop*, pp. 1–8. Chicago, Illinois [Conference Paper] (2013)

[54] Liu P, Ye Y, Liu K: Building a Semantic Repository of Academic Experts. *In: 4th International Conference on Wireless Communications, Networking and Mobile*

Computing, 2008. IEEE Operations Center, Piscataway, NJ [Conference Paper] (2008)

[55] Liu W, Islamaj Doğan R, Kim S, Comeau D C, Kim W, Yeganova L, Lu Z, Wilbur W J: Author name disambiguation for PubMed. Journal of the Association for Information Science and Technology, 65:pp. 765–781 (2014)

[56] Mattern F: Zur Evaluation der Informatik mittels bibliometrischer Analyse. Informatik-Spektrum, 25:pp. 22–32 (2002)

[57] McDonald D W: Just Talk to Me: A Field Study of Expertise Location. Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98) [Conference Paper], pp. 14–18 (1998)

[58] Omidvar A, Garakani M, Safarpour H R: Context based user ranking in forums for expert finding using WordNet dictionary and social network analysis. Information Technology and Management, 15:pp. 51–63 (2014)

[59] Pflugrad A, Bernauer J: Nutzung von String-Ähnlichkeitsmaßen in Talend-Open-Studio zur Desambiguierung von Autorennamen aus PubMed. mdi - Forum der Medizindokumentation und Medizininformatik, 17:pp. 17–18 (2015)

[60] Pflugrad A, Jurkat-Rott K, Lehmann-Horn F, Bernauer J: Towards the automated generation of expert profiles for rare diseases through bibliometric analysis. Studies in health technology and informatics, 198:pp. 47–54 (2014)

[61] Rath A, Olry A, Dhombres F, Brandt M M, Urbero B, Ayme S: Representation of rare diseases in health information systems: the Orphanet approach to serve a wide range of end users. Human mutation, 33:pp. 803–808 (2012)

[62] Rath A, Rodwell C, Arunachalam S: (Orphanet) Discussion about possible usage scenarios of the bibliometric expert finding system within Orphanet projects (2015 pers. comm.)

[63] Sayers E: A General Introduction to the E-utilities. In: Entrez Programming Utilities Help [Internet] (2010-.)
Available from `http://www.ncbi.nlm.nih.gov/books/NBK25497/`

[64] Sayers E, Miller V: E-utility Web Service (SOAP) 2010 Jan 21 [Updated 2015 Jan 23]. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US) (2010-.)
Available from http://www.ncbi.nlm.nih.gov/books/NBK43082/

[65] Schaafsma F, Hulshof C, Verbeek J, Bos J, Dyserinck H, van Dijk F: Developing search strategies in Medline on the occupational origin of diseases. American Journal of Industrial Medicine, 49:pp. 127–137 (2006)
Available from http://dx.doi.org/10.1002/ajim.20235

[66] Schäffter M: Veröffentlichung von statistischen Auswertungen mit Personenbezug: Datenschutzrechtliche Stellungnahme: (2016 unpublished advisory statement)

[67] Schmidtke J: (Orphanet Germany) Response rate of questionnaires sent to clinics. (2013 pers. comm.)

[68] Schmidtke J: (Orphanet Germany) Request for the option of disregarding middle authors in the user client as a filter function (2016 pers. comm.)

[69] Schmidtke J: (Orphanet Germany) Results of a manual evaluation of an author list provided to Orphanet (2016 pers. comm.)

[70] Schwarz K: (Center for Rare Diseases Ulm) Discussion about the necessity of including rare diseases which are not covered by the Orphanet classification (2014 pers. comm.)

[71] Schwarzkopf M: *Aufbereitung von Daten zu seltenen Erkrankungen und deren performante Visualisierung in einem Webportal.* Masterthesis, University of Applied Sciences, Ulm (2015)

[72] Schwarzkopf M, Dangelmaier H, Pflugrad A, Bernauer J: Komponenten eines Expertensuchsystems zu Seltenen Erkrankungen auf der Basis bibliometrischer Daten und automatisierter Internetsuche. Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie [Conference Paper], 2015

[73] Shu L, Chen A, Ming X, Meng W: Efficient Spectral Neighborhood Blocking for Entity Resolution. *In: IEEE 27th International Conference on Data Engineering*

(ICDE), pp. 1067–1078. [Conference Paper] (2011)
Available from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5767835

[74] Slater L: Product: ReviewPubMed PubReMiner. Journal of the Canadian Health Libraries Association, 33:pp. 106–107 (2014)

[75] Song Y, Huang J, Councill I G, Li J, Giles C L: Efficient topic-based unsupervised name disambiguation. *In:* E Rasmussen, R R Larson, E Toms, S Sugimoto (Eds.) *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, p. 342. Vancouver, BC, Canada [Conference Paper] (2007)

[76] Stankovic M, Wagner C, Jovanovic J, Laublet P: Looking for Experts? What can Linked Data do for You? *In:* C Bizer, T Heath, T Berners-Lee, M Hausenblas (Eds.) *Proceedings of the WWW2010 Workshop on Linked Data on the Web.* [Conference Paper] (2010)

[77] Tang J, Fong A, Wang B, Zhang J: A Unified Probabilistic Framework for Name Disambiguation in Digital Library. IEEE Transactions on Knowledge and Data Engineering, 24:pp. 975–987 (2012)
Available from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5680902

[78] Tang J, Zhang J, Zhang D, Yao L, Zhu C, Li J: ArnetMiner: An Expertise Oriented Search System for Web Community. Proceedings of the 2007 International Conference on Semantic Web Challenge [Conference Paper], 295 (2007)

[79] THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION: REGULATION (EC) No 141/2000 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 16 December 1999 on orphan medicinal products on orphan medicinal products (22. 1. 2000)
Available from http://ec.europa.eu/health/files/eudralex/vol-1/reg_2000_141/reg_2000_141_en.pdf

[80] Torvik V I, Smalheiser N R: Author Name Disambiguation in MEDLINE. ACM transactions on knowledge discovery from data, 3 (2009)

[81] Torvik V I, Weeber M, Swanson D R, Smalheiser N R: A probabilistic similarity metric for Medline records: A model for author name disambiguation. Journal

of the American Society for Information Science and Technology, 56:pp. 140–158 (2005)

[82] Vergne M, Susi A: Expert Finding Using Markov Networks in Open Source Communities. *In:* D Hutchison, T Kanade, J Kittler, J M Kleinberg, A Kobsa, F Mattern, J C Mitchell, M Naor, O Nierstrasz, C Pandu Rangan, B Steffen, D Terzopoulos, D Tygar, G Weikum, M Jarke, J Mylopoulos, C Quix, C Rolland, Y Manolopoulos, H Mouratidis, J Horkoff (Eds.) *Advanced Information Systems Engineering*, vol. 8484 of *Lecture Notes in Computer Science*, pp. 196–210. Springer International Publishing, Cham (2014)

[83] Wang X, Tang J, Cheng H, Yu P S: ADANA: Active Name Disambiguation. *In: IEEE 11th International Conference on Data Mining (ICDM)*, pp. 794–803. Vancouver, BC, Canada [Conference Paper] (2011)

[84] Weller K, Vetter-Kauzcok C, Kähler K, Hauschild A, Eigentler T, Pföhler C, Neuber K, Moll I, Krause M, Kneisel L, et al.: Umsetzung von Leitlinien bei seltenen Erkrankungen am Beispiel des Merkelzellkarzinoms. Deutsches Ärzteblatt, 103:pp. A2791–A2796 (2006)

[85] Wetterauer B, Schuster R: Seltene Krankheiten: Probleme, Stand und Entwicklung der nationalen und europäischen Forschungsförderung. Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz, 51:pp. 519–528 (2008)

[86] Wilczynski N L, Haynes R B: Developing optimal search strategies for detecting clinically sound prognostic studies in MEDLINE: an analytic survey. BMC medicine, 2 (2004)

[87] Wilczynski N L, Haynes R B: EMBASE search strategies achieved high sensitivity and specificity for retrieving methodologically sound systematic reviews. Journal of clinical epidemiology, 60:pp. 29–33 (2007)

[88] Wren J D, Kozak K Z, Johnson K R, Deakyne S J, Schilling L M, Dellavalle R P: The write position. A survey of perceived contributions to papers based on byline position and number of authors. EMBO reports, 8:pp. 988–991 (2007)

[89] Wu C J, Chung J M, Lu C Y, Lee H M, Ho J M: Using Web-Mining for Academic Measurement and Scholar Recommendation in Expert Finding System. *In: 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pp. 288–291. Lyon, France [Conference Paper] (2011)

[90] Zhang J, Ackerman M S, Adamic L: Expertise Networks in Online Communities: Structure and Algorithms. *In:* C Williamson, M E Zurko (Eds.) *Proceedings of the 16th International Conference on World Wide Web 2007*. ACM Press, New York, NY [Conference Paper] (2007)

[91] Zhao Z, Zhang L, He X, Ng W: Expert Finding for Question Answering via Graph Regularized Matrix Completion. IEEE Transactions on Knowledge and Data Engineering, 27:pp. 993–1004 (2015)
Available from `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6909004`

Acknowledgements

I would like to thank Prof. Dr. Jochen Bernauer and PD Dr. Karin Jurkat-Rott for making this work possible as well as their supervision and guidance.
Additional thanks go to Prof. Dr. Hans Kestler and the members of the PhD board.

I'd also like to thank Marina Schwarzkopf for her tireless support of this project without which it would not have come this far.

Further thanks go to the Centre of Excellence for Rare Diseases Baden-Württemberg as well as the Centre for Rare Diseases Ulm for their support, especially to Prof. Dr. Dr.h.c. Frank Lehmann-Horn and the late Prof. Dr. Hermann Heimpel for their insights on rare diseases expertise.

I'd also like to express my gratitude towards the Friedrich-Wingert-Foundation, the Hertie-Foundation and IonNeurONet for their support.

Another thanks goes to the Orphanet team in Paris as well as Orphanet Germany for their insights into rare diseases expert data curation and their interest in this project.

Thanks to Prof. Dr. Markus Schäffter and Branislav Babic for their counsel regarding data privacy and data protection.

Many thanks to Neltje Piro and Susanne Nickolmann for proofreading and providing valuable comments on the thesis.

Lastly, thanks to everybody who has not been mentioned here but still supported me in any way.

Very special thanks go to Patricia Krepf whose invaluable support through all the years has helped me to persevere and finish this work.

- Andreas Pflugrad, December 2016

Curriculum Vitae

Personal Details

Name	Andreas Pflugrad
Born	12.03.1986 in Villingen-Schwenningen

Education

2013 - 2016	**Doctoral studies**, Ulm University, Ulm University of Applied Sciences.
2009 - 2012	**Master course (M.Sc.)**: Information Systems with focus on 'Medical Information Systems', Ulm University of Applied Sciences Master thesis: *Entwicklung einer mobilen Applikation zur regelbasierten Interpretation klinischer Laborbefunde*. Final grade: 1.7
2006 - 2009	**Bachelor course (B.Sc.)**: Documentation and Computer Science in Medicine, Ulm University of Applied Sciences Bachelor thesis: *Creating a web-based infrastructure for a protein infrared spectra database (PISD)*. Final grade: 1.3

Professional Experience

03/2013 - 08/2016	**Research associate**, Division of Neurophysiology, Ulm University
10/2011 - present	**Research assistant**, Computer Science Institute, Ulm University of Applied Sciences

Publications

Pflugrad A, Bernauer J. *Nutzung von String-Ähnlichkeitsmaßen in Talend-Open-Studio zur Desambiguierung von Autorennamen aus PubMed.* mdi - Forum der Medizindokumentation und Medizininformatik; 2015; 17(1):17–8.

Schwarzkopf M, Dangelmaier H, **Pflugrad A**, Bernauer J. *Komponenten eines Expertensuchsystems zu Seltenen Erkrankungen auf der Basis bibliometrischer Daten und automatisierter Internetsuche.* Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie; 2015.

Pflugrad A, Jurkat-Rott K, Lehmann-Horn F, Bernauer J. *Automatische Generierung von Expertenprofilen zu Seltenen Erkrankungen.* Deutsche Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie; 2014.

Pflugrad A, Jurkat-Rott K, Lehmann-Horn F, Bernauer J. *Towards the automated generation of expert profiles for rare diseases through bibliometric analysis.* Studies in health technology and informatics 2014; 198:47–54.

Awards and Grants

2015	Friedrich-Wingert-Award
2014	Scholarship of the Friedrich-Wingert-Foundation

Conferences and Invited Talks

03/2016	**Invited Talk**: *Finding experts on rare diseases from literature data*, Rare Diseases Day 2016 Symposium, Berlin
04/2015	**Invited Talk**: *Konzeption, Umsetzung und Evaluation einer computergestützten Expertensuche für Patienten mit Seltenen Erkrankungen*, conhIT, Berlin
03/2015	**Conference Talk**: *Nutzung von String-Ähnlichkeitsmaßen in Talend Open Studio zur Desambiguierung von Autorennamen aus Pubmed*, DVMD-conference, Ulm
09/2014	**Conference Talk**: *Automatische Generierung von Expertenprofilen zu Seltenen Erkrankungen*, GMDS annual conference, Göttingen
05/2014	**Conference Talk**: *Towards the Automated Generation of Expert Profiles for Rare Diseases through Bibliometric Analysis*, eHealth summit Austria, Vienna