ADVANCED METHODS FOR ENTITY LINKING IN THE LIFE SCIENCES

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)
im Fachgebiet Informatik

vorgelegt von
M. Sc. Informatik Victor Christen
geboren am 01. November 1988 in Potsdam

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Erhard Rahm (Universität Leipzig)
2. Prof. Dr. Maria-Esther Vidal (Leibniz Universität Hannover)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 15. Dezember 2020 mit dem Gesamtprädikat *magna cum laude*.

UNIVERSITÄT
LEIPZIG

Acknowledgments

This thesis is the result of the past six years as a research assistant in the Database Group at Leipzig University. The time was filled with fruitful discussions with Prof Dr. Erhard Rahm allowed me to research in the areas of this thesis. Therefore, I like to thank him for his support and guidance during my doctoral time.

I like to thank my colleagues in the Database Group for their support, which led to some amusing lunch breaks and after-work activities. In particular, a special thanks goes to Prof. Dr. Anika Groß who encouraged me for the topics of this thesis and gave me constructive advices as well as ideas resulting in this thesis. Further thanks goes to Markus Nentwig and Martin Franke for the feedback on my dissertation. In addition to the colleagues at the Database Group, I like to thank the colleagues at the Australian National University, especially Prof. Dr. Peter Christen, who listened to my entity resolution approaches and supported me to improve them. I would be lost without the support of our secretary Andrea Hesse who helped me to organize business trips or to fill some of my knowledge gaps.

I owe special thanks to my mother Cornelia Christen and my brother Rafael - who always encouraged me to go ahead with my computer science study and lovingly supported me. Further thanks goes to my old school friend and flatmate Sebastian for philosophical and scientific discussion besides computer science. Lovingly thanks goes to Kathrin for her patience, positive encouragement and culinary insights that enrich my life like entity resolution topics.

Leipzig, 29. Juli 2020 Victor Christen

Abstract

The amount of knowledge increases rapidly due to the increasing number of available data sources. However, the autonomy of data sources and the resulting heterogeneity prevent comprehensive data analysis and applications.

Data integration aims to overcome heterogeneity by unifying different data sources and enriching unstructured data. The enrichment of data consists of different subtasks, amongst other the annotation process. The annotation process links document phrases to terms of a standardized vocabulary. Annotated documents enable effective retrieval methods, comparability of different documents, and comprehensive data analysis, such as finding adversarial drug effects based on patient data.

A vocabulary allows the comparability using standardized terms. An ontology can also represent a vocabulary, whereas concepts, relationships, and logical constraints additionally define an ontology. The annotation process is applicable in different domains. Nevertheless, there is a difference between generic and specialized domains according to the annotation process. This thesis emphasizes the differences between the domains and addresses the identified challenges. The majority of annotation approaches focuses on the evaluation of general domains, such as Wikipedia. This thesis evaluates the developed annotation approaches with case report forms that are medical documents for examining clinical trials. The natural language provides different challenges, such as similar meanings using different phrases. The proposed annotation method, AnnoMap, considers the fuzziness of natural language. A further challenge is the reuse of verified annotations. Existing annotations represent knowledge that can be reused for further annotation processes. AnnoMap consists of a reuse strategy that utilizes verified annotations to link new documents to appropriate concepts. Due to the broad spectrum of areas in the biomedical domain, different tools exist. The tools perform differently regarding a particular domain. This thesis proposes a combination approach to unify results from different tools. The method utilizes existing tool results to build a classification model that can classify new annotations as correct or incorrect.

The results show that the reuse and the machine learning-based combination improve the annotation quality compared to existing approaches focussing on the biomedical domain.

A further part of data integration is entity resolution to build unified knowledge bases from different data sources. A data source consists of a set of records characterized by attributes. The goal of entity resolution is to identify records representing the same real-world entity. Many methods focus on linking data sources consisting of records being characterized by attributes. Nevertheless, only a few methods can handle graph-structured knowledge bases or consider temporal aspects. The temporal aspects are essential to identify the same entities over different time intervals since these aspects underlie certain conditions. Moreover, records can be related to other records so that a small graph structure exists for each record. These small graphs can be linked to each other if they represent the same. This thesis proposes an entity resolution approach for census data consisting of person records for different time intervals. The approach also considers the graph structure of persons given by family relationships.

For achieving qualitative results, current methods apply machine-learning techniques to classify record pairs as the same entity. The classification task used a model that is generated by training data. In this case, the training data is a set of record pairs that are labeled as a duplicate or not. Nevertheless, the generation of training data is a time-consuming task so that active learning techniques are relevant for reducing the number of training examples.

The entity resolution method for temporal graph-structured data shows an improvement compared to previous collective entity resolution approaches. The developed active learning approach achieves comparable results to supervised learning methods and outperforms other limited budget active learning methods.

Besides the entity resolution approach, the thesis introduces the concept of evolution operators for communities. These operators can express the dynamics of communities and individuals. For instance, we can formulate that two communities merged or split over time. Moreover, the operators allow observing the history of individuals.

Overall, the presented annotation approaches generate qualitative annotations for medical forms. The annotations enable comprehensive analysis across different data sources as well as accurate queries. The proposed entity resolution approaches improve existing ones so that they contribute to the generation of qualitative knowledge graphs and data analysis tasks.

Contents

List of Figures

List of Tables

Part I

Foundations

1
Introduction

1.1 Motivation

Information influence our life in economic, social, and especially medical aspects. Different organizations publish their data in different formats, such as unstructured documents or databases. However, the value of the massive amount of data can only be utilized if the data quality is high so that data become information. Data analysis and machine learning-based applications are based on unified and expressiveness data.

Data representations range from unstructured documents to structured databases. Moreover, data sources are autonomous so that they use different semantics to describe their data. Data integration aims to overcome the heterogeneity of different data sources and enrich data with unified knowledge. The different representations result in various levels of semantics: schema level and instance level. Consequently, data integration is divided into subtasks, which focus on different aspects. Figure 1.1 gives an overview of the different data levels and the sub-

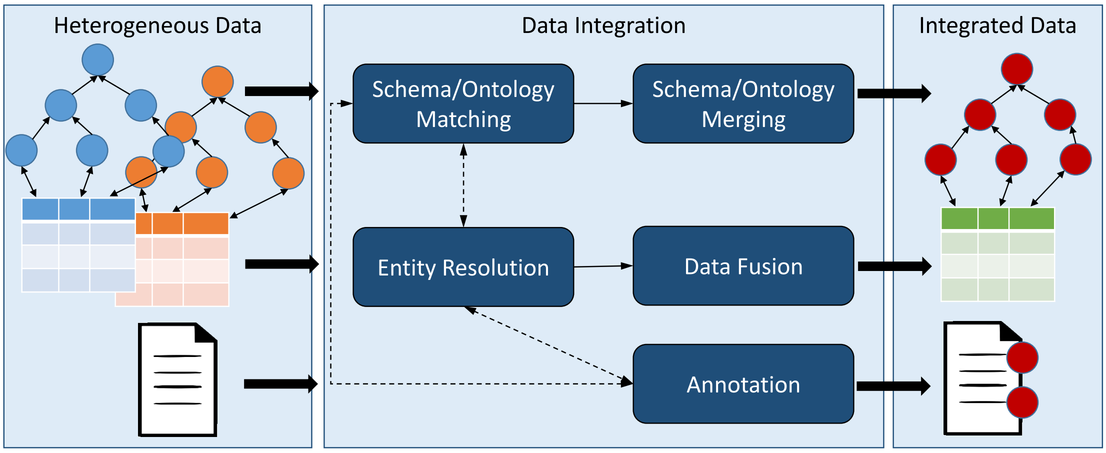


Figure 1.1: Overview of different data types and integration processes. The left side shows the different data types: meta data and instances as well as structured data in form of tables and unstructured documents. The dashed arrows indicate that a process supports another one.

tasks of data integration. *Schema matching*, *Schema Merging*, *Entity resolution* and *Data Fusion*, aim to generate integrated data regarding different schemata and instances. Schema or ontology matching methods identify the same meta elements from different data sources. Entity resolution disambiguates records that correspond to real-world entities across and within data sources. The identified correspondences between meta elements and instances are fused to an integrated data source, which is the goal of *Schema Merging* and *Data Fusion*. A further part of data integration covers the *linking* of data with terms of standardized vocabularies or ontologies being commonly used in the Semantic Web. Entity linking or the *annotation* process represents the task of disambiguating entities regarding their meaning using terms of vocabularies. Different types of data can be annotated, such as medical images, genes, and documents. The different processes support each other. For instance, schema matching methods can utilize integrated instances. Moreover, entity resolution methods use annotations of documents to deduplicate them. The resulting linked data sources are comparable and effectively searchable. Data integration enables a wide spectrum of applications in machine learning and cross-data source query engines.

Especially in the life sciences, an enormous amount of data is semi-structured or unstructured, such as scientific publications, medical documents, and medical images. From the characteristics of Big Data, the variety is high in the medical domain, so that this challenge must be overcome for expressiveness analysis and advanced applications. The potential of applications ranges from personalized medicine to medical AI. The idea of personalized medicine is that physicians advise patients based on their evaluated and integrated electronic health records linked to diagnosis and potential drugs. In addition to the medical data, personal information are relevant to analyze social behavior or genetic diseases. The different applications in medicine and social analysis require the disambiguation of entities by linking them to terms of vocabularies and the entity representing the same, also known as entity resolution. Moreover, the usage of annotations supports the realization of productive scientific data management following the FAIR(findable, access, interoperable, reusable) guiding principles [143] improving the productivity of research and pharmaceutical industry.

A specific type of medical documents are case report forms (CRF) being essential for determining eligibility criteria for clinical trials so that probands can be recruited. There exist different initiatives providing CRFs and information about the results of clinical trials such as ClinicalTrials.gov. ClinicalTrials.gov provides access to roughly 340,000 research studies (June 2020). Linked or annotated forms allow the integration of results from different studies and enable the reuse of probands criteria for examining similar studies. In addition to the analysis of medical data, researchers study genetic diseases and social behavior using historical census data [61, 137].

The application of such analysis requires integrated data. A manual annotation or entity resolution process is infeasible due to the massive number of unannotated documents and vocabulary size. Therefore, automatized approaches are necessary to generate intermediate results being verified by experts. An automated annotation approach identifies for text mentions of a document concept candidates and disambiguates them so that each mention is linked to vocabulary terms. Due to the natural language formulated documents and the considerable amount of existing medical forms, different forms can be highly heterogeneous while they consist of similar semantics so that an annotation approach has to address several challenges. Moreover, census data can consist of ambiguous person data or data quality problems so that entity resolution methods must address these issues. In the following, we list the different challenges for annotation and entity resolution methods.

Heterogeneity Similar semantics can be expressed differently by the natural language. For instance, the concept representing *Myocardial infarct* from UMLS has six different synonyms, such as *myocardial necrosis*, so that different documents consist of mentions with different representatives but are linked to the same concept. Due to the amount of variety of the natural language, an annotation tool has to provide methods to overcome the issues according to synonyms, homonyms, and the identification of entities. The usage of simple string similarity functions is not sufficient since the semantics of phrases are not covered. Therefore, the approach must consider several methods covering the importance of phrases as well as the context of entities how they occur in documents. In the case of linking person records between different data sources, the methods must address data quality issues, such as typos or missing values.

Link Quality The quality of the analysis depends on the quality of the integration process and implicitly on the annotation and entity resolution process. Complex similarity functions are used based on similarities between attribute values of records to address data quality problems and the heterogeneity of data sources. The complex similarity function can be defined in different ways: manual or automatized using machine learning techniques. However, a manual definition of a complex function is a crucial and error-prone task, so that automatized methods lead to more accurate results in terms of quality. Nevertheless, machine learning methods utilize training data consisting of classified record pairs indicating a *match* or a *non match*.

Reuse and Combination The biomedical domain provides a considerable amount of unannotated documents. Moreover, the number of documents will increase over time, which leads to a growing knowledge base. Therefore, the existing annotations can be reused to improve the annotation process for new documents. To enable the reuse of annotations, an effective and efficient representation is required for identifying appropriate concepts based on the main phrases between concept and mention of a new document. In addition to the reuse of annotated documents, many annotation tools are available for processing medical documents. Nevertheless, the different tools result in different qualities depending on the domain and the structure of documents. The combination of annotations from the results of different tools overcomes the deficiencies of the usage of a single tool. Machine learning methods for entity resolution reuse existing verified links. A crucial task for generating a training dataset is the selection of links.

Linking of entity substructures Data sources do not only provide flat data so that each record is described with its attributes. Records can also be structured in a graph. In addition to the challenging identification of record matches, a method dealing with graph data must consider links between similar substructures of graphs such as the same households in census data. The identification of similar substructures needs to consider features characterizing the similarity between relationships. For instance, in historical census datasets, family relationships with other people can be used to obtain more evidence for the similarity.

Analysis of temporal data Data are not static since the environment evolves. Therefore, data models such as schemas or ontologies must be adapted to current requirements. Moreover, entities change their states, such as persons getting a new profession or marrying and changing their surnames. The analysis of evolving data requires the identification of the same entity for different points in time. The resulting links allow the analysis of entities or groups over time. Nevertheless, the analysis is challenging since a standardized operator set for describing the evolution of entities or subgroups of entities is missing. These operators would give an overview of the evolution of dynamic data.

The current applications of annotated documents focus on information retrieval and statistical analysis. In the scientific domain, PubMed provides a term-based search utilizing MeSH as the vocabulary to find scientific publications for a particular topic, so that physicians and medical scientists can determine new insights of a specific area and use the knowledge for their research. In addition to the document search, annotations enable useful analysis. For instance, LePendu[75] analyzed adverse drug events based on electronic health records. To statistically collect the pairs of adverse events and drugs, they must be identified in the documents and annotated for the statistical comparability. As a result, the authors confirmed that patients who took Vioxx showed significantly elevated risk for heart infarction. In [52], similar clinical trials have been clustered by performing a nearest neighbor search using annotated eligibility criteria and applying a dictionary-based pre-annotation method [82] showed to improve the speed of manual annotation for clinical trial announcements. In [89], a set of eligibility criteria in the context of clinical trials on breast cancer, is formalized by defining eligibility criteria for specific patterns to improve their comparability.

The current research does not entirely address the identified challenges. Notably, the reuse of existing annotations as reuse repositories for improving the annotation process are not investigated. Moreover, only a few methods focus on annotating questions of case report forms. The majority of entity resolution methods focus on static and flat data and not on evolving and graph-structured data. This thesis address the different challenges by the following contributions.

1.2 Scientific Contribution

Annotation of heterogeneous medical forms

Medical forms are an essential part of medical science, such as the examination of clinical trials. Based on the necessity, we develop an annotation framework - AnnoMap - that is customized for processing medical forms. The annotation process addresses the challenges of natural language and data quality issues. Further, we evaluate the implemented approach with real-world datasets provided by the medical data models platform from Münster University. The annotation process was accepted for presentation at the DILS conference in 2015 and published in the conference proceedings [27].

Concept for Reuse of Annotations

The research on medical annotations provides many tools as well as few initiatives for storing documents and annotations. We conceptualize and implement a strategy to reuse annotations of medical forms regarding certain domains to annotate new forms. The developed method utilizes annotation clusters represented by key phrases of annotations. The usage of the annotation cluster improves the annotation quality compared to an existing method. Moreover, the implemented context similarity method leads to an improvement in the selection step compared to basic selection strategies. The approach was presented at the ISWC 2016 and published in the conference proceedings [26].

Machine Learning-based Combination of annotation tools

The biomedical domain covers a wide range of topics so that different tools were developed that perform qualitatively different depending on the domain. We develop a machine learning-based ensemble method that combines the annotation results from different tools. The evaluation shows that the quality increases more compared to set based combination approaches. The method was presented at the DILS conference in 2018 and published in the conference proceedings [28].

Graph integration and Temporal analysis

Traditional entity resolution approaches only consider data sources at a given point in time and focus on record-wise processing. In our approach, we combine temporal aspects with graph-based aspects to determine matches between small communities, such as households. The resulting matches of households reduce the search space for identifying matches between person records. Moreover, we introduce evolution operators to understand the dynamics of communities how they change and which members move from one community to another community. The evaluation shows an improvement of the linkage quality compared to other collective entity resolution approaches. Furthermore, the evolution analysis for a British census dataset using the operators provides an overview of common patterns regarding person movement and community behavior. The approach was published in the proceedings of the EDBT conference [25].

Classifier Independent Active Learning

The identification of matches is a challenging task due to the heterogeneity of data sources and entity representations. Therefore, approaches based on machine learning are used to determine classification models automatically. However, these approaches require training data that must be created manually. Active learning techniques aim to reduce the amount of training data. In our approach, we propose a strategy for selecting informative training instances based on their location in a vector space. This strategy is the main difference to other approaches that use intermediate classification results. The method was presented at the DINA workshop in 2019 and published in the ECML workshop proceedings [24].

1.3 Structure of Thesis

This dissertation consists of three main parts. The first part consists one further chapter:

Chapter 2 introduces the data structures for the annotation process. With these data structures, the annotation process is described, and different techniques for the different steps are explained. We emphasize the differences between annotating documents from generic domains compared to specific domains. Furthermore, we explain the general steps for entity resolution and, in this context, the related work. In the end, we propose the quality measures to evaluate annotation as well as entity resolution methods.

Part II - Medical Entity Linking - focuses on methods and strategies for annotating medical documents.

Chapter 3 proposes a method for identifying annotations for medical forms. We suggest a combination of different metrics to consider the fuzziness of natural language. Moreover, we use a novel group selection strategy to determine the final result.

Chapter 4 introduces the reuse of annotations and extends the basic approach using annotation cluster repositories, where an annotation cluster is a detailed representation. The extended approach utilizes the annotation clusters. Moreover, we propose a context similarity measure to improve the previous group-based selection strategy.

Chapter 5 complements the reuse of annotations with an ensemble method for combining results from different annotation tools. The method utilizes the computed scores of each tool to build annotation vectors. Using the vectors, a machine learning model is built that can classify a candidate as annotation or not.

Part III - Application of Entity Resolution methods - presents methods for improving the quality for entity resolution results considering temporal graph data as well as heterogeneous data where machine learning based approaches are helpful.

Chapter 6 focuses on a method for identifying links in temporal graph data, such as census data. The method utilizes household information and relationships between persons of the same households to determine links between them. Further, we propose a set of operations representing the evolution of households and persons. We compare our subgraph-based entity resolution approach with other approaches.

Chapter 7 complements the entity resolution part with an budget limited active learning approach for determining informative links. The selected links are used to determine a classification model. Our selection method uses the location of vectors in the vector space to select informative links. The evaluation shows a comparison with other active learning methods and supervised approaches.

The last Part IV - Conclusion and Outlook - concludes the results of this dissertation and provides concepts and ideas for future work.

2

Background and Related Work

This chapter describes semantic annotation in Section 2.1 and entity resolution in Section 2.2. Subsection 2.1.1 consists of the description of the used data model and Subsection 2.1.2 describes the general annotation approach. Moreover, we refer to related work that propose methods for the different steps. A detailed discussion is found in the related chapters 3, 4 and Chapter 5. Subsection 2.1.3 emphasizes the difference in the annotation process for general domains and specialized domains like the life sciences.

Section 2.2 describes the general entity resolution process and refer to related work for the different steps. Advanced techniques that consider the context of records as well as machine learning techniques are discussed in the corresponding Chapter 6, respectively, Chapter 7. At the beginning of this section, the problem definition is proposed. Section 2.3 describes the quality measurements for evaluating the methods in this thesis. Section 2.4 summarizes the chapter.

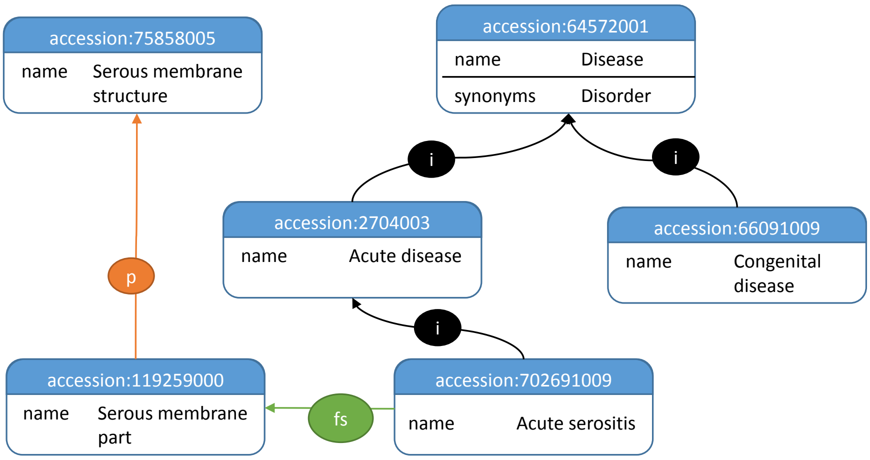


Figure 2.1: Subset of the SNOMED CT ontology, i = *is_a*, fs=has finding site, p=part anatomy structure of.

2.1 Semantic Annotation

2.1.1 Model

Ontology

Ontologies and standardized vocabularies are commonly used to annotate data. The literature provides different definitions for the term ontology. In computer science, a standard definition was given by Gruber [48], who defines an ontology as an explicit specification of a conceptualization. A conceptualization determines the objects, concepts, and other entities that exist in a specific domain. Conceptualizations are typically in the mind of humans so that they are implicitly defined. Ontologies define the semantic of concepts by logical constraints so that the conceptualization is machine-interpretable.

The complexity of ontologies ranges from simple thesauri to formally defined description logics [74]. A vocabulary consists of terms that are represented by textual descriptions. In contrast to vocabularies, taxonomies provide hierarchical relationships between concepts such as a product catalog.

An ontology $\mathbf{O} = \{\mathbf{C}, \mathbf{A}, \mathbf{R}\}$ consists of a set of concepts $\mathbf{C}$, a set of attributes $\mathbf{A}$, and a set of relationships $\mathbf{R}$. Each concept $c \in \mathbf{C}$ is uniquely identifiable by an ID frequently called *accession* number. Moreover, concepts and relationships are textually described by a set of attributes **A** such as a name, synonyms, and further metadata. The relationship between concepts represents the logical constraints such as the generalization, or so-called *is_a* relationship that expresses a subclass relationship. The set of concepts and relationships form a graph. Figure 2.1 shows exemplary a graph structure representation of a subset from the SNOMED CT ontology. Vertices with attributes represent concepts, and relations are direct edges between two concepts. For instance, the concept with accession *702691009* has a name *Acute serositis* and is a subclass of the concept with the accession *2704003*. Moreover, it is related to the concept *Serous membrane part* by the relation *has finding site*. Due to the standardized conceptualization, ontologies are appropriate for annotations where the identifiers of concepts are associated with entities or phrases from unstructured data. The annotated documents are efficient to compare and enable useful analysis.

Different ontologies can overlap if they cover similar topics so that two ontologies can consist of concepts representing the same semantic. Nevertheless, even if the concepts are semantically the same, they can be described differently. *Ontology matching* approaches [108] determine ontology mappings between different ontologies so that the information from all concept definition across different ontologies are useable. An ontology mapping consists of pairs of concepts representing the same semantic. Different platforms provide ontology mappings such as BioPortal [142].

The US National Library of Medicine provides an integrated metathesaurus called Unified Medical Language System(UMLS) [9] consisting of concepts from over 100 different vocabularies. The UMLS consists of three components: Metathesaurus, Semantic Network, and Specialist Lexicon. The Metathesaurus contains concepts where each concept is identifiable by a *concept unique identifier*(cui) and represents concepts from different source ontologies. The textual information of concepts and the relationships from the source ontologies are integrated into UMLS by applying *Schema Merging*. Each concept is related to a *semantic type* from the semantic network representing a topic categorization with relationships between semantic types. Furthermore, the Specialist Lexicon provides linguistic variation for names and synonyms of concepts.

Annotation Mapping

The conceptualization of ontologies provides the possibility to describe real-world entities such as genes, proteins, and medical documents. The association between a concept of an ontology and an entity is called an annotation. For instance, genes are annotated with Gene Ontology concepts to describe the location where a function performs, and the processes of the corresponding gene products.

Annotations are also used to describe the content of documents in a standardized way. We distinguish between two types of documents: unstructured and semi-structured documents. We assume that the content of unstructured documents is represented by *named entities*. A *named entity* is a real-world object such as a person, organization, product, drug, or therapy that occurs as a phrase in a document. For instance, the following sentence consists of the named entities *autosomal dominant neurohypophyseal diabetes insipidus (ADNDI)* and *inherited disease* representing diseases:

> *Autosomal dominant neurohypophyseal diabetes insipidus (ADNDI) is an inherited disease caused by progressive degeneration of the magnocellular neurons of the hypothalamus leading to decreased ability to produce the hormone arginine vasopressin (AVP).*

Different named entities can be extracted depending on the application, e.g., *hormone arginine vasopressin (AVP)*. The representations of named entities can differ, which complicates the comparability among documents regarding the content. For instance, the named entity *autosomal dominant neurohypophyseal diabetes insipidus (ADNDI)* can also be represented with the phrase *pituitary diabetes insipidus*. To unify different representations of named entities, they are annotated with concepts from an ontology. However, to annotate named entities, they must be identified in unstructured documents using named entity recognition approaches before. Electronic health records, publications, or abstracts are examples of unstructured documents.

Semi-structured documents consist of separated paragraphs, such as XML documents. In the medical domain, case report forms (CRF) or questionnaires are used to recruit probands for clinical trials. These forms are structured by a set of separated questions about eligibility criteria or quality assurance of medical

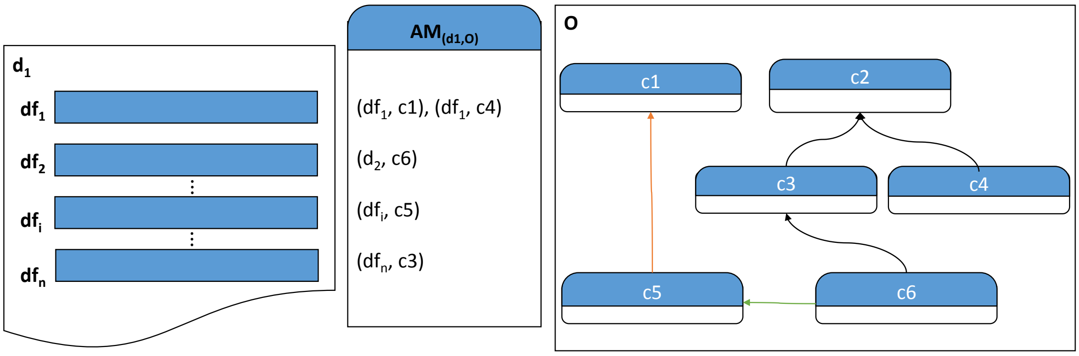


Figure 2.2: Annotation mapping model schema for a document d_1 with document fragments $df_1, ..., df_n$ and an ontology O.

services. Each question represents a semantic unit that can be annotated with concepts from an ontology. The semantic unit of a semi-structured document d is called as document fragment df. A named entity can also be seen as a document fragment df so that the goal of an annotation approach can be formally defined as follows.

The goal of an annotation approach is the identification of an *annotation mapping* $AM_{d,O} = \{(df_k, c_m) | df_k \in d \wedge c_m \in O\}$ for a document d and an ontology O. Figure 2.2 shows an abstract example of an annotation mapping for one document d_1 with document fragments $df_1, ...df_4$ and an ontology O. The annotation mapping $AM_{d,O}$ consists of pairs (df_k, c_m) of a document fragment $df_k \in d$ and a concept $c_k \in C_O$ such as (df_1, c_1) in the example.

2.1.2 General Approach

Annotated documents enable interoperability, standardized analysis, and effective question answering systems. Due to the vast diversity of document types covering domains of general topics to domain-specific documents, e.g., electronic health records, a general approach does not exist that results in the best quality considering all domains. In the following, the annotation process is discussed.

Document fragments and a knowledge base such as an ontology are the input of an annotation process. Figure 2.3 shows the complete method from extracting document fragments of a document to annotated documents with concepts

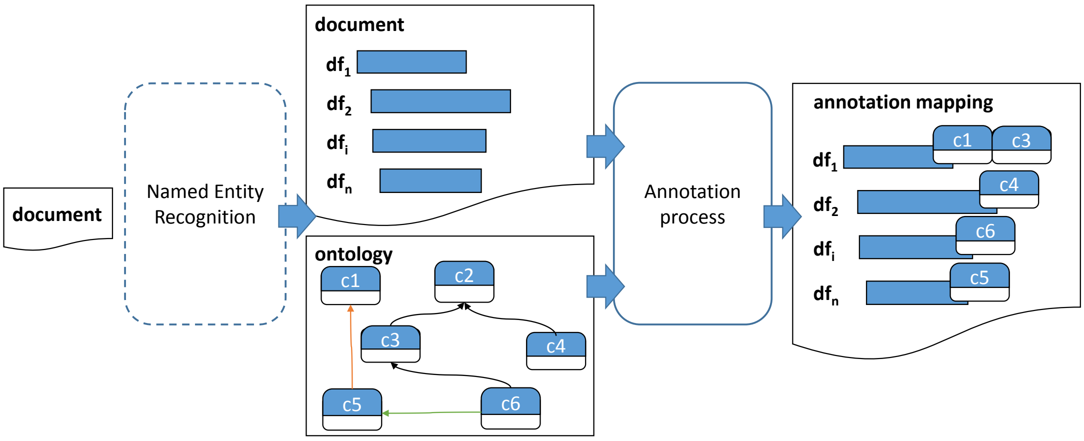


Figure 2.3: Annotation process for a document with concepts from an ontology. The named entity recognition step(dashed box) is optional if the document is already split into document fragments.

from an ontology. This process consists of the annotation process. If document fragments are not available, named entities must be identified applying a *named entity recognition* (NER) approach [95], to annotate unstructured documents. The resulting named entities that also represent document fragments are linked to concepts or entities from an ontology, respectively, Wikipedia. In general domains, Yago [130] and DBPedia [5] are often used as an ontology.

Named Entity Recognition

The goal of a NER approach is to identify text phrases representing named entities within a text document. The methods range from handcrafted rules to supervised learning techniques utilizing features and dictionaries. The features characterize a word such as the number of characters, number of digits, part of speech, morphology, and document features such as the total number of occurrences and positions in a sentence or paragraph. These features can be used to train a model such as hidden Markov models [133], maximum entropy models [7], support vector machines, or classifier ensemble [93]. The general idea of hidden Markov models and maximum entropy models is to identify the most likely sequence of named entity classes for a sequence of words.

Annotation process

The annotation process, also called as entity linking, identifies an annotation mapping for a set of named entities or document fragments and an ontology. This thesis proposes in Chapter 3, Chapter 4 and Chapter 5 improvements for this process. Entity linking approaches consist of two main steps: *Candidate generation* and *Ranking* [121]. In the candidate generation step, the method determine for each entity or document fragment $df \in d$ a set of candidates $C_{df} = \{c_1, ..., c_k\}$. A candidate can be an entity from a knowledge base such as Wikipedia or a concept from an ontology.

Candidate Generation Different techniques for determining candidates are available such as dictionary or search engine based methods and surface form expansions. The majority of approaches rely on textual similarity comparisons between the document fragment and the name or synonyms of concepts. Dictionary-based techniques utilize the entries of knowledge bases such as Wikipedia, knowledge graphs, or ontologies. The majority of entity linking approaches link document fragments to Wikipedia entities. They utilize features of Wikipedia such as entity pages, redirect pages, disambiguation pages, and bold phrases to build an offline dictionary [13, 29, 71, 49]. Entity linking systems that link document fragments to an ontology utilize the ontology as a dictionary. A dictionary consists of $\langle key, value \rangle$ pairs where the key is the name of an entity or the identifier of a concept, and the value is a set of named entities synonyms, names as well as variations that refer to the entity respectively the concept.

Nevertheless, the dictionary-based method does not work if the named entity is an abbreviation, and the dictionary does not consist of the abbreviation. Therefore, one type of methods focuses on surface form expansion from the local document. The idea is to determine the expanded variation of the abbreviation. The expanded variation can be used for the dictionary lookup. Further entity linking systems [92, 34] utilizes search engines such as Google to determine candidates for named entities. The named entity mention with its short context represents a search query using the Google API. The resulting Web pages from Wikipedia are used as candidate entities.

The resulting candidates C_{df} for a document fragment df are ranked using context-independent as well as context-dependent features to select appropriate concepts as annotations. The determined ranking between entities and concept candidates represents the probability of how likely it is that a concept is the correct annotation of an entity. Context-independent features mainly utilize the textual surface form of document fragments and compare it with the synonyms and names of concepts using string similarity measures such as edit-distance or dice coefficient scores. Furthermore, many entity linking systems [49, 57] utilize the popularity of a concept regarding a specifically named entity mention. Formally, the popularity is the ratio regarding the number of annotations and the number of occurrences of a particular named entity mention. In contrast to that, context-dependent features consider the context of an entity mention and the context of a concept. The approaches determine a similarity between the context of a mention and the document associated with the entity.

Besides the textual context information, the coherence between concepts represents the semantic relatedness between the annotated mentions and the linked concepts. Two concepts are coherent if a relationship exists defined by the knowledge graph or ontology, or they often co-occur in other documents. For instance, the concept for a particular disease such as *osteoarthritis* is related to the concept representing the treating drug *Rofecoxib*. The coherence of concepts can be used to rank the candidates using the candidates of other mentions in the same document. Different approaches [50, 112, 122] utilize the link structure from Wikipedia and consider the number of documents where two concepts co-occur. The cooccurrence represents a semantic relatedness between concepts. The computed number is used to compute a coherence measure using Wikipedia Link-based Measure [90], Point-wise Mutual Information, or Jaccard similarity.

The effectiveness of these measures depends on the link structure of concepts. New concepts have few or no related concepts so that these measures cannot work well. The combination of different measures by using Wikipedia Link Measure, point-wise mutual information, and Jaccard similarity [17] overcomes such issues. ML techniques can be applied to determine an optimal combination.

2.1.3 Differences between generic and specific domains

Summarizing the annotation process, we emphasize the difference of annotating documents in the biomedical domain compared to generic domains. The majority of approaches focus on linking named entities to generic knowledge bases.

Zwicklbauer et al. [147] compared the application of entity linking systems in general domains as well as the biomedical domain.

DBpedia and Yago comprise a broad range of general entities. Nevertheless, concepts for certain domains occur rarely, such as specific diseases, drugs, or therapies, so that domain-specific vocabularies are necessary for annotating documents. These vocabularies, such as UMLS cover different aspects of the biomedical domain. Consequently, the vocabularies are enormous and consist of heterogeneous concepts ranging from gene functions to diseases. Depending on the documents to be annotated, this variety of concepts is also necessary but simultaneously makes the annotation process more difficult, as the probability of ambiguities is high.

Context information is essential for resolving ambiguous concepts. The assumption is that the quality of identifying annotations highly depends on the amount and the quality of context information. Concepts are described either extensionally or intensionally. Intensional information consists of the description, name variations, and relationships between concepts defined by the ontology. Furthermore, already existing annotated documents provide information about the usage of concepts as annotations. For instance, if a specific disease often co-occurs with a particular drug or therapy, the concepts occur in the unannotated documents.

Moreover, annotated documents provide context information such as the title as well as the topic of the document. In the biomedical domain, these information are limited compared to generic domains. For instance, Wikipedia provides many features such as redirect pages, disambiguation pages or description paragraphs of an entity. The description paragraph consists of further entities so that the current one is related to those. Consequently, entity linking systems in the biomedical domains have to address the lack of available annotated documents. Furthermore, they must challenge the heterogeneity of large vocabularies.

2.2 Entity Resolution

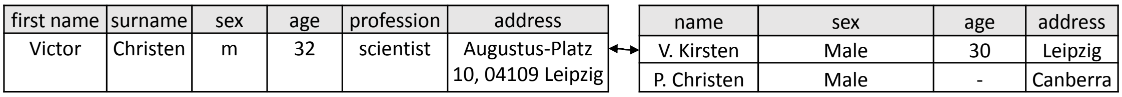
Entity resolution is the process of determining records from one or different data sources that represent the same real-world entity. Due to the autonomy of data sources, there exists no global identifier regarding the records of the data sources, so that an exact match of the identifiers is not sufficient. Moreover, each data source can be differently structured according to the attributes of records and the granularity of values. Figure 2.4 shows an example of two data sources consisting of records representing persons. Both data sources are differently structured, e.g., *first name* and *surname* are separately represented in the first data source whereas the name in the second data source is represented as the concatenation of these attributes.

Moreover, each data source's quality can be different concerning typing errors, data errors, or missing values. For instance, the first record of the second data source consists of a misspelled name "Kirsten" instead of "Christen" and the age is inconsistent.

Different approaches utilize comparisons between the attribute values of records to overcome such issues. The result of the comparisons represents similarities concerning the different characteristics. These similarities are utilized to determine a record pair as a match, non-match, or possible match. In the following, we give the problem definition in Subsection 2.2.1. After that we propose the general approach in Subsection 2.2.2.

2.2.1 Problem Definition

The goal of entity resolution is the identification of record pairs or clusters representing the same entity. The input of entity resolution is a set of different data sources $R_1, ...R_m$. Each data source R_k consists of records $r_1, ..., r_n$. The output is a set of pairs $\mathcal{M}_{(R_k,R_l)}$ or a set of clusters where each pair respectively cluster consists of records representing the same entity. Moreover, each record r_k is characterized by a set of attributes A. For instance, the first record of the left data source in Figure 2.4 is characterized by the attributes $firstname, surname, sex, age,$ $profession$ and $address$.



first name	surname	sex	age	profession	address
Victor	Christen	m	32	scientist	Augustus-Platz 10, 04109 Leipzig

name	sex	age	address
V. Kirsten	Male	30	Leipzig
P. Christen	Male	-	Canberra

Figure 2.4: Entity resolution example for person data.

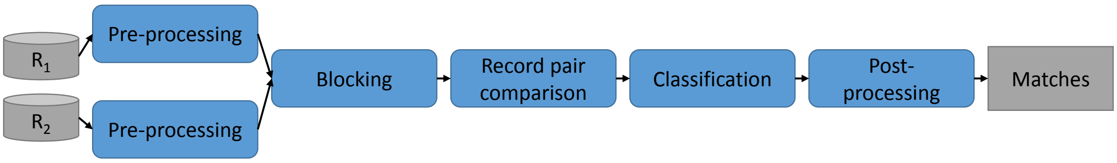


Figure 2.5: Entity resolution process to determine matches between two data sources.

2.2.2 General Approach

In this section, we describe the entity resolution process for two data sources [21]. This process consists of 5 tasks: preprocessing, blocking, record pair comparison, classification, and postprocessing shown in Figure 2.5. The preprocessing step transforms the data sources in the same format. The blocking step eliminates dissimilar record pairs to reduce the number of record comparisons. The resulting record pairs are compared using similarity functions and the attribute values of records. The classification step classifies each record as match or non-match based on the computed similarities for the different attributes. In the end, potential erroneous matches are removed by considering all computed matches globally. Moreover, new matches are added, assuming the transitivity of equivalence.

Preprocessing

Due to different formats, structure, and content, records are not comparable. The preprocessing step standardizes and cleans the attributes so that the content follows the same format, to ensure the comparability of records. The quality of the matching result highly depends on the standardization level [55], so that this step is a crucial part. The standardization consists of three steps: remove unwanted characters, misspelling correction, and abbreviation expansion. The textual information of attributes represents the content instead of punctuation symbols such as commas, colons, semicolons, periods, hashes, and quotes. Depending on the application, certain words are unnecessary for the semantic of a record such as stop words. Therefore, these characters and words are removed from the attribute values of each record. The match effectiveness depends on the data quality so that misspellings lead to a considerable variation that impacts the similarity computation. Common misspellings or name variations for personal data are replaced by standardized representations using dictionary-based approaches.

Blocking and Filtering

The identification of duplicates between two data sources, R_1 and R_2, potentially requires the evaluation of the Cartesian product consisting of all record pairs between R_1 and R_2. However, the comparison of all record pairs does not scale for huge data sources. For instance, two data sources consisting of 1,000,000 records results in 1,000,000,000,000. If 100,000 comparisons can be performed in 0.5 s, it would take 1388.89 h to compare these data sources. In general, the number of potential comparisons grows quadratically with the number of records. To avoid the evaluation of the Cartesian product, blocking and filtering techniques are applied to reduce the number of comparisons [22].

The idea of blocking is to group records into blocks $B = \{b_1, b_2, ...b_n\}$ based on a blocking key that is a composition of attributes or is derived from attribute values. The records that share the same or similar blocking key represent a block b_k, and the records are pairwise compared.

In contrast to blocking, filtering techniques reduce the number of comparisons by discarding dissimilar pairs with respect to a similarity function *sim* with a similarity threshold δ. Filtering techniques discard all record pairs (r_1, r_2) where $sim(r_1, r_2) < \delta$ holds.

The area of blocking approaches is divided into standard blocking [41], Q-gram indexing [128], suffix array indexing, Sorted-Neighborhood [65, 110], and meta-blocking [124, 105]. Standard blocking approaches map each record to a blocking key that is used as a key for an inverted index. A common approach for personal data is the usage of the Soundex algorithm [103] that encodes a word to a code based on its phonetic characteristics. Further approaches utilize the attribute values of selected attributes to generate a blocking key. For instance, the gender attribute of the example in Figure 2.4 results in two blocking keys male m and female persons f for the attribute *sex*. The main issues of blocking are the missing comparisons of records being true matches and the resulting number of comparisons.

Especially, for record pair comparisons in a parallel environment, inappropriate blocking keys lead to imbalanced groups so that the computational effort is unequally distributed regarding the processes working in parallel. Approaches [102, 64] aim to generate equal-sized blocks.

Candidate generation

The record comparison step computes different similarity functions for comparing attribute values of record pairs to determine a record pair as a match. Due to data quality issues, two records representing the same can consist of different attribute values so that an exact comparison is insufficient. Similarity functions represent an indication of how similar two attribute values are. A similarity function sim_k computes a value between 0 and 1 for attribute values of a record pair considering two attributes A_k and A_l. The candidate generation step of the annotation process in Subsection 2.2.2 uses similarity functions as well. A similarity of 0 represents that the attribute values are entirely different. In contrast, a similarity of 1 is an exact match between the attribute values.

Due to the variety of errors, information representations as well as data types, different similarity functions are available [21]. The majority of similarity functions are used for comparing textual attribute values. They are distinguished by edit-distance based comparisons and token-based string comparisons as well as hybrid similarity computations.

The general edit distance-based similarity is the Levenshtein distance [76]. This distance determines the number of operations such as insertions, deletions, and substitutions to convert a string s_1 to a string s_2. For instance, the number of edit operations between *Peter* and *Petr* is 1 since an 'e' must be removed from *Peter* to obtain *Petr*. The similarity can be computed by considering the number of edit operations normalized by the aggregated length of the two strings. A variant of the Levenshtein distance is the Smith-Waterman distance that is applied for sequence alignments in the biomedical domain. This distance allows gaps and different weights according to the type of edits.

Token-based similarities split a string s into sets of tokens T_k. A token t can be a word or a substring of a certain length q that is called q-gram. Q-grams are generated by a sliding window of size q. Special characters are added to weight the start and the end of a string equally. Since the length of a string can be smaller than the size of the window, the string is extended with special characters at the beginning and the end of the string. The similarity can be computed in different ways:

- overlap coefficient: $sim_{overlap}(s_k, s_l) = \frac{|T_k \cap T_l|}{min(|T_k|,|T_l|)}$
- jaccard coefficient: $sim_{jaccard}(s_k, s_l) = \frac{|T_k \cap T_l|}{|T_k \cup T_l|}$
- dice coefficient: $sim_{dice}(s_k, s_l) = 2 \cdot \frac{|T_k \cap T_l|}{|T_k|+|T_l|}$

Moreover, each token t can be weighted using TFIDF weights $w_{t,s}$. This value considers the frequency of a token t within an attribute value of a record as well as across all values of all records from all data sources. The weight $w_{t,s}$ is computed by the term frequency tf for a token t in a string s and the logarithm of the ratio between the total number of strings and the number of occurrences of token t in all strings S shown in Equation 2.1.

$$w_{t,s} = tf(w_{t,s}) \cdot log \frac{|S|}{|\{s' \in S | t \in s'\}|} \tag{2.1}$$

The similarity sim between two strings s_k and s_l is determined by computing the cosine similarity based on the vectors $\vec{s_k}$ and $\vec{s_l}$. Each entry s^i of a vector $\vec{s}$ represents a token t_i across all attribute values. The entry is equal to the weight $w_{t_i,s}$ if $t_i \in s$, otherwise it is equal to zero. The resulting vectors are used to compute the cosine similarity defined as follows:

$$cosine(s_k, s_l) = \frac{\vec{s_k} \cdot \vec{s_l}}{||\vec{s_l}|| \cdot ||\vec{s_k}||}$$

where $||\vec{s_l}||$ is the length of a vector s_l.

In addition, hybrid functions combine two different similarity functions. For instance, Soft-TFIDF determines for two strings s_l and s_k the cosine similarity considering similar tokens as well. To consider similar tokens, the set $CLOSE(\delta, T_k, T_l)$ consists of tokens $t_k \in T_k$ that are similar to at least one $t_l \in T_l$ regarding a similarity function sim', so that $sim'(t_k, t_l) \geq \delta$ holds. T_k and T_l are sets of words from attribute values. The cosine similarity is computed as follows:

$$sim_{softtfidf}(s_k, s_l) = \sum_{t \in CLOSE(\delta, T_k, T_l)} w(t, s_k) \cdot w(t, s_l) \cdot max(\{sim'(t, t_l) | t_l \in T_l\}) \tag{2.2}$$

Besides the attribute-based similarity functions for comparing two records, context-based similarity functions utilize the relationships between records [8]. The resulting graph consists of records as vertices and semantic relationships or hierarchies as edges. The neighborhood of a record to related records represent the context that is used to determine a context similarity. To compute the context similarity between two records r_1 and r_2, the overlap between neighborhoods $N(r_1)$ respectively $N(r_2)$ are considered. Similarly to the dice or Jaccard coefficient, the context similarity can be computed using the record sets represented by the neighborhoods. Collective entity resolution and group linkage approaches utilize context similarity functions in addition to attribute-based approaches [73, 44, 70]. The idea of collective entity resolution approaches is to select iteratively "safe" matches based on the hybrid similarity function. These matches are used to compute the context similarity based on overlapping neighborhoods. We discuss collective entity resolution approaches in Chapter 6 in more detail.

Classification

The goal of entity resolution is the identification of matches $m \in \mathcal{M}$. The identification of duplicates is determined by using the similarities computed in the record comparison step. A classification model can be either manual or automatized generated. A simple classification model computes a weighted sum over all similarities for a record pair r_k and r_l shown in Equation 2.3 and classifies a record pair as a match that is above a manually defined similarity threshold δ. The weights w_i represent the importance of an attribute comparison.

$$\sum_{1 \leq i \leq n} w_i \cdot sim_i(r_k, r_l) \geq \delta \tag{2.3}$$

The sum of all weights must be one so that the aggregated similarity ranges from 0 to 1. Further similarity combinations are *max* and *min* functions for a set of similarities. The *max* and *min* aggregation function computes the maximum, respectively, minimum similarity for a set of similarities regarding a record pair r_k and r_l. Domain knowledge and expertise are necessary to determine the weights and similarity combinations for classification. Moreover, a manual configuration is a time consuming and erroneous task. Therefore, automatized approaches for generating classification models are essential. Machine learning techniques are utilized to automatize the generation of models. The general idea of machine learning is to generate a model M based on training data T that represent the problem. The generated model M can classify unseen pairs. The quality of classification depends on the model and implicitly on the training data. The majority of ML-based entity resolution approaches use training data consisting of similarity vectors $\vec{w}$, where a vector represents a record pair. Each entry of a similarity vector $\vec{w}$ represents the result of a similarity function regarding an attribute comparison. Moreover, each vector is labeled as *match* or *non-match*. A generated model classifies a record pair based on a similarity vector with a certain *confidence*.

A crucial part of machine learning is the generation of training data. On the one hand, the generation of training data is a time-consuming task so that the number of manually verified data should be small. On the other hand, the training data must be representative so that the trained model generalizes and hence does not overfit. In this context, overfitting means that the model not only classifies the training data well but also unclassified record pairs. Active learning approaches [39] focus on reducing the amount of training data where the selected training data is representative. In the context of entity resolution, the number of to be classified pairs as match and non-match is reduced. We discuss different active learning approaches for entity resolution in Chapter 7.

Postprocessing

In the postprocessing step, the final set of matches $\mathcal{M}_{(R_i,R_k)}$ between two data sources R_i and R_k is determined. The majority of approaches assumes that the data sources are duplicate free. A similarity graph is utilized to determine inconsistent record pairs. This graph consists of records as vertices and matches as edges. Each edge is weighted by an aggregated similarity computed in the *candidate generation* step(see Section 2.2.2) or the confidence of a machine learning model determined in the *classification* step(see Section 2.2.2). Consequently, a record mapping is inconsistent if a connected component consists of records from the same data source. A connected component is a subgraph where each vertex is reachable from an arbitrary vertex of the connected component. A postprocessing method removes edges so that the resulting graph is consistent, and the aggregated similarity of removed record pairs is small.

The approaches can be divided into global and local selection strategies. Local selection strategies consider the similarities between pairs for a particular record. In contrast, global strategies aim to minimize the total similarity of removed edges considering all record pairs. Franke et. al [43] gives an overview of different selection strategies for two data sources in the privacy-preserving record linkage context that is also applicable to entity resolution. Further approaches [117, 100, 97] focus on resolving conflicts in multi-source record linkage problems.

2.3 Quality Measurements

The result of an annotation approach, as well as an entity resolution method, is a set of pairs of document fragments df and concepts c, respectively of records from data sources R_i and R_k. To measure the quality of different approaches to decide which approach should be used in an application, a measure for the quality is necessary. Therefore, we use in our experiments *Precision*, *Recall* and *F-Measure*. Precision and Recall are determined by computing the ratio between *true positives* (TP) and all pairs determined by the methods, respectively, all existing pairs. The set of true positives consists of all pairs that are correctly determined. The set of *false positives* (FP) comprises the pairs that are incorrectly identified as annotations or matches by the method. The set of *false negatives* consists of pairs being not identified, even if they are correct annotations or matches. Figure 2.6 shows the relationship between true positives, true negatives, false positives and false negatives. The left red rectangle represents the set of identified annotations or matches by an approach, and the circle comprises all annotation or matches that exist in reality. The true negative pairs represent the correctly identified pairs being not an annotation or match. Precision and Recall are computed as follows:

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

Precision and Recall are contrary measures since a method can achieve a high precision if it is very restrictive, but the resulting Recall will be potentially very small. In contrast to that, a method can classify each candidate pair as correct so

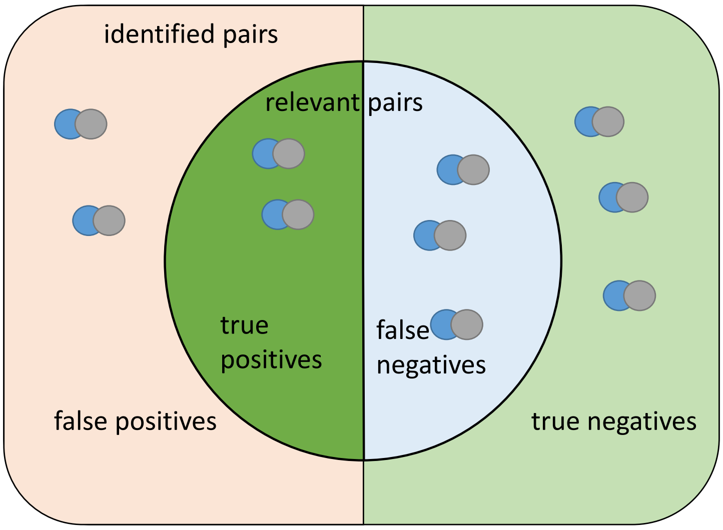


Figure 2.6: Representation of true positives, false positives, false negatives and true negatives. The circle pairs represent the annotations, respectively, the record pairs.

that the Recall is high, but this leads probably to a decreasing precision. Therefore, the F-Measure is used as the harmonic mean between both measures that is computed as follows:

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.4}$$

The F-Measure is used as quality measurement for the developed methods and for the comparison with existing approaches.

2.4 Conclusion

This chapter described the different processes: semantic annotation and entity resolution. The semantic annotation generates an annotation mapping for an ontology and a set of document fragments described in Subsection 2.1.1. The annotation process and the named entity recognition process are parts of the extraction and enrichment task for documents. Subsection 2.2.2 described both processes generally and referred to standard techniques in the related work.

The second part of the background focused on the entity resolution process. The Section 2.2 consists of the problem definition and the description of the general process with its steps. The description of each step includes techniques and methods from the current research.

Both processes aim to generate qualitative results. Therefore, Section 2.3 consists of the quality measurements being used to evaluate the proposed methods.

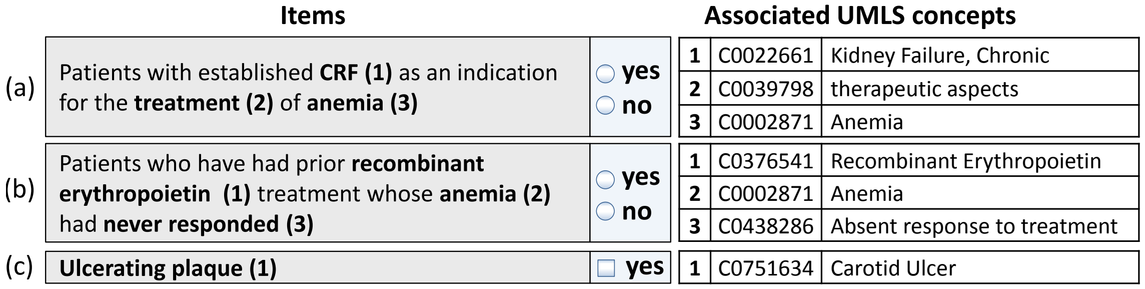
Part II

Medical Entity Linking

3 AnnoMap: Annotation of medical forms

Preamble

This chapter is based on [27]. AnnoMap is an annotation tool for annotating structured medical documents such as case report forms. This tool provides a set of textual similarity functions to determine annotation candidates. Moreover, it consists of a group-based selection strategy to resolve conflicts if a phrase is annotated with multiple concepts. To reduce the number of candidates, a semantic blocking approach was introduced that is based on topic detection. The basic workflow of AnnoMap was presented at the DILS 2015 conference with the following publication in the conference proceedings.



	Items		Associated UMLS concepts		
(a)	Patients with established **CRF (1)** as an indication for the **treatment (2)** of **anemia (3)**	○ **yes** ○ **no**	**1**	C0022661	Kidney Failure, Chronic
			2	C0039798	therapeutic aspects
			3	C0002871	Anemia
(b)	Patients who have had prior **recombinant erythropoietin (1)** treatment whose **anemia (2)** had **never responded (3)**	○ **yes** ○ **no**	**1**	C0376541	Recombinant Erythropoietin
			2	C0002871	Anemia
			3	C0438286	Absent response to treatment
(c)	**Ulcerating plaque (1)**	□ **yes**	**1**	C0751634	Carotid Ulcer

Figure 3.1: Example medical form items and associated annotations to UMLS concepts. (CRF = '*Chronic Renal Failure*' = '*Chronic Kidney Failure*').

3.1 Motivation

Medical forms are frequently used to document patient data within electronic health records (EHRs) or to collect relevant data for clinical trials. For instance, case report forms (CRFs) ask for different eligibility criteria to include or exclude probands of a study or to document the medical history of patients. Currently, there are more than 180,000 studies registered on http://clinicaltrials.gov and every clinical trial requires numerous CRFs for data collection. Often these forms are created from scratch without considering existing CRFs from previous trials. Thus, there is a huge amount and diversity of existing medical forms until now, and this number will increase further. As a consequence, different forms can be highly heterogeneous impeding the interoperability and data exchange between different clinical trials and research applications.

To overcome such issues, it is important to annotate medical forms with concepts of standardized vocabularies such as ontologies [35]. In the biomedical domain, annotations are frequently used to semantically enrich real-world objects. For instance, the well-known Gene Ontology (GO) is used to describe molecular functions of genes and proteins [59], scientific publications in PubMed are annotated with concepts of the Medical Subject Headings (MeSH) [83], and concepts of SNOMED CT [33] are assigned to EHRs supporting clinical applications like diagnosis or treatment. These diverse use cases for annotations show that they can represent a variety of relationships between real-world objects improving semantic search and integration for comprehensive analysis tasks. In particular, ontology-based annotations of medical forms facilitate the identification of similar questions (items) and commonly used medical concepts. Well-annotated items can be re-used to design new forms avoiding an expensive re-definition in every clinical trial. Moreover, the integration of results from different trials will be improved due to better compatibility of annotated forms. Beside clinical trials, also other medical applications like routine documentation in hospitals can profit from form annotation. For instance, the fusion of two or more hospitals requires the integration of hospital data which will be less complex if data semantics are well-defined due to the use of ontology-based annotations.

The open-access platform *Medical Data Models* (MDM)[1] already aims at creating, analyzing, sharing and reusing medical forms in a central metadata repository [11].

Currently, MDM provides more than 9,000 medical form versions and over 300,000 items. Beside overcoming technical heterogeneities (e.g. different formats), MDM intends to semantically enrich the medical forms with concepts of the widely used Metathesaurus of the Unified Medical Language System (UMLS) [9], a huge integrated data source covering more than 100 different biomedical vocabularies. So far, medical experts could assign UMLS concepts to items of some medical forms in MDM, but many forms have no or only preliminary annotations. However, such a manual annotation process is a very time-consuming task considering the high number of available forms within and beyond MDM as well as the huge size of UMLS ($>$ 2.8 Mio. concepts). Thus, it is a crucial aim to develop automatic annotation methods supporting human annotators with recommendations.

The automatic annotation of medical forms is challenging since questions are written in free text, use different synonyms for the same semantics and can cover several different medical concepts. Moreover, the huge size of UMLS makes it difficult to identify correct medical concepts. So far, there has been some research on processing and annotation of different kinds of medical texts (e.g. [115, 52, 82]). However, (semi-) automatic annotation of medical forms has only rarely been studied (see Related Work in Section 3.8). We propose a solution to semi-automatically annotate medical forms with UMLS concepts and make the following contributions:

- We first discuss the challenges to be addressed for automatically annotating items in medical forms (Section 3.2).

[1] www.medical-data-models.org/?locale=en

- We propose an annotation workflow to automatically assign UMLS concepts to items of medical forms. The workflow encompasses three phases: a novel semantic blocking to reduce the search space, a matching phase and a postprocessing phase employing a novel grouping method to finally select the correct annotations. (Section 3.3).
- We evaluate our approaches based on reference mappings between MDM forms and UMLS. Results reveal that we are able to annotate medical forms in a largely automatic way. We further manually verify recommended annotations and present results for this semi-automatic annotation (Section 3.7).

Finally, we discuss related work in Section 3.8 and conclude in Section 3.9.

3.2 Challenges

The automatic annotation of medical forms requires first of all the correct identification of medical concepts in form items. Figure 3.1 illustrates three annotated items: (a) and (b) ask for eligibility criteria for a study w.r.t. anemia, and item (c) asks for the abnormality 'ulcerating plaque' in the context of a quality assurance form. An item consists of the actual question and a response field or list of answer options. In our example, question (c) has one annotation, whereas (a) and (b) are annotated with three UMLS concepts. Thus, one form item can address several different aspects like diseases (e.g. CRF, anemia), treatments or a patient's response to a treatment. In the following we discuss general challenges that need to be addressed during the annotation process.

Natural language items: Typically, a form consists of a set of items. Questions can be short phrases like in item (c) or longer sentences written in free text (Figure 3.1 (a), (b)). It is a difficult task to correctly identify medical concepts in these natural language sentences. Moreover, the use of different synonyms complicate a correct annotation, e.g. in Figure 3.1(a) '*CRF*' (= *Chronic Renal Failure*) needs to be assigned to C0022661 ('*Kidney Failure, Chronic*'). Simple string matching methods are not sufficient to generate annotations of high quality for medical form items. We will thus apply NLP (natural language processing) techniques such as named entity recognition and document-based similarity measures like TF/IDF to identify meaningful medical concepts that can be mapped to UMLS.

Complex mappings: Every question can contain several medical concepts and one UMLS concept might be mapped to more than one question. In our example in Figure 3.1 three UMLS concepts need to be assigned to questions (a) and (b) and the concept '*anemia*' occurs in both questions. By contrast, question (c) is only annotated with one concept. Thus, we might need to identify complex N:M mappings and do not know a priori how many medical concepts need to be tagged to one item. Conventional match techniques often focus on the identification of 1:1 mappings, but solely assigning one source concept to one target concept is a much simpler task. We thus need to develop sophisticated match techniques to correctly annotate items with several UMLS concepts.

Number and size of data sources: There is high number of forms (e.g. 9000 only in MDM) that need be to annotated and every form can contain tens to hundreds of items. Moreover, UMLS Metathesaurus is a very large biomedical data source covering more than 2.8 million concepts. Matching 100 forms each comprising only 10 items to the whole UMLS would already require 2.8 billion comparisons. On the one hand this leads to serious issues w.r.t. memory consumption and execution time. On the other hand it is extremely hard to identify correct annotations in such a huge search space. It is thus essential to apply suitable blocking schemes to reduce the search space and restrict automatic annotation to the most relevant subset of UMLS.

Instances: Form items are not only characterized by medical concepts in the actual question but also by its possible instances or response options. Item answers have a data type (e.g. Boolean "'yes/no"' in Figure 3.1) and might be associated with value scales (e.g. between 1 and 5) or specific units (e.g. mg, ml). Often possible answers are restricted to a list of values (e.g. a list of symptoms). To improve the comparability of different forms, such instance information should be semantically annotated with concepts of standardized terminologies. In this paper, we focus on the annotation of item questions but see a correct annotation of answer options as an important future challenge.

In summary, the automatic identification of high-quality annotations for medical forms is a difficult task. However, studying automatic annotation is very useful to support human experts with recommendations. For a semi-automatic annotation

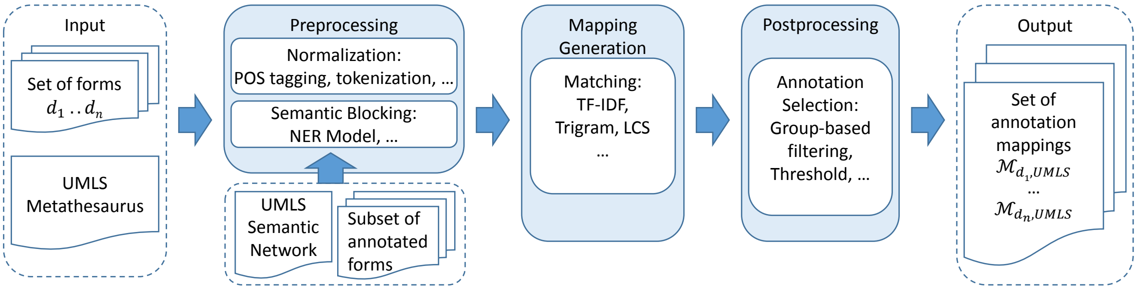

Input
Set of forms $d_1 .. d_n$
UMLS Metathesaurus
Preprocessing
Normalization: POS tagging, tokenization, ...
Semantic Blocking: NER Model, ...
UMLS Semantic Network
Subset of annotated forms
Mapping Generation
Matching: TF-IDF, Trigram, LCS ...
Postprocessing
Annotation Selection: Group-based filtering, Threshold, ...
Output
Set of annotation mappings
$\mathcal{M}_{d_1,UMLS}$
...
$\mathcal{M}_{d_n,UMLS}$

Figure 3.2: Overview of the annotation workflow.

process it is especially important to identify a high number of correct annotations without generating too many false positives. Thus, achieving high recall values is a major goal while precision should not be too low, since the number of presented recommendations should be manageable for human experts. Moreover, a fast computation of annotation candidates is desirable to support an interactive annotation process. To address these challenges, we present a workflow for semi-automatic annotation of medical forms in the following.

3.3 Annotation Workflow

Our annotation workflow semantically enriches a set of medical forms by assigning UMLS concepts to form questions. An annotation is an association between a question and an UMLS concept. UMLS concepts are identified by their *Concept Unique Identifiers* (CUI) and are further described by attributes like a preferred name or synonyms. To identify annotations for a given medical form D, we determine a mapping $\mathcal{M}$ between the set of form questions that can be treated as document fragments $D = \{df_1, df_2, ..., df_k\}$ and the set of UMLS concepts $UMLS = \{c_1, c_2, ..., c_m\}$ where $UMLS$ represents the ontology. Moreover, we extend the annotation definition from subsection Section 2.1.1 by a score sim. The mapping covers a set of annotations and is defined as:

$$\mathcal{M}_{D,UMLS} = \{(df, c, sim) | df \in D, c \in UMLS, sim \in [0,1]\}.$$

A question df in a form d is annotated with a concept c from $UMLS$. Our automatic annotation method computes a similarity value sim indicating the strength of a connection. Greater sim values denote a higher similarity between the question and the annotated concept. Our annotation workflow (see Figure 3.2) consists of three main phases that address the challenges discussed in Section 3.2. The input is a set of medical forms $d_1, \ldots, d_n$ each comprising a set of item questions as well as the UMLS Metathesaurus. During preprocessing we further use the UMLS Semantic Network and a subset of annotated forms. The output is a set of annotation mappings $\mathcal{M}_{d_1,UMLS}, \ldots, \mathcal{M}_{d_n,UMLS}$.

- In the *Preprocessing* phase we normalize input questions and UMLS concepts. Since a medical form is usually only associated to some domains covered by UMLS, we develop a novel semantic blocking technique to identify relevant concepts for the annotation generation. The approach is training-based and involves semantic types of UMLS concepts.

- In the *Mapping Generation* phase we identify annotations by matching the questions to names and synonyms of relevant UMLS concepts. We use a combination of a document retrieval method (*TF/IDF*) and classic match techniques (*Trigram, LCS (Longest Common Substring)*). By doing so we are able to identify complex annotation mappings for long natural language sentences as well as annotations to single concepts for shorter questions.

- During *Postprocessing* we remove probably wrong annotations to obtain a manageable set of relevant annotations for expert validation. Beside threshold selection we apply a novel group-based filtering to address the fact that questions might cover several medical concepts. For each question, we cluster similar concepts and keep only the best matching one per group.

Our workflow generates annotation recommendations which should be verified by domain experts since automatic approaches can not guarantee a correct annotation for all items. In the following, we discuss the methods in more detail.

3.4 Preprocessing

During preprocessing, we normalize the questions of a medical form as well as names and synonyms of UMLS concepts. In particular, we transform all string

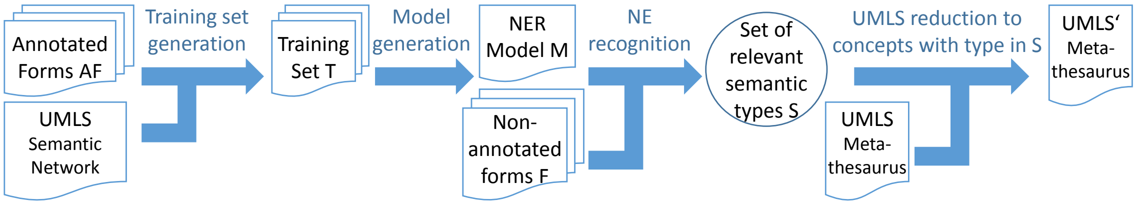


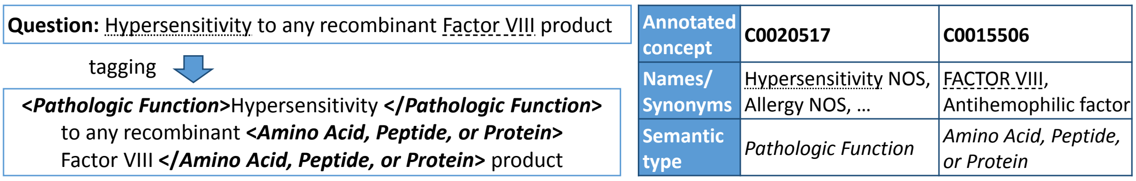
Figure 3.3: Semantic blocking workflow.

values to lower case and remove delimiters. We then remove potentially irrelevant parts of item questions. For instance, prepositions or verbs are typically part of natural language sentences, however they rarely cover information on medical concepts. We therefore apply a part-of-speech (POS) tagger[2] and keep only nouns, adjectives, adverbs and numbers/cardinals. We tokenize all strings into trigrams and word-tokens for the later annotation generation.

We further apply a semantic blocking to reduce the size of UMLS. UMLS Metathesaurus is a huge data source covering a lot of different subdomains. However, medical forms are usually only associated to a part of UMLS such that a comparison to the whole Metathesaurus should be avoided. We therefore aim at reducing UMLS by removing concepts that are probably not relevant for the annotation process. Our semantic blocking technique involves the UMLS Semantic Network. It covers 133 different semantic types and every UMLS concept is associated to at least one of the types. Our blocking technique follows a training-based approach and uses Named Entity Recognition (NER) to identify relevant semantic types for item questions. The general procedure is depicted in Figure 3.3.

First, we build a training set T based on a subset of manually annotated forms AF. For each question in AF, we identify annotated named entities. Therefore, we compute the longest common part between a question and the names/synonyms of its annotated UMLS concepts. We then tag the identified question parts with the semantic types of the corresponding UMLS concept. Figure 3.4 illustrates an example for the training set generation. The given question is annotated with two UMLS concepts. The longest common part of the question and the concept *C0020517* is *Hypersensitivity*, while *C0015506* corresponds to the question part *Factor VIII*. Thus, *Hypersensitivity* is tagged with the semantic type of *C0020517* (*'Pathologic Function'*) and *Factor VIII* is labeled with *'Amino Acid, Peptide, or Pro-*

[2] http://nlp.stanford.edu/software/tagger.shtml



Question: Hypersensitivity to any recombinant Factor VIII product

<Pathologic Function>Hypersensitivity ***</Pathologic Function>*** to any recombinant ***<Amino Acid, Peptide, or Protein>*** Factor VIII ***</Amino Acid, Peptide, or Protein>*** product

Annotated concept	C0020517	C0015506
Names/ Synonyms	Hypersensitivity NOS, Allergy NOS, …	FACTOR VIII, Antihemophilic factor
Semantic type	*Pathologic Function*	*Amino Acid, Peptide, or Protein*

Figure 3.4: Training set generation: example for tagging a question with semantic types.

tein′. Based on the tagged training set T of forms AF we learn a NER-model M using the Open-NLP framework[3]. Our semantic blocking (see Figure 3.3) then performs a named entity recognition using the model M to a non-annotated set of forms D. By doing so, we can recognize named entities for the questions in D and identify a set of relevant semantic types S. Finally, we reduce the UMLS Metathesaurus to those concepts that are associated to a semantic type in S and obtain the filtered $UMLS'$.

3.5 Candidate Generation

We generate annotation mappings between a set of medical forms $d_1, \ldots, d_n$ and the reduced $UMLS'$ using a combination of a document retrieval method (*TF/IDF*) and classic match techniques (*ExactMatch, Trigram, LCS*). These methods can complement each other such that we are able to identify complex annotation mappings for long natural language sentences as well as shorter questions covering only one concept. To generate annotations for each considered form, we compute similarities between all questions of a form and every concept in $UMLS'$. Note that, we tokenized strings during preprocessing. To enable an efficient matching, we encode every token (word or trigram), and compare integer instead of string values. Furthermore, we separate UMLS into smaller chunks and distribute match computations among several threads.

We apply for each question the three match methods. *Trigram* compares a question with concept names and synonyms, identifies overlapping trigram tokens, and computes similarities based on the Dice Metric. This is useful for shorter questions that slightly differ from the concept to be assigned. In our example in Figure 3.1 the annotation for item (c) *′Ulcerating plaque′* needs to be assigned to

[3] https://opennlp.apache.org/

the concept C0751634 (*'Carotid Ulcer'*). This correspondence can be identified by the synonym *'Carotid Artery Ulcerating Plaque'* of C0751634. Since there is only a partial overlap, it is feasible to identify the longest sequence of successive common word-tokens (*LCS*) between a question and a concept. *LCS* is also useful for complex matches when a question contains several medical concepts, e.g., 'recombinant erythropoietin' and 'anemia' in item (b) (Figure 3.1).

Moreover, we use *TF/IDF* to especially reward common, but infrequent tokens between questions and UMLS concepts. For instance, in medical forms the token 'patient' occurs essentially more often then 'erythropoietin'. Thus, the computed similarity value should be higher for matches of rarely occurring, meaningful tokens compared to frequent tokens that appear in many questions and concepts. We compute tf-idf values for each token w.r.t. a question and an UMLS concept. The term frequency (tf) denotes the frequency of a token within the considered question or concept while the inverse document frequency (idf) characterizes the general meaning of a token compared to the total set of tokens. The tf-idf values are then used to compute the similarity between a token vector of the question and a token vector for names and synonyms of an UMLS concept. We choose a hamming-distance based measure to compare two token vectors. We compute distances between tf-idf values of two token vectors and normalize it based on the vector length. The normalized distance is converted into a similarity value. We assign a smaller weight to the length of the longer vector to address cases, when one string consists of considerably more tokens than the other one, as this occurs for annotating long sentences. Thus, the measure does not penalize differences that are triggered by a differing vector length. High similarities between a shorter and a longer token vector can be achieved when a considerable number of meaningful tokens are contained in both vectors.

The generated annotation mappings are finally unified and similarities are aggregated by selecting the maximum *sim* value of a correspondence identified of several match methods to maximize the recall. Note that, we optimize the precision by performing the postprocessing phase. The match methods can identify overlapping results, but complement each other since they address different aspects of document and string similarity. We choose to adopt the three match methods in order to achieve a good recall by finding simple 1:1 as well as complex mappings for longer questions.

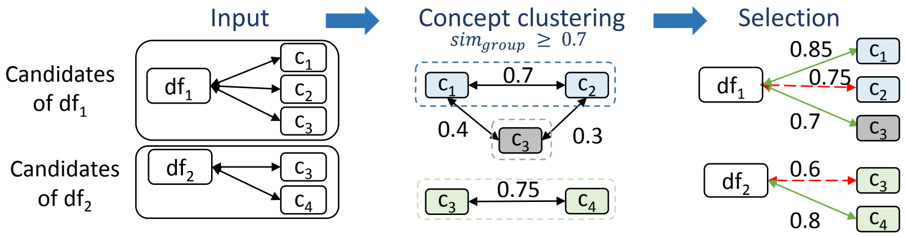


Figure 3.5: Group-based filtering for two questions df_1 and df_2 and their annotations to concepts c_{1-4}. Uniformly colored concepts represent a group of similar concepts.

3.6 Postprocessing

Beside a simple threshold filtering, we apply a more sophisticated postprocessing step to filter the generated annotation mapping. Our aim is to identify all annotations to a question that are likely to be correct, i.e. to obtain high recall values. However, the result should not contain too many false positives in order to obtain a manageable set of recommendations to be presented to human experts. This is a complicated task when questions cover more than one medical concept, i.e. when we need to identify complex mappings. A simple approach would be to select the top k similar concepts for each question. However, it is possible that several annotations for the same medical concept in a question are among the top k. A top k selection could eliminate all annotations of medical concepts with lower *sim* values. We therefore apply a novel group-based filtering.

The group-based filtering first clusters concepts that are likely to belong to the same medical concept and then selects the most similar concept within a group. Figure 3.5 exemplarily describes the overall procedure for two questions df_1 and df_2 and their annotations to several concepts. Given a set of annotations for a question, we compute similarities between all UMLS concepts that are annotated to a question using trigram matching on concept names and synonyms. We than cluster concepts in one group if their similarity exceeds the required sim_{group} threshold. In our example, we compare c_1, c_2 and c_3 for df_1, and identify two groups $(\{c_1, c_2\}, \{c_3\})$. c_1 and c_2 are very similar ($sim_{group} \geq 0.7$), while c_3 builds an own group. Finally, the best annotation per group is selected to be included in the final mapping based on the annotation similarities from the previous phase. For instance, we remove (df_1, c_2) due to the lower annotation similarity within its group. Applying a simple top 2 selection would have preserved (df_1, c_2) but removed (df_1, c_3), although (df_1, c_3) is likely to be the best match for a different medical concept covered by question df_1. Using the group-based filtering, we are able to keep one annotation for each medical concept in a question and thus allow for complex annotation mappings. In the following, we evaluate the proposed annotation methods for real-world medical forms.

3.7 Evaluation

To evaluate the proposed annotation workflow we consider three datasets covering medical forms from the MDM portal [11]. Table 3.1 gives an overview on the number of considered forms, the average number of items per form, the average number of tokens per item question and the average number of annotations per item. The first set of medical forms considers *eligibility criteria* (EC) that are used for patient recruitment in clinical trials w.r.t. diseases like Diabetes Mellitus or Epilepsy. The dataset covers 25 medical forms each comprising about 12 items on average. To recruit trial participants, a precise definition of inclusion and exclusion criteria is required, such that most questions are long natural language sentences (∼8 tokens on average) possibly covering several medical concepts.

Table 3.1: Overview of the used datasets.

Dataset	Eligibility Criteria (EC)	Quality Assurance (QA)	Top Items (TI)
#forms	25	23	1
avg(#items)	12.4	26.5	101
avg(#tokens)	8.3	3.3	2.4
avg(#anno.)	1.86	1.1	1.08

A correct identification of all annotations is very challenging for this dataset. Moreover, we consider medical forms for standardized *quality assurance* (*QA*) w.r.t. cardiovascular procedures. Since 2000 all German health service providers are obliged by law to apply these QA forms to prove the quality of their services [10]. The 23 QA forms contain about 27 items on average, but questions are shorter (∼3 tokens on average). We further consider a set of top items (TI) from the MDM portal. In [136], these items have been manually reduced to the relevant semantic question parts resulting in a low token number per question.

We handle the 101 top items as one medical form. For UMLS, we only consider concepts that possess a preferred name or term, which is the case for ∼1 Mio. UMLS concepts. We involve names and synonyms of these UMLS concepts.

To evaluate the quality of automatically generated annotation mappings we use reference mappings between all considered MDM forms and UMLS. Our team consists of computer scientists as well as medical experts (two physicians), such that we could manually create the reference mappings based on expert knowledge. We compute precision, recall and F-measure for the annotation mappings of every medical form and show average values for the respective dataset (EC, QA or TI). Note, that the average of F-measures is not equal to a harmonic mean of average precision and average recall. Since a manual annotation is a difficult and time-consuming task, the initial reference mappings might not be complete. We therefore follow a semi-automatic annotation approach and manually validate the automatically generated annotations for the QA dataset to find further correct annotations (see Subsection 3.7.4). We first show evaluation results for EC and QA w.r.t. the methods of our annotation workflow (Subsection 3.7.1,Subsection 3.7.2) and then give an overview on results for all datasets (Subsection 3.7.3)

3.7.1 Semantic Blocking

To evaluate our semantic blocking approach we measure the quality of the generated annotation mappings as well as matching execution times. We run experiments on an Intel i7-4770 3.4GHz machine with 4 cores. Our aim is to reduce execution times without affecting the recall. The generation of training data is an important step for the semantic blocking. So far, we generated training data by randomly selecting half of the manually annotated datasets. Note, that the training sets have some bias since we consider a special type of medical forms, namely eligibility criteria and quality assurance forms. However, it is feasible to choose relevant semantic types in UMLS based on form annotations in the considered domain. It is an interesting point for future work to study the training set generation for the semantic blocking in more detail. We evaluate the impact of the semantic blocking using a basic trigram matching (*Tri*) without group-based filtering (threshold $t = 0.8$). Figure 3.6 shows quality differences and execution

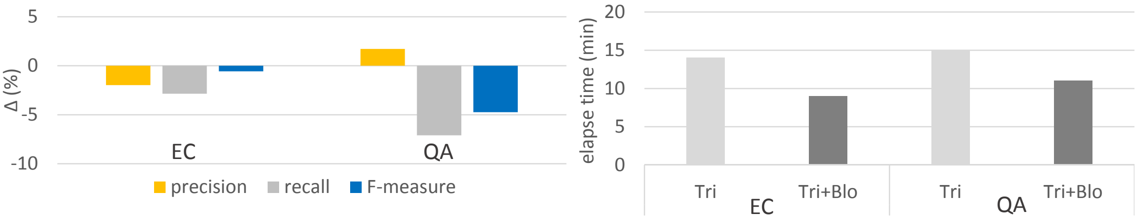


Figure 3.6: Semantic blocking: quality differences (left) and execution time (right) for QA and EC, comparison of trigram without (*Tri*) and with semantic blocking (*Tri+Blo*).

time results for QA and EC. The overall number of tokens was to small to apply the named entity recognition for TI. Applying the semantic blocking (*Blo*), UMLS could be reduced to ∼600.000 concepts. This results in good execution time reductions of $26 - 36\%$ for both datasets. However, we observe for each dataset a reduction of the quality of -0.5% for EC and -4.73% for QA. In both cases, the semantic blocking might be too restrictive by filtering some relevant UMLS concepts. A reason might be that the selection of our training set is not representative for the unannotated set of forms. We plan to further study the NER model generation to improve the blocking of UMLS concepts. Overall, our semantic blocking leads to good execution time reductions by fairly preserving recall values.

3.7.2 Matching and Group-based Filtering

We now generate annotation mappings by using a simple trigram matching (*Tri*), compare it to our combined match strategy based on TF/IDF, Trigram and LCS (*Comb*), and evaluate the impact of the group-based filtering (*Group*) for the QA dataset (see Figure 3.7). We disable the blocking for this experiment and consider different threshold settings to evaluate the annotation quality. The combined match approach leads to higher recall values for all thresholds compared to trigram, since *Comb* detects a higher number of correct annotations compared to the single matcher. In particular, the combined matching achieves the best recall of ∼66% ($t = 0.6$) which is 17% more than for trigram. Trigram is more restrictive and results in higher precision values, such that the overall F-measure is better for low thresholds. In general, increasing the threshold improves the overall annotation quality due to a higher precision, e.g. for $t = 0.8$ the F-measure is 15% higher than for $t = 0.6$ (*Comb*). However, we want to find a high number of correct

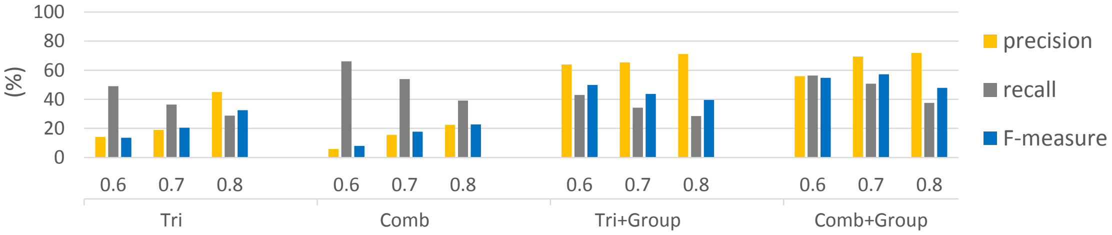


Figure 3.7: Quality evaluation: comparison of trigram (*Tri*), combined matching (*Comb*) and group-based filtering (*Tri+Group* and *Comb+Group*) for QA forms.

annotations (high recall) during the annotation generation phase. Therefore, we then filter wrong correspondences using our group-based selection strategy (Figure 3.7 right). This leads to significantly improved precision values and preserves the high recall. Since the combined match strategy results in higher recall values than the trigram matching, the F-measure values of the combined match strategy with the group-based selection (*Comb+Group*) are better than the trigram matching with the group-based selection (*Tri+Group*). For $t = 0.7$, we achieve the best average F-measure of 57% for the QA dataset. Thus, the group-based filtering is a valuable selection strategy to remove wrong but keep correct annotations.

3.7.3 Result Summary

To give a result overview w.r.t. the annotation quality, we show average F-measure values for all datasets (EC, QA, TI) in Figure 3.8. Since the semantic blocking decrease the quality, we compare the trigram matching (*Tri*), trigram matching with group-based filtering (*Tri+Group*) and combined matching with group-based filtering(*Comb+Group*) Due to a different amount of free text within the datasets, a uniform threshold not results in the best quality for each dataset, e.g., the TI dataset consists of mostly two words per item compared to the QA and EC dataset which have mostly more than three words per item. Therefore, we calculate the average for the thresholds 0.6, 0.7 and 0.8. The vertical lines indicate the minimum and the maximum F-measure values for the underlying thresholds. We observe for each dataset an increasing of F-measure by applying group-based filtering compared to trigram matching. The precision increases heavily while most correct annotations are preserved. Since the combined matching strategy results in higher recall values than the trigram matching, the combination with group fil-

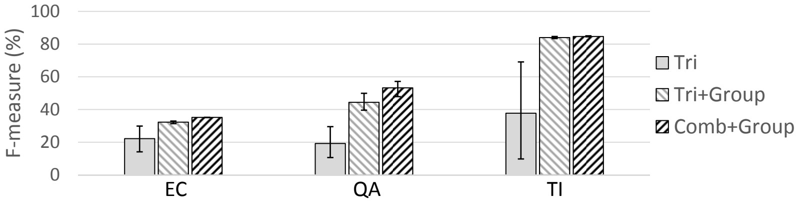


Figure 3.8: Comparison of effectiveness of the combined matching strategy and group-based filtering approach for each dataset.

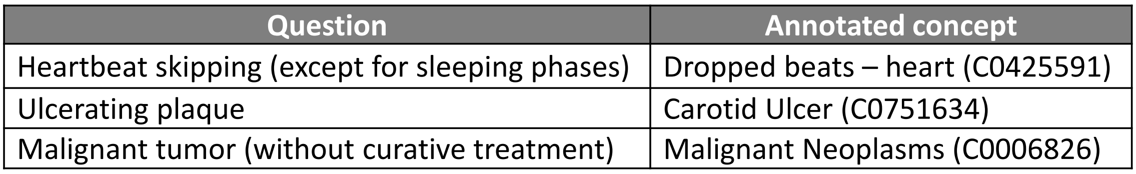
tering leads to better F-measure values such that the difference of best F-measure values is ∼3%(EC), ∼7%(QA) and ∼0.5%(TI). We achieve the best F-measure of ∼85% for TI followed by ∼57% for QA and ∼35% for EC.

The automatic annotation of the EC dataset showed to be very difficult, since EC contains items with specifically long natural language sentences covering an unknown number of medical concepts. The annotation of QA forms leads to better results, but still needs improvement. For the annotation of the top items (TI) we achieve very good results. These items have been manually reduced to the relevant medical terms having a positive impact on the automatic assignment of UMLS concepts for this dataset. The semantic blocking was valuable to reduce executions times, and the combined match strategy together with the group-based filtering showed to produce very good results compared to a simple trigram matching. Overall, the automatic annotation of medical forms is a challenging task and requires future research, e.g. to further improve the recall.

3.7.4 Validation

We applied a semi-automatic annotation for the QA dataset by manually validating recommendations generated by our automatic annotation workflow. We computed mappings for all 23 QA forms using semantic blocking, combined matching and group-based filtering. For every form and question, we presented the expected correct annotations as well as our recommendations, and highlighted false negatives, false positives and true positives.

Medical experts could identify 213 new correct annotations out of the set of false positives. We further found 5 wrong annotations in the reference mappings based on our automatically generated recommendations. According to these findings we adapted the QA reference mappings leading to an average F-measure improvement of 9% (for $t = 0.7$). Note, that we used these adapted QA reference mappings in the previous sections. Some of the recommendations were especially valuable. In particular, we found correct UMLS concepts for 38 so far not annotated questions, e.g.:



Question	Annotated concept
Heartbeat skipping (except for sleeping phases)	Dropped beats – heart (C0425591)
Ulcerating plaque	Carotid Ulcer (C0751634)
Malignant tumor (without curative treatment)	Malignant Neoplasms (C0006826)

The manual annotation of medical forms is difficult for curators. UMLS Metathesaurus is very huge, and even for medical experts it is hard to find a complete set of annotations. Sometimes it is difficult to decide for the correct concept, since UMLS contains similar concepts that might be suitable for the same medical concept in a question of a medical form [136]. Applying our automatic annotation workflow led to new correct annotations and could even indicate some false annotations. Our results point out the importance of semi-automatic annotation approaches. Combining manual and automatic annotation techniques 1) reduces the manual annotation effort and 2) leads to more complete and correct overall results. Semi-automatic annotation is especially relevant, since many medical forms are sparsely or not annotated. For instance, in MDM most items are only pre-annotated and need to be curated again. Part of the forms could not be annotated so far, and MDM is continuously extended by new non-annotated forms. Medical forms in MDM and can be semantically enriched by applying our annotation workflow in combination with expert validation.

3.8 Related Work

Our work on automatic annotation of medical forms is related to the areas of information retrieval [85] and ontology matching [108, 40]. Both research fields have been studied intensively and provide useful methods to process free-text and match identified concepts to standardized vocabularies. GOMMA [63] already allows for efficient and effective matching of especially large life science ontologies and can be a basis to align items with concepts of large ontologies. However, GOMMA does not provide methods to match free-text like form items.

In the medical domain, manual and automatic annotation methods have been studied to semantically enrich different kinds of documents. For instance, in [52] the authors clustered similar clinical trials by performing nearest neighbor search based on similarly annotated eligibility criteria. In [82] the application of a dictionary-based pre-annotation method could improve the speed of manual annotation for clinical trial announcements. The work in [115] focuses on the manual annotation process by presenting a semantic annotation schema and guidelines for clinical documents like radiology reports. In own previous work we already used manual annotations to compare and cluster different medical forms from the MDM platform [36]. We further identified most frequent eligibility criteria in clinical trial forms and performed a manual annotation for these top terms [136].

Previous research showed the usefulness of semantic annotations for different kinds of clinical documents. However, the problem remains that annotations, in particular, for medical forms are only sparsely available. So far, there is no automatic annotation tool to support the semantic annotation of large medical form sets as provided by MDM. In contrast to previous work on document annotation in the medical domain, we here focus on the development of automatic annotation methods for medical forms. In particular, we use a novel blocking technique to reduce the complexity of UMLS as well as a combined match approach to cope with shorter as well as free-text questions. A novel group-based filtering allows to select the most likely set of question annotations to be presented for further manual validation.

3.9 Conclusion

In this chapter, we proposed a basic workflow to (semi-)automatically annotate items in medical forms with concepts of UMLS. The automatic annotation is challenging since form questions are often formulated in long natural language sentences and can cover several medical concepts. The huge size of UMLS further complicates the annotation generation. We used a combined match strategy and presented a novel semantic blocking as well as a group-based filtering of annotations. We applied our methods to annotate real-world medical forms from the MDM portal and performed a manual validation of the generated annotations. Our methods showed to be effective and we could generate valuable recommendations. Medical experts can benefit from automatic form annotation since it reduces the manual effort and can prevent from missing or incorrect annotations.

In the following chapter, we propose a reuse repository to facilitate the annotation of existing medical forms based on well-annotated items. Moreover, we extend our annotation workflow with a context based similarity measure that consider the coherence of concepts.

4
Reuse of annotations

Preamble

This chapter is based on [26]. We propose an approach to generate a reuse repository using well-annotated items from medical forms. The repository is used in the introduced annotation workflow described in Chapter 3 to improve the annotation mapping quality. Moreover, we propose a context-based similarity measure based on graph-based measures.

4.1 Motivation

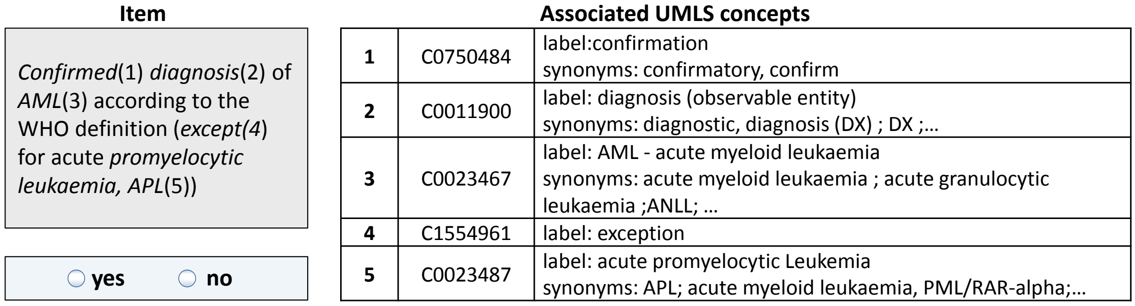
The annotation of data with concepts of standardized vocabularies and ontologies has gained increasing significance due to the huge number and size of available datasets as well as the need to deal with the resulting data heterogeneity.

Annotations of medical documents such as EHRs can also support advanced analyzes, e.g. significant co-occurrences between the use of certain drugs and negative side effects in terms of occurring diseases [75]. Still many medical documents are not annotated at all, impeding data analysis and data integration. Every study requires a set of so-called case report forms (CRFs), e.g. to ask for the medical history of probands. For every new clinical trial, CRFs are usually built from scratch, although previous forms might already cover similar topics. CRF annotations are helpful to search for existing form collections, e.g., in the MDM repository of medical data models [11].

To improve the value of medical documents for analysis, reuse and data integration it is thus crucial to annotate them with concepts of ontologies. Since the number, size and complexity of medical documents and ontologies can be very large, a manual annotation process is time-consuming or even infeasible. Hence, automatic annotation methods become necessary to support human annotators with recommendations for manual verification.

Figure 4.1 shows an exemplary annotation for one item in a medical form (CRF) on eligibility criteria for a clinical trial on acute myeloid leukaemia (AML). Such an item comprises a question as well as a response field or a list of answer options. The shown question has been manually annotated based on a reference mapping with five concepts of the Unified Medical Language System (UMLS) [9], a comprehensive knowledge base integrating many biomedical ontologies. The associated UMLS concepts relate to different terms of the item text (italicized) as indicated by the numbers (1) to (5).

The automatic annotation of medical documents is challenging for several reasons. In particular, it is difficult to correctly identify relevant terms and medical concepts within natural language sentences such as the items (questions) occurring in medical forms. This is because concepts typically have several synonyms that may occur in sentences in different variations. Furthermore, concepts are often described by labels or synonyms consisting of several words, e.g., *AML-Acute myeloid leukaemia* (*C0023467*), that can match many irrelevant terms in the items to be annotated. We might further need to identify complex many-to-many mappings between items and ontology concepts without knowing a priori how many medical concepts should be associated per item. Moreover, UMLS is very large (2.8 mio. concepts) making it difficult to identify the best fitting concepts for annotation.



Item

Confirmed(1) *diagnosis*(2) of *AML*(3) according to the WHO definition (*except(4)* for acute *promyelocytic leukaemia, APL*(5))

○ **yes** ○ **no**

Associated UMLS concepts

1	C0750484	label:confirmation synonyms: confirmatory, confirm
2	C0011900	label: diagnosis (observable entity) synonyms: diagnostic, diagnosis (DX) ; DX ;…
3	C0023467	label: AML - acute myeloid leukaemia synonyms: acute myeloid leukaemia ; acute granulocytic leukaemia ;ANLL; …
4	C1554961	label: exception
5	C0023487	label: acute promyelocytic Leukemia synonyms: APL; acute myeloid leukaemia, PML/RAR-alpha;…

Figure 4.1: Example medical form item and associated annotations.

The results of the previous Chapter 3 revealed the mentioned challenges and showed the difficulty of automatically achieving high quality annotations especially for long natural language sentences. Moreover, we observed frequent errors due to the high number of available concept synonyms and misleading terms in synonyms. In this chapter, we aim at improving the quality of annotations and reducing the manual annotation effort by reusing already determined and manually verified annotations. This assumes that there are similar questions in different medical forms of a domain of interest so that previous annotations can be reapplied. For this purpose, we propose and evaluate a new reuse-based annotation approach for annotating medical forms and documents.

Specifically, we make the following contributions:

- To enable annotation reuse, we propose to cluster all previously annotated items that are annotated with the same medical concept. For such annotation clusters, we identify representative features that are more compact than the large set of terms in concept labels and synonyms. We use these clusters and their features to find likely annotations for new items that are similar to already annotated ones.

- We propose a new context-based strategy to select the most promising annotations from a set of previously determined candidates. The strategy considers both the semantic relatedness of the annotating concepts as well as their co-occurrence in previously annotated items.

- We evaluate the proposed approaches based on reference mappings between a set of medical forms and UMLS and compare them with a baseline annotation approach as well as with using the MetaMap tool [4] to identify UMLS concepts within medical documents.

The remainder of this chapter is organized as follows. We first discuss related work where we focus on MetaMap and cTakes in more detail that are also used in the evaluation of this chapter as well as in the next Chapter 5. We give a short remainder of our base workflow for determining annotations (Section 4.3). We then propose our reuse-based annotation approach and the context-based selection strategy (Section 4.4). Section 4.5 presents evaluation results for the approaches. Finally, we discuss related work in Section 4.2 and conclude in Section 4.6.

4.2 Related Work

The automatic annotation of medical forms and documents with concepts of standardized vocabularies is related to the well-studied fields of ontology matching [108, 40] and entity linking [121]. Both research domains provide useful generic methods to identify concepts or names in full-text documents and match them to concepts or entities of a knowledge base or standardized vocabulary. These techniques can also be applied to the medical domain. In fact, our base workflow proposed in Chapter 3 uses linguistic ontology matching techniques to map terms of medical forms to the concepts and their synonyms of the UMLS ontology. Entity linking approaches focus on the identification of named entities in text documents and their linking to corresponding entities of a knowledge base for enrichment. Many approaches (e.g. [29, 88, 144]) use a dictionary-based strategy to identify entity occurrences by searching the whole knowledge base.

Moreover, there are many approaches to select the correct entities from a set of candidates (e.g. [29, 71, 50]). For instance, in [50] co-occurrences of entities in Wikipedia articles are transformed into a graph model to consider the global interdependence between different candidate entities in a document. As we discussed in Chapter 2 these approaches achieve qualitative results for general domains but lack to generalize for specific domains such as the life sciences. Therefore, there is also some research focusing on the manual or automatic annotation of certain kinds of medical documents such as MetaMap and cTakes that are commonly used in the biomedical domain. In this thesis, we selected the common used tools *MetaMap* [4] and *cTAKES* [118]. Lin et al. [81] evaluate comprehensively both tools regarding the annotation quality for medical forms. MetaMap is

Table 4.1: Components and functions of MetaMap, cTAKES and AnnoMap. POS: Part of Speech, LCS: Longest Common Substring

tool	ontology prepocessing	form preprocessing	candidate generation	post-processing
MetaMap	dictionary construction (UMLS, SPECIALIST lexicon)	sentence detector, tokenizer, POS tagger/filter, shallow parser, variant generation (static/dynamic), abbreviation identifier	dictionary lookup (first word)	word sense disambiguation, score-based filtering
cTAKES	dictionary construction (UMLS)	sentence detector, tokenizer, POS tagger/filter, shallow parser, variant generation (dynamic)	dictionary lookup (rare word)	-

a well established tool and has been applied in many different types of tasks such as text mining, classification and question answering [4]. We chose cTAKES as it performed best in the mentioned evaluation study [135]. In the following, we discuss the two dictionary-based tools in more detail. Table 4.1 gives an overview of the main features for each tool regarding the different steps.

4.2.1 MetaMap

MetaMap was originally developed to improve the retrieval of bibliographic documents such as MEDLINE citations [4]. It is designed to map biomedical mentions to concepts in UMLS Metathesaurus. MetaMap is based on a dictionary-lookup by using several sources such as UMLS itself as well as SPECIALIST lexicon. The SPECIALIST lexicon contains syntactic, morphological, and spelling variations of commonly occurring English words and biomedical terms of UMLS [87]. The input text is first split into sentences and further parsed into phrases. These phrases are the basic units for the variant generation and candidate retrieval. MetaMap provides several configurations for the lookup of annotation candidates per phrase such as *gap* allowance, *ignore* word order, and *dynamic* as well as *static* variant generation. For each annotation candidate MetaMap computes a complex score function considering linguistic metrics [4] for each phrase of a sentence. The final result is determined by the combination of candidates maximizing the aggregated score. MetaMap also provides an optional postprocessing step, word sense disambiguation (WSD), for cases when the final result has several Metathesaurus concepts with similar scores. WSD selects the concept that is semantically most consistent with the surrounding text [58].

4.2.2 cTAKES

cTAKES[1] is built on the Apache UIMA framework[2] providing a standardized architecture for processing unstructured data. To annotate medical documents, cTAKES provides several components for specifying preprocessing and lookup strategies. The components are used to define customized annotation pipelines where each component uses the intermediate output of the previous component as input. In addition to general components used in a default pipeline, cTAKES offers domain-specific components such as for the classification of smoking status [127], the extraction of drug side effects [126], and coreference resolution [145].

In the following, we describe the default pipeline with its components. During (offline) preprocessing, an ontology dictionary is built where each property of a concept becomes an entry in the dictionary. The rarest word of an entry is used to index it for fast lookup. The rareness of a word is based on the global occurrence frequency in the ontology. For the (online) preprocessing of the input documents, cTAKES uses the following components: sentence boundary detector, customized part of speech (POS) tagger and a lexical variant generator. The model of the POS tagger is trained for medical entities based on clinical data since general POS taggers do not cover domain-specific characteristics such as abbre-

[1] Clinical Text Analysis and Knowledge Extraction System http://ctakes.apache.org

[2] Unstructured Information Management Architecture[118] https://uima.apache.org

viations. In general, medical entity mentions within documents can be different according to the name and synonyms of concepts. Therefore, cTAKES applies a lexical variant generator (LVG) to transform differently inflected forms, conjugations or alphabetic cases to a canonical form for improved comparability. While cTAKES permits the addition of customized postprocessing steps to the pipeline such strategies are not part of the cTAKES core project.

4.2.3 Distinction to our approach

In contrast to previous research, we propose a novel reuse-based annotation approach for medical documents. Our method is especially valuable to annotate documents from different biomedical domains with ontology concepts, i.e. it is not restricted to a specific medical subdomain. The proposed use of annotation clusters and their feature sets has not been explored before. Furthermore, we apply a novel context-based selection of annotations considering both, the co-occurrences of verified annotations as well as the semantic relatedness of concepts. Our comparative evaluation showed that the new approaches outperform previous annotation schemes including tools like MetaMap. Furthermore, there is evidence in the literature that MetaMap results are not fine-grained enough [84], contain many spurious annotations [104] and do not cover mappings to longer medical terms [113]. These observations confirm that a correct annotation of medical documents with UMLS concepts is challenging.

Algorithm 1: annotation method $\mathcal{A}$

```
Input: Set of forms D, ontology O= $(C, A, R)$, threshold $\delta$
Output: Annotation mapping $AM_{D,O}$
O ← preprocess (O)
$AM_{D,O} \leftarrow \emptyset$
foreach $d_i \in$ D do
    $d_i \leftarrow$ preprocess ($d_i$)
    $AM'_{d_i,O} \leftarrow$ identifyCandidates ($d_i, C, \delta$)
    $AM_{d_i,O} \leftarrow$ selectAnnotations ($AM'_{d_i,O}$)
    $AM_{D,O} \leftarrow AM_{D,O} \cup AM_{d_i,O}$
return $AM_{D,O}$
```

4.3 Base Workflow

In Chapter 3, we used the basic workflow shown in Algorithm 1 to determine annotation mappings for medical forms. The input of the workflow is a set of forms **D**, an ontology **O**, and a similarity threshold δ. First, we normalize the label and synonyms of ontology concepts by removing stop words, transforming all string values to lower case and removing delimiters. The same preprocessing steps are applied for each form d_i. We identify an intermediate annotation mapping $AM'_{d_i,O}$ by lexicographically comparing each question with the label and synonyms of ontology concepts. For this purpose we apply three string similarity measures, namely trigram, TF/IDF as well as a longest common sequence string similarity approach. We keep an annotation (df, c, sim), if the maximal similarity of the three string similarity approaches exceeds the threshold δ. Finally, we select annotations from the intermediate result by not only choosing the concepts with the highest similarity but also by considering the similarity among the concepts. For this purpose, we group the concepts associated with a question based on their mutual similarity and only choose the concept with the highest similarity per group in order to avoid the redundant selection of highly similar concepts. This group-based selection proved to be quite effective in Chapter 3 albeit it only considers the string-based (linguistic) similarity between questions and concepts, and among concepts.

4.4 Reuse-based Annotation Approach

In this section we outline an extended workflow to determine annotation mappings that reuses previously found annotations for similar questions. The goal is to reduce the complexity of the annotation problem by avoiding to search a very large ontology for finding concepts that describe or match terms of a question to annotate. By reusing verified annotations we also hope to achieve a good annotation quality since the previous annotations may include concepts that are difficult to find by common match techniques based on linguistic similarity. The reuse approach is also motivated by the existence of a high number of related forms in a specific domain, e.g. dealing with a specific disease. It would thus be desirable to reuse the annotation of a subset of these forms to more quickly and effectively annotate the remaining ones. The proposed approach is not limited to the annotation of medical forms but could be generalized for other medical documents such as electronic health records (EHRs) where we would associate medical concepts from an ontology to specific sentences or sections of the document rather than to questions.

We will first outline the new workflow for reuse-based annotation and then provide more details about its main steps, i.e., the generation of so-called annotation clusters (Subsection 4.4.2), determination of candidate annotations (Subsection 4.4.3) and a context-based strategy for selecting the final annotations (Subsection 4.4.4).

4.4.1 Workflow for Reuse-based Annotation

Algorithm 2: Extended annotation method $\mathcal{A}^{reuse}$

```
Input: Set of unknown forms D, ontology O= $(C, A, R)$, set of verified annotation
        mappings $AM_{D,O}^{verified}$, threshold $\delta$
Output: Annotation mapping $AM_{D,O}$
AC ←determineAnnotationCluster ($AM_{D,O}^{verified}$);
AC ← determineFeatureSets (AC, O);
foreach $d_i \in$ D do
    $d_i \leftarrow$ preprocess ($d_i$);
    $AM_{d_i,\mathbf{O}}^{Reuse} \leftarrow$ identifyCandidatesByReuse ($d_i$, AC, $\delta$);
    $d'_i \leftarrow$ findUnannotatedQuestions ($d_i$, $AM_{d_i,O}^{Reuse}$);
    $AM_{d'_i,O}^{reduced} \leftarrow$ identifyCandidates ($d'_i$, O, $\delta$);
    $AM'_{d_i,\mathbf{O}} \leftarrow AM_{d'_i,O}^{reduced} \cup AM_{d_i,O}^{Reuse}$;
    $AM_{d_i,O} \leftarrow$ selectAnnotationsByContext ($AM'_{d_i,O}$);
    $AM_{D,O} \leftarrow AM_{D,O} \cup AM_{d_i,O}$;
return $AM_{D,O}$;
```

The workflow for the reuse-based annotation approach is shown in Algorithm 2. Its input includes a set of verified annotation mappings containing the annotations for reuse. The result is a set of annotation mappings $AM_{D,O}$ for the unannotated input forms **D** w.r.t. ontology **O**. In the first step, we use the verified annotations to determine a set of *annotation clusters* $\mathbf{AC} = \{ac_{c_1}, ac_{c_2}, ..., ac_{c_m}\}$. For each concept c_i used in the verified annotations, we have an annotation cluster ac_{c_i} containing all questions that are associated to this concept. To calculate the similarity between an unannotated question and the questions of an annotation cluster we determine for each cluster a *representative* (feature set) $ac_{c_i}^{fs}$ consisting of relevant term groups in this cluster. These term groups are identified based on common terms between the questions $q \in ac_{c_i}$ and the description (label and synonyms) of the corresponding concept of ac_i.

After these initial steps we determine the annotation mapping for each unannotated input form d_i (lines 3-7 in Algorithm 2). We first preprocess a form as in the base approach of Algorithm 1. Then we determine an annotation mapping $AM_{d_i,O}^{Reuse}$ for the form based on the annotation clusters. Depending on the degree of reusable annotations the determined mapping is likely to be incomplete. We thus identify all questions that are not yet covered by the first mapping. For these questions we apply the base algorithm to match them to the whole ontology and obtain a second annotation mapping (line 7). We then take the union of the two partial mappings to obtain the intermediate mapping $AM'_{d_i,O}$. Finally, we apply a new strategy to select the annotations for the final mapping $AM_{D,O}$. This selection strategy considers the context of concepts, their linguistic similarity as well as their co-occurrences in previous annotations.

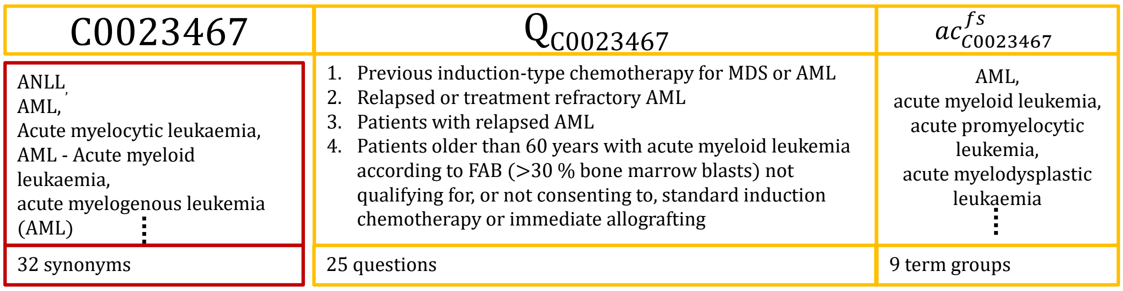
4.4.2 Generation of Annotation Clusters and Representatives

We build annotation clusters from verified annotation mappings by creating a cluster for each applied ontology concept c_k and associating to it all input questions that are assigned to this concept. Formally, an annotation cluster ac_{c_k} is represented as triple:

$$ac_{c_k} := (c_k, Q_{c_k}, ac_{c_k}^{fs}).$$

It includes the concept c_k, the set of questions Q_{c_k} annotated with c_k, as well as a cluster representative or feature set $ac_{c_k}^{fs}$. The purpose of the cluster representative is to provide a compact cluster description that is more suitable for finding further annotations than the free text questions or the label and synonym terms of the ontology concept.

A feature set is formed by terms or groups of terms that frequently co-occur in the questions of the cluster and that are similar to the synonym description of



C0023467	$Q_{C0023467}$	$ac^{fs}_{C0023467}$
ANLL, AML, Acute myelocytic leukaemia, AML - Acute myeloid leukaemia, acute myelogenous leukemia (AML) ⋮	1. Previous induction-type chemotherapy for MDS or AML 2. Relapsed or treatment refractory AML 3. Patients with relapsed AML 4. Patients older than 60 years with acute myeloid leukemia according to FAB (>30 % bone marrow blasts) not qualifying for, or not consenting to, standard induction chemotherapy or immediate allografting	AML, acute myeloid leukemia, acute promyelocytic leukemia, acute myelodysplastic leukaemia ⋮
32 synonyms	25 questions	9 term groups

Figure 4.2: Sample annotation cluster $ac_{C0023467}$ for UMLS concept *C0023467* with its set of associated questions $Q_{C0023467}$ and feature set $ac^{fs}_{C0023467}$.

the corresponding concept. To identify frequently co-occurring terms, we use a frequent itemset mining algorithm where the frequency of term groups has to exceed a given $min\_support$. Moreover, we only keep term groups that maximize the overlap between the terms of a question and the synonyms or the label of a concept, i.e., we do not use term groups that build a subset of another frequently occurring term group. The resulting feature sets build representatives for the annotation clusters that will be used to identify new annotations by matching unannotated forms to cluster representatives.

As an example, Figure 4.2 shows the resulting annotation cluster $ac_{C0023467}$ for UMLS concept *C0023467* about the disease *Acute Myeloid Leukaemia*. In the UMLS ontology, this concept is described by a set of 32 synonyms (Figure 4.2 left). The annotation cluster also contains 25 questions associated to this concept in the verified annotation mappings. Most questions only relate to some of the synonym terms of the concept while other synonyms remain unused. So the abbreviation 'AML' that is a part of some synonyms is often used but the abbreviation 'ANLL' does not occur in the medical forms used to build the annotation clusters. For this example, we generate only 9 relevant term groups, i.e., the representative feature set of the cluster is much more compact than the free text questions and large synonym set.

4.4.3 Identification of Annotation Candidates

To reuse the confirmed annotations for unannotated forms we have to determine the annotation clusters (and thus their concepts) that match best the new questions to be annotated. One difficulty is that we need to find several annotations per question, i.e., we aim at identifying several annotation clusters. Since we may find too many related annotation clusters it is also important to select the most promising ones from the set of candidates.

We first describe how we determine the set of candidate annotation clusters. The example in Figure 4.1 showed that annotating concepts typically refer to some portion, i.e., succeeding terms, of the question text. Our approach to find matching annotation clusters thus uses a sliding window with a specified size *wnd_size* that partitions a given question into smaller portions according to the order of words in the question. Every text portion is compared with the feature set of every existing annotation cluster using a linguistic similarity measure. For this linguistic comparison we apply a soft TF/IDF string similarity function. TF/IDF weights the different terms based on their significance in all considered documents. A soft variant of TF/IDF is more robust than TF/IDF w.r.t. different word forms. An annotation cluster and thus its concept is an annotation candidate for a given question, if the linguistic similarity exceeds a threshold δ for one portion of the question.

In the final selection of annotations, we want to avoid choosing similar annotations referring to the same medical concept. We therefore group the annotation candidates per question that relate to the same tokens and text portions of a question. For selecting the best matching concept per candidate group we apply the context-based selection strategy to be described next.

4.4.4 Context-based Selection of Annotations

The input for the final selection of annotations is a set of grouped candidate concepts for each question in the medical forms $\mathcal{F}$. To determine the final annotations per question, we rank the candidate concepts within each group based on a combination of both linguistic and context-based similarity among the candidate concepts. For this purpose, we calculate an aggregated similarity (*aggSim*) for each question and candidate concept based on weighted linguistic (*lsim*) and context (*csim*) similarity scores:

$$aggSim_{df,Candidates}(c_k) = \omega_{lsim} \cdot lsim(df, c_k) + \omega_{csim} \cdot csim(c_k, Candidates)$$

The linguistic similarity between candidate concepts is determined by the linguistic similarity of their concept descriptions, similarly as in the selection strategy of the base approach (Section 4.3). The calculation of the context-based similarity is more involved and will be described below. For each question in the set of input forms, we select the concepts with the highest *aggSim* value per candidate group to obtain the final set of annotations.

For the context-based similarity between candidate concepts we consider two criteria: first, the degree to which concepts co-occurred in the annotations for the same question within the verified annotation mapping, and second, the degree of semantic (contextual) relatedness of the concepts w.r.t. the ontological structure. The goal is to give a high contextual similarity (and thus a high chance of being selected) to frequently co-occurring concepts and to semantically close concepts. These concepts are more likely to fit the context of a question which is typically about one subject, e.g. different medical aspects such as medications for a specific disease.

For the context-based selection of candidate concepts, we construct a *context graph* $G_{df} = (V_{df}, E_{df})$ for each question df. The vertices V_{df} represent candidate concepts that are interconnected by two kinds of edges in E_{df} to express that concepts have co-occurred in previous annotations or that concepts are semantically related within the ontology. In both cases we assign distance scores to the edges that will be used to calculate the context similarity between concepts. Figure 4.3 a shows the sample input for annotation selection consisting of a question and the set of grouped candidate concepts. In the context graph of the question (Figure 4.3 b), green edges interconnect concepts that have co-occurred before and red edges interconnect semantically related concepts.

To determine the co-occurrence score between concepts c_1 and c_2 we count how often the two concepts have been annotated to the same question and compute the following normalized overlap of their annotation clusters:

$$cooccDist(c_1, c_2) = 1 - \frac{|ac_{c_1} \cap ac_{c_2}|}{|ac_{c_1}|}$$

Concepts that often co-occur thus have a small distance score.

We further assign a semantic distance between concepts in the context graph based on the shortest path between two considered concepts in the ontological structure (see Figure 4.3 c), similarly to common techniques [107]. The ontolog-

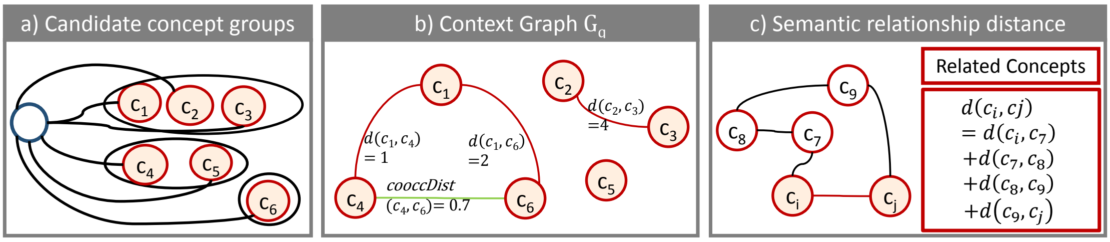


Figure 4.3: Context-based similarity computation. **a)** candidate concept groups for one question; **b)** context graph with different edges for concept co-occurrence (green edges) and semantic relatedness (red edges); **c)** computation of semantic relatedness between concepts with related concepts from UMLS.

ical structure consists of the $is-a$, $part-of$ relationships and further domain specific relationships. We determine the semantic distance between two candidate concepts by summarizing the weighted distances of each relationship within the shortest path. We currently use the same distance 1 for each relationship type. Hence the semantic distance between two concepts corresponds to the path length, e.g., distance 4 for the concept pair in the example of Figure 4.3 c.

Based on the context graph and its distance scores we compute a context-based similarity for each concept by computing the distance to all other concepts in the candidate set of a question. Thereby, we favor concepts that often co-occur and those with a close semantic relatedness for our selection, i.e. selected concepts should have a small distance to other annotated concepts. We use the closeness centrality measure cc that computes the reciprocal of the sum of all distances d between a vertex v and all other vertices w in the graph G:

$$cc(v) = \frac{1}{\sum_{w \in G} d(v,w)}$$

We adopt a modified version of the closeness centrality to compute the context-based similarity score as follows. In our graph concepts can be isolated in case they do not co-occur with any other concepts and have a very different semantic context (e.g., concept c_5 in the context graph of Figure 4.3 b). Such isolated concepts should get a lower similarity score than concepts in a larger subgraph of G_{df}. However, isolated concepts have infinite distances d to all other vertices such that $cc(v)$ would often converge to zero. To compute a normalized context-based similarity score $csim(c_i) \in [0,1]$ for each concept c_i in the set of vertices V_{df} of the context-graph G_{df}, we sum up single reciprocal values of distances and normalize it by the number of concepts in the context-graph:

$$csim(c_i, V_{df}) = \frac{\sum_{c_j \in V_{df} \setminus \{c_i\}} \frac{1}{d(c_i,c_j)}}{|V|-1}$$

Concepts with a small distance to every other concept in the graph have high *csim* values meaning they are highly related to the other candidate concepts due to annotation co-occurrences and relationships from UMLS.

For instance, the context similarity for the concept c_4 is computed by the semantic distance $d(c_4, c_1) = 1$ and the co-occurrence distance $cooccDist(c_4, c_6) = 0.7$. The distances to the other concepts in the context graph are infinite. Therefore, we get the following context-similarity $csim(c_4) = \frac{\frac{1}{1}+\frac{1}{0.7}+\frac{1}{\infty}+\frac{1}{\infty}+\frac{1}{\infty}}{6-1} \approx 0.49$.

4.5 Evaluation

We now evaluate the proposed reuse-based annotation approach for medical forms and compare it with the baseline approach and the MetaMap tool. In the next subsection we introduce the used datasets and workflow configurations. We then evaluate the annotation quality compared to the baseline approach (Subsection 4.5.2) and analyze the effectiveness of the context-based selection strategy (Subsection 4.5.3). Finally, we provide the comparison with MetaMap (Figure 4.5.4).

4.5.1 Evaluation Setting

Our evaluation uses medical forms about eligibility criteria EC and about quality assurance QA w.r.t cardiovascular procedures from the MDM platform [11]. The forms in the first dataset are used to recruit patients in clinical trials. Most questions in this dataset are long natural language sentences since the recruitment of clinical trial participants requires a precise definition of inclusion and exclusion criteria. The sentences contain $\sim$ 8 tokens on average and often mention several medical concepts. The QA forms are used by health service providers in Germany since 2000 to document the quality of their services. The questions of the QA forms are shorter than the eligibility criteria ($\sim$ 3 tokens on average), therefore a question is probably annotated with only one concept. The forms will be annotated with concepts of a reduced version of UMLS [9] covering all UMLS concepts that possess at least one preferred label or synonym ($\sim$1 Mio. concepts with $\sim$ 7 Mio. labels/synonyms). Moreover, we do not consider general concepts ($\sim$ 12000 concepts) that are associated with one of the following semantic types: *Qualitative Concept, Quantitative Concept, Functional Concept, Conceptual Entity*.

To evaluate the quality of automatically generated annotations, we use manually created reference mappings from the MDM portal [11]. These reference mappings might not be perfect ("a silver standard") since the huge size of UMLS makes it hard to manually identify the most suitable concepts for each item. We divide the set of input forms into disjoint reuse and evaluation datasets. For both use cases, EC and QA, we consider two reuse datasets of different sizes to study the impact of the amount of reusable annotations. Table 4.2 shows the number of forms, items and verified annotations for the reuse and evaluation datasets. To analyze the quality of the resulting annotation mappings, we compute precision, recall and F-measure using the union of all annotated form items in the evaluation dataset.

Table 4.2: Statistics on the reuse and evaluation datasets for EC and QA

dataset	ECRD1	ECRD2	ECeval	QARD1	QARD2	QAEval
#forms	100	200	25	16	32	23
#items	1638	3125	310	453	795	609
#annotations	6911	13027	578	694	1054	668

For our reuse-based annotation workflow, we set a fixed window size *wnd_size* of five tokens for the *Candidate Identification and fixed weights ω_lsim/ω_csim to 0.5 for the* Context-based Selection*. In our experiments, we observe that these parameters only slightly affected the results for the considered datasets.* We evaluate different thresholds $\delta = \{0.5, 0.6, 0.7\}$ to present the recall and precision trends. For the selection strategy we consider both the previously proposed group-based strategy [27] as well as the new context-based strategy. Note that we can use the group-based strategy not only for the base workflow but also in the reuse-based approach by setting the weight ω_{csim} for the context similarity to 0.

4.5.2 Reuse-based Annotation

Figure 4.4 shows evaluation results w.r.t. the mapping quality (precision, recall, F-measure) for the baseline approach and the different configurations of the reuse-based approach for the two datasets. For the baseline approach we only show the results for the best threshold of $\delta = 0.7$ for both datasets. The reuse-based approaches uses the context-based selection strategy. We observe that the reuse-based approach can significantly improve the annotation quality and that the improvement grows with the amount of annotations that we can reuse. Compared to the baseline approach, the reuse of existing annotations increases the F-measure from 39.1% to 50.7% for the EC dataset and from 57.5% to 59% for the QA dataset for the best threshold setting of $\delta = 0.6$. Using more existing annotations (EC_{RD2} and QA_{RD2}) improves the mapping quality - and especially recall - compared to the smaller reuse datasets (EC_{RD1} and QA_{RD1}) since annotation clusters and their feature sets become more accurate and are thus more valuable to match to unannotated questions. The reuse-based approach is especially effective for the EC dataset where we could apply more annotations (Table 4.2) to build the annotation clusters compared to the QA dataset. The results confirm that matching questions to the feature sets of annotation clusters (*reuse-based*) helps to identify more correct annotations than trying to find the best matches in the UMLS (*baseline*). At the same time, the reuse-based approaches with the context-based selection strategy usually improve precision compared to the baseline approach.

An added benefit is that the execution time of the reuse-based approaches is lower than for the baseline approaches since matching questions with the compact annotation clusters is much faster than matching with the large UMLS ontology. Overall, runtimes could be reduced by half for our experiments compared

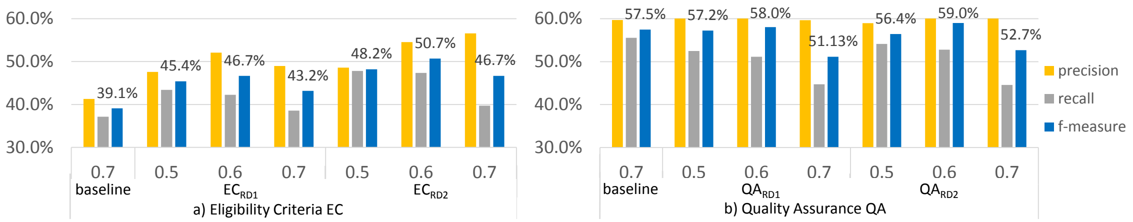


Figure 4.4: Results on the quality of annotation results for the baseline and reuse-based annotation using the EC dataset and the QA dataset with both configurations.

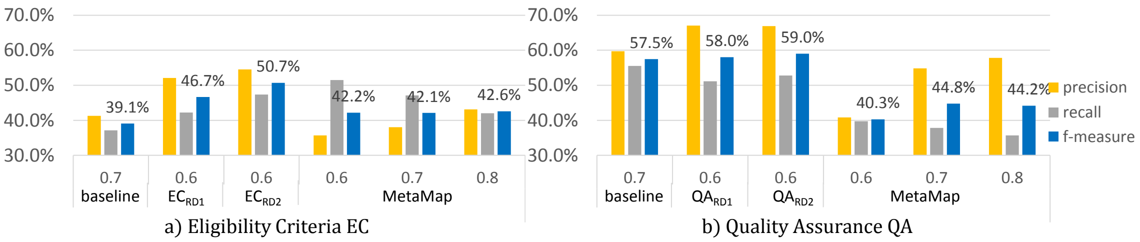
to the baseline. Moreover, the execution time depends on the number of reused forms and the coverage of reused annotation clusters.

4.5.3 Context-based Selection

To analyze the effectiveness of the proposed context-based selection strategy (CS), we now compare its use with the group-based selection strategy (GS) that was used in the baseline approach but can also be applied for the reuse-based approaches. Table 4.3 shows the resulting mapping quality for the two selection strategies for the different EC and QA reuse configurations and threshold 0.6 that led to the best mapping quality for the reuse-based approach. The results show that the context-based selection strategy improves F-measure in all cases (up to 2.2%) compared to the simpler group-based approach. While recall is generally reduced this is more than outweighed by an increase in precision by up to almost ~7% (EC_{RD2}). This indicates that considering the context eliminates many false candidates.

Table 4.3: Results on the quality of annotation results for the group-based (GS) and context-based (CS) selection strategies for both datasets

$dataset_{configuration}$	EC_{RD1}		EC_{RD2}		QA_{RD1}		QA_{RD2}	
selection-strategy	gs	cs	gs	cs	gs	cs	gs	cs
precision	45.9%	**52.1%**	47.9%	**54.5%**	61.9%	**67.0%**	60.4%	**66.9%**
recall	**43.6%**	42.2%	**49.2%**	47.3%	51.0%	**51.2%**	**54.6%**	52.8%
f-measure	44.7%	**46.7%**	48.5%	**50.7%**	55.9%	**58.0%**	57.4%	**59.0%**



a) Eligibility Criteria EC

b) Quality Assurance QA

Figure 4.5: Comparison of the quality for the resulting annotation mappings from the baseline approach, reuse-based approach and MetaMap.

4.5.4 Comparing reuse-based annotation approach with MetaMap

We finally compare our reuse-based annotation method with the MetaMap tool that is commonly used for annotating medical documents (see Section 4.2). We generate the annotations with a local installation of a MetaMap server and the MetaMapAPI and use the provided word sense disambiguation service and the configuration considering several variants for a concept. We select annotations based on the generated MetaMap score. This score ranges from 0 to 1000 and is computed by applying several ranking functions for each identified term. If MetaMap generates more than one annotation per question, we select the annotations with an aggregated score above a threshold. We normalize the scores by dividing by 1000 for comparing with our approach and evaluate different thresholds $\delta = \{0.6, 0.7, 0.8\}$ for selecting the candidates.

Figure 4.5 shows the results for the two datasets and different configurations. Our reuse-based approach outperforms Meta-Map in terms of mapping quality for each dataset. For the EC dataset, F-Measure is improved by ∼4%(EC_{RD1}) and ∼8.6% (EC_{RD2}) indicating that the the computed annotation clusters allow a more effective identification of annotations than with the original concept definition. In addition, our approach benefits from using the ontological relationships for selecting annotations resulting in a much better precision than using MetaMap (54.5% for EC_{RD2} than compared to 43.1%). While MetaMap achieved a better F-Measure than the baseline approach for the EC dataset it performed poorly for the QA dataset where its best F-Measure of 44.8% was much lower for the baseline approach and reuse-based approaches (57.5 and 59%), mainly because of a very low recall for Metamap.

A positive side of MetaMap is its high performance due to the use of an indexed database for finding annotations. Its runtimes were up to 13 times faster than for the baseline approach and it was also faster than the reuse-based approach. In future work we will study whether the use of MetaMap in combination with the reuse approach, either as an alternative or in addition to the baseline approach, can further improve the annotation quality.

4.6 Conclusion

We proposed and evaluated a new reuse-based approach to semantically annotate medical documents such as CRFs with concepts of an ontology. The approach utilizes already found and verified annotations for similar CRFs. It builds so-called annotation clusters combining all previously annotated questions related to the same medical concept. Clusters are represented by features covering meaningful term groups from the annotated questions and concept description. New questions are matched with these cluster representatives to find candidates for annotating concepts. We further presented a context-based selection strategy to identify the most promising annotations based on the semantic relatedness of concept candidates and well as known co-occurrences from previous annotations. In a real-world evaluation, our methods showed to be effective and we could generate valuable recommendations to reduce the manual annotation effort. Moreover, reusing annotation clusters is more efficient than searching a large knowledge base such as UMLS for suitable annotation candidates.

In the following chapter, we propose an approach that reuse annotation mappings generated from different annotation tools to determine an annotating mapping with machine learning techniques. For example, the MetaMap tool alone was inferior to the reuse-based scheme but it could be used in a combined scheme to find further annotation candidates.

5
Machine-Learning based Tool-Combination

Preamble

This chapter is based on [28]. We extend the work [81] that combines annotation results from different tools by applying machine learning. Therefore, we reuse generated annotation mappings and utilize the computed confidence scores of each tool.

5.1 Motivation

The annotation of entities with concepts from standardized terminologies and ontologies is of high importance in the life sciences to enhance semantic interoperability and data analysis. For instance, exchanging and analyzing the results from different clinical trials can lead to new insights for diagnosis or treatment of diseases. In the healthcare sector there is an increasing number of documents such as electronic health records (EHRs), case report forms (CRFs) and scientific publications, for which a semantic annotation is helpful to achieve an improved retrieval of relevant observations and findings [1, 2].

Unfortunately, most medical documents are not yet annotated, e.g., as reported in [37] for CRFs, despite the existence of several tools to semi-automatically determine annotations. This is because annotating medical documents is highly challenging since documents may contain mentions of numerous medical entities that are described in typically large ontologies such as the Unified Medical Language System (UMLS) Metathesaurus. The mentions may also be ambiguous and incomplete and thus difficult to find within the ontologies. The tools thus typically can find only a fraction of correct annotations and may also propose wrong annotations. Furthermore, the tools typically come with many configuration parameters making it difficult to use them in the best way.

The huge number of documents, the use of natural language within the documents as well as the large complexity of biomedical ontologies such as UMLS make it challenging to find correct annotations for both automatic approaches as well as human experts. The most promising approach is thus to first apply a tool to automatically determine annotation candidates. A human expert can then select the final annotations from these candidates.

Given the limitations of individual tools it is promising to apply several tools and to combine their results to improve overall annotation quality. Lin et al. [81] investigated simple approaches to combine the results of three annotation tools (MetaMap [4], cTAKES [118], AnnoMap [27]) based on set operations such as union, intersection and majority consensus. In this chapter, we propose and evaluate a machine learning (ML) approach for combining several annotation results.

Specifically, we make the following contributions:

- We propose a ML approach for combining the results of different annotation tools in order to improve overall annotation quality. It utilizes training data in the form of a so-called annotation vectors summarizing the scores of the considered tools for selected annotation candidates. In contrast to the previously studied majority or intersection schemes the new combination approach can select annotations determined by only a single tool.

- We evaluate the new approach with different parameter and training settings and compare it with the results of single tools and the previously proposed combinations using set operations.

We first discuss related work on finding annotations and combining different annotation results. In Section 5.3, we propose the ML-based method. We then describe the evaluation methodology and analyze the results in Section 5.4. Finally, we conclude.

5.2 Related work

Many annotation tools utilize a dictionary to store the concepts of the ontologies of interest (e.g., UMLS) to speedup the search for the most similar concepts for certain words of a document to annotate. Such dictionary-based tools include MetaMap, NCBO Annotator [30], IndexFinder [146], ConceptMapper [131], NOBLE Coder [135] cTAKES[118] and AnnoMap that combines several string similarities and applies a post-processing to select the most promising annotations. There have also been annotation approaches using machine le0arning [16]. They can achieve good results but incur a substantial effort to provide suitable training data.

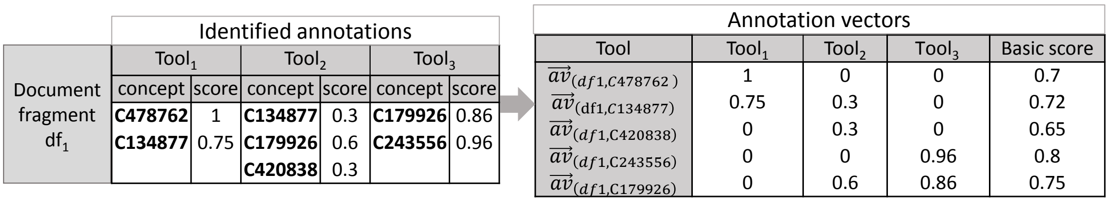
Lin et al. [81] combined annotation results for CRFs determined by the tools MetaMap, cTAKES and AnnoMap using the set-based approaches *union*, *intersection* and *majority*. The *union* approach includes the annotations from any tool to improve recall while *intersection* only preserves annotations found by all tools for improved precision. The *majority* approach includes the annotations found by a majority of tools, e.g., by at least two of three tools. Overall the set-based approach could significantly improve annotation quality, in particular for *intersection* and *majority*.

Though ML approaches have been used for annotating entities, so far they have rarely been applied for combining annotation results as we propose in this chapter. Campos et al. [15] utilized Conditional Random Fields model to recognize named entities of gene/protein terms using the results from three dictionary-based systems and one machine learning approach [14]. The learned combination could outperform combinations based on *union* or *intersection*. Our ML-based combination approach is inspired by methods proposed in record-linkage domain where the goal is to identify record pairs representing the same real-world entity [68]. Instead of a manually configured combination of different similarity values for different record attributes the ML approaches learn a classification model (e.g., using decision tree or SVM learning) based on a training set of matches and non-matches. The learned models automatically combine the individual similarities to derive at a match or non-match decision for every pair of records.

5.3 Machine Learning-based Combination Approach

The task of *annotation* has as input a set of documents $\mathbf{D} = \{d_1, d_2, ..., d_n\}$ to annotate, e.g., EHRs, CRFs or publications, as well as the ontology O from which the concepts for annotation are to be found. The goal is to determine for each document fragment df (e.g., sentences) the set of its most precisely describing ontology concepts. The annotation result includes all associations between a document fragment df_j and its annotating concepts from $\mathbf{O}$. The problem we address is the *combination of multiple annotation results* for documents $\mathbf{D}$ and ontology $\mathbf{O}$ that are determined by different tools. The tool-specific annotation results are obtained with a specific parameter configuration selected from a typically large number of possible parameter settings. The goal is to utilize complementary knowledge represented in the different input results to improve the overall annotation result, i.e., to find more correct annotations (better recall) and to reduce the number of wrongly proposed annotations (better precision).

The main idea of the proposed ML-based method is to train a classification model that determines whether an annotation candidate (df_j, c) between a document fragment df_j and a possibly annotating concept c is correct or not. The classification model is learned based on a set of positive and negative annotation examples for each tool (configuration). For each training example (df_j, c) we maintain a so-called annotation vector $\overrightarrow{av}$ with $n + 1$ elements, namely a quality score for each of the n annotation tools plus a so-called *basic score*. The basic score is a similarity



Identified annotations

	Tool$_1$		Tool$_2$		Tool$_3$	
Document fragment df$_1$	concept	score	concept	score	concept	score
	C478762	1	**C134877**	0.3	**C179926**	0.86
	C134877	0.75	**C179926**	0.6	**C243556**	0.96
			C420838	0.3		

Annotation vectors

Tool	Tool$_1$	Tool$_2$	Tool$_3$	Basic score
$\overrightarrow{av}_{(df1,C478762\ )}$	1	0	0	0.7
$\overrightarrow{av}_{(\mathrm{df1},C134877)}$	0.75	0.3	0	0.72
$\overrightarrow{av}_{(df1,\mathrm{C420838})}$	0	0.3	0	0.65
$\overrightarrow{av}_{(df1,\mathrm{C243556})}$	0	0	0.96	0.8
$\overrightarrow{av}_{(df1,\mathrm{C179926})}$	0	0.6	0.86	0.75

Figure 5.1: Sample annotations and corresponding annotation vectors

between df_j and c that is independently computed from the annotation tools, e.g., based on a common string similarity function such as soft-TF/IDF or q-gram similarity. The use of the basic similarity is motivated by the observation that many concepts may be determined by only one or few tools leading to sparsely filled annotation vectors and thus little input for training the classification model. The learned classification model receives as input annotation vectors of candidate annotations and determines a decision whether the annotation is considered correct or not.

Figure 5.1 shows sample annotation vectors for three tools and the annotation of document fragment df_1. The table on the left shows the annotations found by the tools together with their scores (normalized to a value between 0 and 1). In total, the tools identify five different concepts resulting into the five annotation vectors shown on the right of Figure 5.1. For example, the annotation of df_1 with concept C478762 has the annotation vector $\overrightarrow{av}_{(df1,\mathrm{C478762})}$ of $(1, 0, 0, 0.7)$ since tool 1 identified this annotation with a score of 1, tools 2 and 3 did not determine this annotation (indicated by score 0), and the basic score is 0.7.

We use three classifiers: decision tree, random forest and support vector machines (SVM), to train classification models. A *decision tree* consists of nodes and each node represents a binary decision function based on a score threshold of a tool, e.g. $score_{\mathrm{MetaMap}} > 0.7$. When an annotation vector $\overrightarrow{av}$ is input into a decision tree, decisions are made from the root node to the leaf node according to the values of $\overrightarrow{av}$. As output, $\overrightarrow{av}$ is classified as a correct or incorrect annotation. Random forest [12] utilizes an ensemble of decision trees and derives the classification decision from the most voted class of the individual decision trees. To determine a random forest classification model, each decision tree is trained by different samples of the training dataset. The goal of an *SVM* is to compute a hyperplane that separates the correct annotation vectors (represents a true annotation) from the incorrect ones. To separate vectors that are not linearly separable, SVM utilizes a kernel function to map the original vectors to a higher dimension so that the vectors can be separated.

A key step for the ML-based combination approach is the provision of suitable training data of a certain size. For this purpose, we determine annotation results with different tools and a specific configuration for a set of training documents. From the results we randomly select a subset of n annotations and generate the corresponding annotation vectors AV_{train} and label them as either correct or incorrect annotations. Providing a sufficient number of positive and negative training examples is of high importance to determine a classification model with enough discriminative power to correctly classify annotation candidates. To control the ratio between these two kinds of annotations we follow the approach of [68] and use a parameter $tpRatio$ (true positive ratio). For instance, $tpRatio = 0.4$ means 40% of all annotations in AV_{train} are correct. In our evaluatuion, we will consider the influence of both the training size n and $tpRatio$.

5.4 Evaluation and Results

We now evaluate our ML-based combination approach and compare it with the simpler set-based combination of annotation results. After the description of the experimental setup we analyze the influence of different training configurations and learners. In Subsection 5.4.3, we compare the results of the ML approach with the single tools and set-based combination. The evaluation focuses on the standard metrics *recall*, *precision* and their harmonic mean *F-measure* as the main indicator for annotation quality.

5.4.1 Experimental Setup

We used the two datasets consisting of forms about *eligibility criteria* (EC) and *quality assurance* (QA). These datasets have also been used in previous chapters and turned out to be very challenging.

For annotation we use five UMLS ontologies of version 2014AB: UMLS Metathesaurus, NCI Thesaurus, MedDRA , OAC-CHV , and SNOMED-CT_US . Since we use different subsets of UMLS in this chapter, the results are not directly comparable with the results from Chapter 3 and Chapter 4. The reduced set of concepts results from the observation of the reuse of annotations so that not the whole ontology is relevant. As in the study [81] we combine annotation results of the tools MetaMap, cTAKES and AnnoMap and apply the same set of configurations. In the annotation vectors, we use the normalized scores of the tools and determine the *basic score* by using soft-TF/IDF. For the classifiers (decision tree, random forest, SVM) we apply Weka as machine learning library. We generate training data of sizes 50, 100 or 200 selected from the *union* of the three tools. A $tpRatio \in \{0.2, 0.3, 0.4, 0.5\}$ is applied for each sample generation. For each ML test configuration (i.e., choice of classifier, sample size, $tpRatio$ and tool configuration) we produce three randomly selected training sets and use each to generate a classifier model so that our results are not biased by just one sample. For each test configuration we measure average precision, average recall and macro F-measure that is based on the average precision and the average recall.

5.4.2 Machine Learning-based Combination of Annotation Tools

For the analysis of our ML-based combination approach we first focus on the impact of parameter $tpRatio$ and the size of the training sets. We then compare the three classifiers decision tree, random forest and SVM. Due to space restrictions we present only a representative subset of the results.

Figure 5.2 shows the annotation quality for dataset EC using random forest learn-

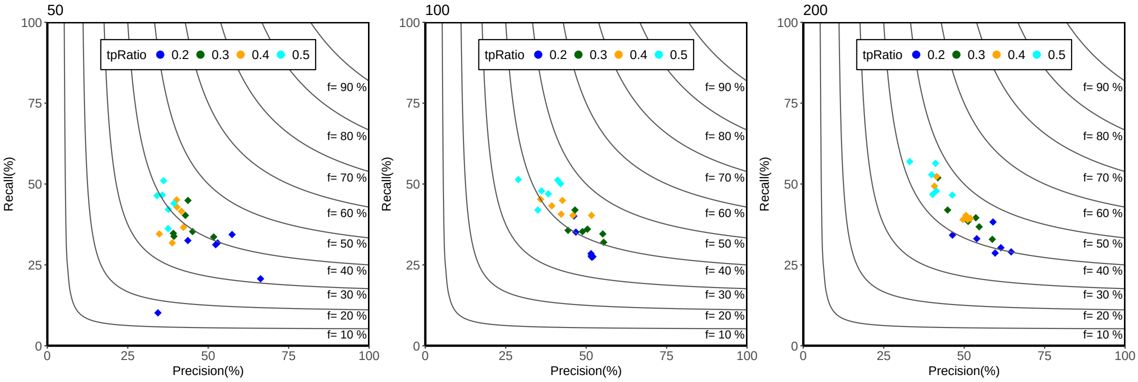


Figure 5.2: Precision/recall results for different $tpRatio$ values and training sizes n (dataset EC, random forest learning)

ing for different *tpRatios* (0.2 to 0.5) and three different training sizes (50, 100 and 200). Each data point represents the classification quality according to a certain *tpRatio* with a certain configuration of the considered tools. We observe that data points with the same *tpRatios* are mostly grouped together indicating that this parameter is more significant than other configuration details. We further observe for all training sizes that models trained with a larger *tpRatios* of 0.5 or 0.4 tend to reach a higher recall (but lower precision) than for smaller *tpRatios* values. Apparently low *tpRatio* values provide too few correct annotations so that the learned models are not sufficiently able to classify correct annotations as correct. By contrast, higher *tpRatio* values can lead to models that classify more incorrect annotations as a correct thereby reducing precision. For random forest, a *tpRatio* of 0.4 is generally a good compromise setting.

Figure 5.2 also shows that larger training sizes tend to improve F-measure since the data points for the right-most figure (training size *n*=200) are mostly above the F-measure line of 50% while this is not the case for the left-most figure (*n*=50). Figure 5.3 reveals the influence of the training size in more detail by showing the macro-average precision, recall and F-measure obtained by random forest using different training sizes. For both datasets, EC and QA, we observe that larger training sizes help to improve both precision and recall and thus F-measure. Hence, average F-measure improved from 40.1% to 42.5% for dataset EC and even from 52.0% to 56.9% for QA when the training size increases from 50 to 200 annotation samples.

Figure 5.4 depicts the macro-average precision, recall and F-measure over differ-

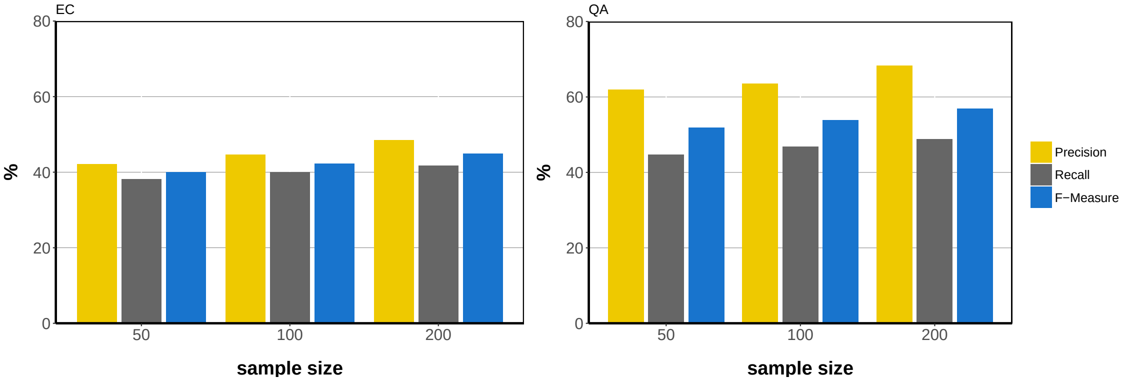


Figure 5.3: Impact of training sizes on annotation quality for datasets EC and QA

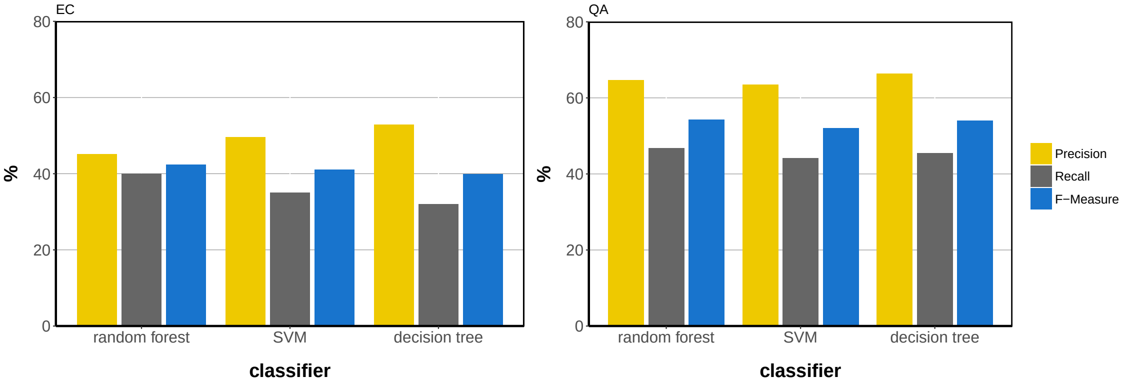


Figure 5.4: Average annotation quality for random forest, SVM and decision tree.

ent *tpRatios*, sample sizes and configurations. For both datasets, random forest obtains the best recall values(EC: 40.0%, QA: 46.8%) while decision tree achieves the best precision (EC: 52.9%, QA: 66.4%). In terms of average F-measure the three learning approaches are relatively close together, although random forest (42.4%) outperforms SVM and decision tree by 1.4% resp. 2.5% for EC. For the QA dataset, random forest (54.3%) outperforms decision tree and SVM by 0.3% resp. 2.2%. Moreover, we experimentally tested our approach with or without using the basic scores in addition to the tool results. We observed that using the basic score improves F-Measure by 1.6%(EC) and 1%(QA), indicating that it is valuable to improve annotation results.

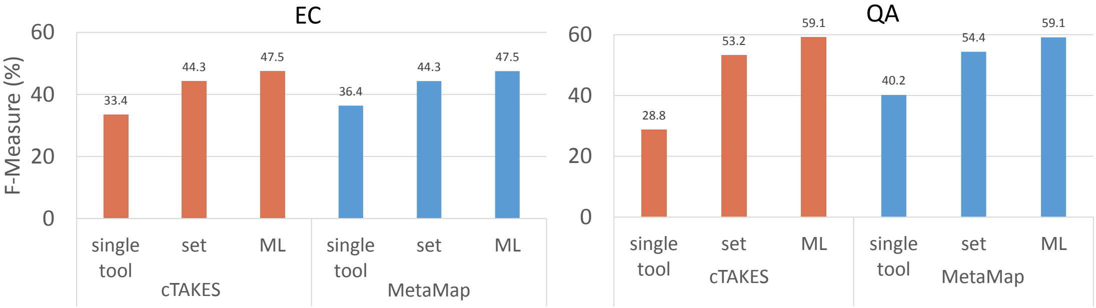


Figure 5.5: Summarizing F-measure results for cTAKES and MetaMap and the set-based and ML-based result combinations for the EC and QA datasets.

5.4.3 Comparison with set-based combination approaches

We finally compare the annotation quality for the ML-based combinations with that of the individual tools cTAKES and MetaMap as well as with the results for the set-based combinations in [81]. Figure 5.5 summarizes the best F-measure results for both datasets. We observe that the F-measure of the individual tools is substantially improved by both the set-based and ML-based combination approaches, especially for cTAKES (by about a factor 3 - 4.5). The ML-based combination outperforms the set-based combinations for both datasets. Consequently, the best results can be improved for EC (from 44.3% to 47.5%) and QA (from 56.1% to 59.1%) by using a sample size of 200. This underlines the effectiveness of the proposed ML-based combination approach.

5.5 Conclusion

In this chapter, we proposed and evaluated a machine learning approach to combine the annotation results of several tools. Our evaluation showed that the ML-based approach can dramatically improve the annotation quality of individual tools and that it also outperforms simpler set-based combination approaches. The evaluation showed that the improvements are already possible for small training sizes (50-200 positive and negative annotation examples) and that random forest performs slightly better than decision tree or SVM learning. In future work, we plan to apply the ML-based combination strategy to annotate further kinds of documents and to use machine learning also in the generation of annotation candidates.

Part III

Application of Entity Resolution methods

6
Temporal group linkage and evolution analysis

Preamble

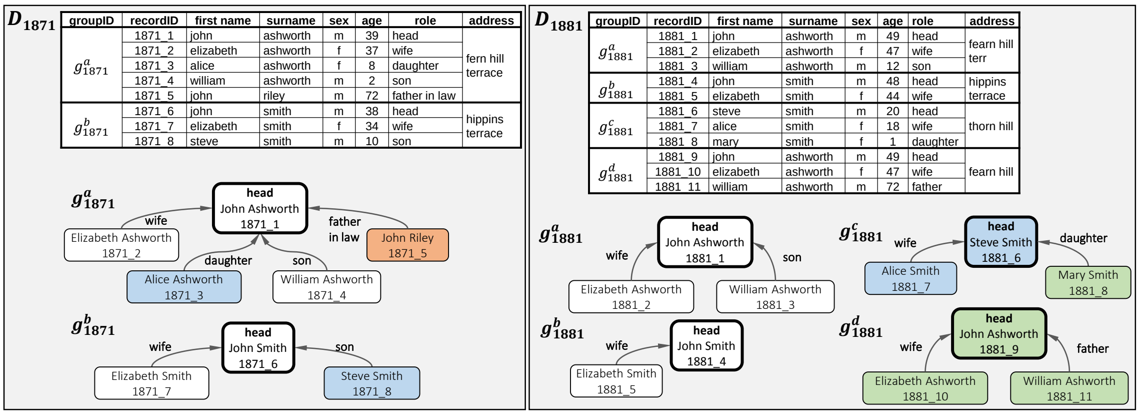
This chapter is based on [25]. Entity resolution is an essential part of data integration to enable qualitative analysis. An application of entity-resolution is the identification of the same person over time in census data. We propose an approach that is more robust than traditional entity resolution approaches regarding temporal changing attributes and that can link households over time. Therefore, we utilize the relationships between persons. The linked persons and households are utilized for temporal analysis. Therefore, we introduce evolution operators representing the temporal aspects of individuals as well as communities.

6.1 Motivation

Census data provides valuable information about individuals and households within cities or regions at a specific point in time [91]. Moreover, the temporal linkage of different census datasets allows analyzing the changes that occur in a population which is of increasing importance for social, demographic, economic and health-related studies [91, 45, 72]. In general, the temporal analysis of changing information about individuals and other entities is seen as a major requirement and challenge for future data analysis [32].

There is a large number of available census datasets for different regions of interest. Normally such census datasets are collected on a regular basis, e.g., every ten years, so that multiple successive versions can be utilized to analyze population- and household-related changes. A key prerequisite for such change studies is the temporal linkage of person records as well as of households, representing a group of individuals living together. There has been a modest amount of previous work on such temporal linkage problems, mainly focusing on temporal record linkage taking into account that linkage-relevant attributes such as surname, address or occupation may change over time [79, 23, 18, 77] (see Section 6.3). These studies mostly ignore the relationships between individuals, e.g., people living together in a household. Moreover, they do not consider the linkage and evolution of groups of related individuals, such as in a household, which is a main focus of this paper.

Figure 6.1 illustrates the problem for two successive historical census datasets from 1871 and 1881. In each dataset, individuals are associated to a single household and have a household-specific relationship or role, such as head of household or daughter (of the head of household). These relationships can be represented in *household graphs* as shown in the lower part of Figure 6.1. To understand the changes between the two considered points in time, one has to find matching individuals and their changes which is challenging, in particular due to the occurrence of frequent names (first names like 'John' and 'Elizabeth' or surnames like 'Ashworth' and 'Smith' in our dataset) and attribute changes. Of course, we also need to identify people who occur only in one of the datasets because of deaths, emigration, births and immigration. Obviously, a person in one census dataset should match to at most one person in another census dataset so that



D_{1871}

groupID	recordID	first name	surname	sex	age	role	address
g^a_{1871}	1871_1	john	ashworth	m	39	head	fern hill terrace
	1871_2	elizabeth	ashworth	f	37	wife	
	1871_3	alice	ashworth	f	8	daughter	
	1871_4	william	ashworth	m	2	son	
	1871_5	john	riley	m	72	father in law	
g^b_{1871}	1871_6	john	smith	m	38	head	hippins terrace
	1871_7	elizabeth	smith	f	34	wife	
	1871_8	steve	smith	m	10	son	

D_{1881}

groupID	recordID	first name	surname	sex	age	role	address
g^a_{1881}	1881_1	john	ashworth	m	49	head	fearn hill terr
	1881_2	elizabeth	ashworth	f	47	wife	
	1881_3	william	ashworth	m	12	son	
g^b_{1881}	1881_4	john	smith	m	48	head	hippins terrace
	1881_5	elizabeth	smith	f	44	wife	
g^c_{1881}	1881_6	steve	smith	m	20	head	thorn hill
	1881_7	alice	smith	f	18	wife	
	1881_8	mary	smith	f	1	daughter	
g^d_{1881}	1881_9	john	ashworth	m	49	head	fearn hill
	1881_10	elizabeth	ashworth	f	47	wife	
	1881_11	william	ashworth	m	72	father	

Figure 6.1: Example census data for two points in time (1871 and 1881). Red / green / blue colored nodes denote individuals who disappear / newly appear / moved to another household.

temporal linkage aims at a 1:1 mapping between person records. Moreover, we want to identify household-related changes, e.g., to what degree the individuals in a household have stayed together or moved to other households. In this case, we have to identify a many-to-many mapping between households.

In our example in Figure 6.1, the daughter of the head of household in g^a_{1871} (*Alice*) married *Steve* from household g^b_{1871} and they both moved into the new household g^c_{1881} as shown in the 1881 census data (see blue nodes in household graphs). *John Riley* died within the considered time period (red node for the first census), while the child *Mary Smith* was born (green node for the second census). Furthermore, a new family (household g^d_{1881}) moved into the region. Note that the groups g^a_{1881} and g^d_{1881} have highly similar attribute values, but only g^a_{1871} should be linked to g^a_{1881}. To overcome such ambiguities of person-related attributes, our linkage approach will utilize stable attributes (such as birth year) as well as stable relationships between records, such as family relations or age differences.

In this paper, we propose and evaluate a novel approach for temporal group and record linkage for historical census data that considers the relationships between individuals. Moreover, we use the linked information for an initial change analysis for individuals and households. Specifically, we make the following contributions:

- We propose a new graph-based approach to linking households and person records between successive versions of census data. The approach works in several steps and utilizes an approximate record matching approach to identify pairs of related households. The linkage of households is based on their graph representation, and identifies common subgraphs referring to individuals with stable attributes and relationships. The final record links are derived from the linked subgraphs. The approach is iterative and determines group and record links in multiple rounds with decreasing restrictiveness. In this way we start with finding the best matches and apply less restrictive similarity criteria only for the more difficult to match records and groups.

- We utilize the determined record and group links for an initial change analysis based on different evolution patterns, including the splitting and merging of households.

- We apply and evaluate the proposed approaches for six historical UK census datasets. The evaluation shows that the proposed linkage approaches are highly effective and that they allow insightful observations regarding the changes over time.

In the next section, we formalize our problem of temporal record and group linkage. The linkage approach is described in Section 3, while Section 4 discusses the use of evolution patterns for change analysis. In Section 6.6, we evaluate our temporal linkage approach and analyze the evolution of households for the considered census datasets. We then discuss related work and conclude.

6.2 Problem Definition

Our approaches to temporal linkage and evolution analysis work on a set of census datasets $\mathbb{D}$ referring to different points in time. Each dataset D_i of time t_i consists of a set of person records R_i and a set of groups G_i representing households. The records in R_i are homogeneously structured and have attributes such as *first name*, *surname*, *age*, *occupation*, and so on. A group $g_i \in G_i$ consists of associated person records (household members) of R_i as well as relationships between them. Each record is part of one group (household) only, i.e., groups are not overlapping.

Groups are represented as (household) *graphs* g_i=(V_i,E_i) where the vertices of V_i correspond to the group members and the edges of E_i represent their relationships. Relationships (edges) have attributes or properties, in particular a relationship type or role, e.g., *daughter*. Such relationships can be part of the input data (as in Figure 6.1) or can be derived later, e.g., the age difference between two persons. For our example, we may record in the graph for group g^a_{1871} not only the role *daughter* between *Alice* and her father *John* but also the age difference 31 (39-8). Our algorithm not only determines additional properties such as age differences but also additional relationships among group members, e.g., that *Alice* and *William* are siblings with an age difference of 6.

Given these datasets and graphs, we want to determine for each pair $D_i = (R_i, G_i)$ and $D_{i+1} = (R_{i+1}, G_{i+1})$ of successive census datasets a so-called record mapping $\mathcal{M}_R^{i,i+1}$ and a group mapping $\mathcal{M}_G^{i,i+1}$. The *record mapping* $\mathcal{M}_R^{i,i+1}$ includes all pairs of records referring to the same real-world person (person links). The mapping is of cardinality 1:1 since each person in R_i can match with at most one person in R_{i+1} and vice versa:

$$\begin{aligned}\mathcal{M}_R^{i,i+1} :&= \{(r_i, r_{i+1}) | (r_i, r_{i+1}) \in R_i \times R_{i+1} \wedge \\ &\exists (r_i, r'_{i+1}) \in \mathcal{M}_R \rightarrow r'_{i+1} = r_{i+1} \wedge \\ &\exists (r'_i, r_{i+1}) \in \mathcal{M}_R \rightarrow r'_i = r_i\}\end{aligned} \tag{6.1}$$

A *group mapping* $\mathcal{M}_G^{i,i+1}$ consists of group pairs where a group g_i of G_i corresponds completely or partially to a group g_{i+1} of G_{i+1} according to the common records:

$$\mathcal{M}_G^{i,i+1} := \{(g_i, g_{i+1}) | (g_i, g_{i+1}) \in G_i \times G_{i+1}\} \tag{6.2}$$

Group mappings can be of cardinailty many-to-many (N:M) since persons of a household can match persons of several households in a different census.

For our running example of Figure 6.1, the record mapping includes seven person links between the white and blue colored graph vertices, e.g. link (1871_1, 1888_1) for *John Ashworth* and (1871_3, 1888_7) for the link between *Alice Ashworth* and *Alice Smith*. The two groups in the first census dataset are split among two groups each in the second dataset, so that there are four group links including

(g^a_{1871}, g^a_{1881}). In our evolution analysis, we will also consider person records and groups that are not reflected in these mappings, e.g. relating to newly occurring or disappeared persons and households.

6.3 Related Work

Record linkage or entity resolution has been intensively studied in the past (see [38, 67, 21] for overviews). While the majority of approaches focus on evaluating the similarity of record attributes only, collective or context-based approaches additionally consider the similarity of relationships between entities for improved linkage decisions (e.g. [8, 73, 111, 60, 45, 132]). This idea has also been utilized in our approach but in a tailored way for use within groups such as households. Our approach is especially powerful as it considers different kinds of semantic relationships as well as the similarity of relationship attributes. Previous collective approaches have also not addressed temporal record linkage in contrast to our scheme.

Relatively few studies have investigated temporal record linkage (e.g., [79, 77, 18]) to link records within dynamically changing data. Existing approaches explicitly consider changing attribute values when matching individual records over time, e.g., by computing value transition probabilities [77]. Temporal clustering approaches as proposed in [19] group temporal records that belong to the same entity to reflect the entity history. Temporal record linkage approaches typically focus on matching individual person records while we also match groups of individuals and identify a record as well as group mapping to interconnect temporal records from census data.

Most closely related to our work is the group-based approach of [45] for matching households in historical census datasets. The main different features compared to the previous scheme are the iterative group linkage and subgraph matching based on different semantic relationships. Richards and colleagues investigate in [114] the use of learning-based methods to optimize the use of attribute similarities for temporal record linkage (not group linkage) for census datasets. The observations of this study are complementary to ours and could be used for choosing alternate similarity functions for record matching.

Our work is further related to research on time and evolution-based analysis that is gaining increasing interest. For instance, there are studies analyzing historical web contents to find interesting patterns and trends [141], analyzing person histories on Twitter [78], or collecting and analyzing temporal knowledge from Wikipedia [140]. Our definition of change patterns is further related to previous work in the domain of ontology evolution [129, 54], in particular regarding change detection and diff computation (e.g. [101, 53]). These approaches typically identify basic and complex change operations between different ontology versions. We used this idea to identify time dependent patterns between groups of records to represent the semantics of changes in households over time. Based on the change patterns we are able to realize more comprehensive analysis, e.g., on complex evolution graphs.

6.4 Temporal group linkage

Determining the record and group mappings for the temporal linkage of census datasets is challenging not only due to changing attribute values for the same person (e.g., for surname or occupation) but also due to the high ambiguity and frequent occurrence of certain attribute values, as well as because of data quality issues, e.g., misspelled names, errors for age etc. Group linkage has hardly been studied before [1] and requires a flexible approach to determine many-to-many mappings taking into account that households may split or merge. Similar in spirit to collective entity resolution [8, 111], we determine the similarity between records not only based on attribute values but also considering relationships between records (persons) within a graph-based approach. Furthermore, we not only address record linkage but solve record and group linkage jointly within a combined approach. To better deal with the partially low similarity of matching person records and the need to determine many-to-many group mappings we propose an iterative approach for temporal linkage. We first identify safe matches with a high similarity and then continuously relax the similarity criterion to find additional record and group links.

[1] We are only aware of one approach for group-based linkage of census data [45] that is non-iterative and less sophisticated regarding the use of relationships. In our evaluation in Section 6.6, we will compare the results for this scheme with our approach.

Algorithm 3 describes our approach for determining a group mapping $\mathcal{M}_G^{i,i+1}$ and a record mapping $\mathcal{M}_R^{i,i+1}$ between two successive census datasets D_i and D_{i+1}. The input of the algorithm includes two similarity functions for record matching and parameters for the iterative adjustment of a similarity threshold δ. We first give a high-level description of the algorithm and its main steps. These steps are then explained in more detail in the four following subsections of this section.

At first, we enrich the graphs for each group (household) in the two input datasets by adding implicit relationships between group members, such as derivable family relations. Moreover, we compute for each relationship between persons the age difference as an additional relationship property for later use in the similarity computations.

The main part of the algorithm is a loop to iteratively identify and extend the group mapping $\mathcal{M}_G^{i,i+1}$ and the record mapping $\mathcal{M}_R^{i,i+1}$. In each iteration, we first apply a similarity function Sim_func to determine an initial linking and clustering of person records based on attribute similarities only (pre-matching step). The similarity function Sim_func specifies the person attributes, a weighting vector ω, and a similarity threshold δ (i.e., two persons are considered to match if the weighted sum of their attribute similarities exceeds δ). In the first iteration, we apply a high value δ_high for δ to start with identifying safely matching persons as a basis for also finding safe group matches. Group matches are only determined for pairs of groups connected by at least one (initial) person link. For such group pairs, we apply a *subgraph matching* to determine shared subgraphs with both matching persons and matching relationships. In general, a group of the first census dataset has several candidate group matches in the second dataset so that we select the best group matches considering multiple criteria such as the degree of record and relationship similarity. The matching subgraphs of linked groups are then used to extract the matching records for inclusion into the record mapping (line 10 of Algorithm 3).

Further iterations only process records not yet included in the record mapping determined so far. We continuously relax the similarity threshold by a decrement Δ until a minimal similarity threshold δ_low is reached (or no further group links are identified). Using such relaxed similarity thresholds aims at finding additional matches between records and groups even in the presence of erroneous or

Algorithm 3: Iterative record and group linkage

```
Input:
-$D_i$: old census dataset
-$D_{i+1}$: new census dataset
-$Sim\_func$: similarity function for initial record matching
-$\Delta$: delta for relaxing similarity threshold
-$\delta\_high$: upper bound of similarity threshold
-$\delta\_low$: lower bound of similarity threshold
-$Sim\_func_{rem}$: similarity function for remaining records
Output:
-$\mathcal{M}_R^{i,i+1}$: record mapping
-$\mathcal{M}_G^{i,i+1}$: group mapping
// initialization
$\mathcal{M}_R^{i,i+1} \leftarrow \emptyset, \mathcal{M}_G^{i,i+1} \leftarrow \emptyset$
$\mathcal{M}_R^{p} \leftarrow \emptyset, \mathcal{M}_G^{p} \leftarrow \emptyset$
$G_i \leftarrow$completeGroups ($G_i$)
$G_{i+1} \leftarrow$completeGroups ($G_{i+1}$)
$Sim\_func.\delta \leftarrow \delta\_high$
// iterative subgraph matching
repeat
    // identification of candidates
    $\mathcal{C} \leftarrow$ prematching ($R_i, R_{i+1}, Sim\_func$)
    // subgraph matching and criteria computation
    $Sub_G \leftarrow$subgroups ($\mathcal{C}, G_i, G_{i+1}, Sim\_func$)
    $\mathcal{M}_G^{p} \leftarrow$selectGroupMatches ($Sub_G$)
    // extend group mapping
    $\mathcal{M}_G^{i,i+1} \leftarrow \mathcal{M}_G^{i,i+1} \cup \mathcal{M}_G^{p}$
    // extend record mapping
    $\mathcal{M}_R^{p} \leftarrow$extractRecordMapping ($\mathcal{M}_R^{p}, Sub_G, R_i, R_{i+1}$)
    $\mathcal{M}_R^{i,i+1} \leftarrow \mathcal{M}_R^{i,i+1} \cup \mathcal{M}_R^{p}$
    // extract unlinked records and records that are related to unlinked
       records
    $R_i \leftarrow$ nonMatchedRecords ($R_i, \mathcal{M}_R^{i,i+1}$)
    $R_{i+1} \leftarrow$ nonMatchedRecords ($R_{i+1}, \mathcal{M}_R^{i,i+1}$)
    $Sim\_func.\delta \leftarrow Sim\_func.\delta - \Delta$
until $\mathcal{M}_G^{p} = \emptyset \vee Sim\_func.\delta < \delta\_low$
// match remaining records
$\mathcal{M}_R^{p} \leftarrow$match ($R_i, R_{i+1}, Sim\_func_{rem}$)
$\mathcal{M}_R^{i,i+1} \leftarrow \mathcal{M}_R^{i,i+1} \cup \mathcal{M}_R^{p}$
$\mathcal{M}_G^{i,i+1} \leftarrow \mathcal{M}_G^{i,i+1} \cup extractGroupLinks(\mathcal{M}_R^{p}, G_i, G_{i+1})$
return $< \mathcal{M}_R^{i,i+1}, \mathcal{M}_G^{i,i+1} >$
```

changed attribute values.

After all iterations are performed we have finished subgraph -based group linkage. For the remaining records not yet associated within matching subgraphs, we apply a second attribute-based similarity function Sim_func_{rem} to identify further person links for inclusion into the record mapping (line 17). Moreover, we extend the group mapping by adding the group pairs that are now linked by the newly found record links $\mathcal{M}_G^{i,i+1}$ (line 19).

In the following subsections, we describe the discussed steps in more detail. We start with explaining the preprocessing step to enrich the existing household graphs by implicit relationships and additional relationship properties (Subsection 6.4.1). In Subsection 6.4.2, we describe the pre-matching step of records. In Subsection 6.4.3, we outline our subgraph matching approach to identify common subgraphs. We then introduce the criteria and algorithm used to select the group matches (Subsection 6.4.4).

6.4.1 Group Enrichment

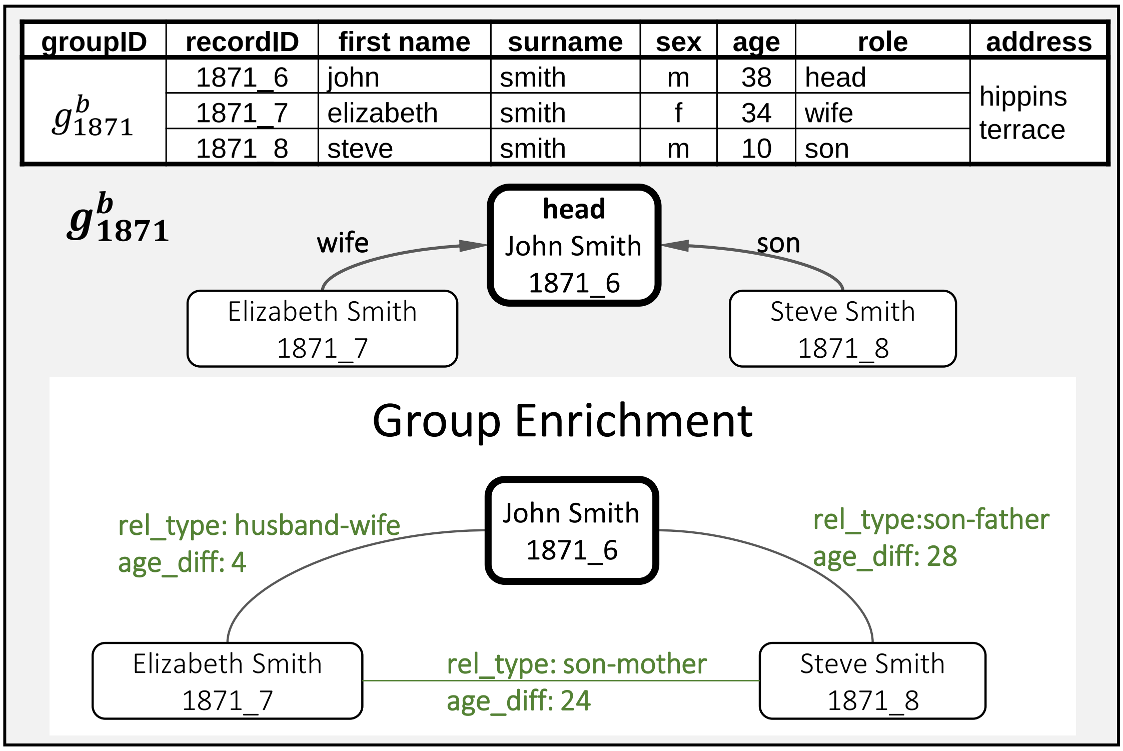
In the initialization phase, we enrich each household group by adding implicit relationships and stable properties such as age differences between persons. In our case, each individual of a household is given a role related to the head of household (which is a special role). This role may not be preserved in future census datasets since individuals may become members of a different household and the head of household may change as well. Hence, comparing households based on these relations only is insufficient in the presence of household changes. We therefore enrich the household graphs by implicit relationships for each record pair of the original group and replace the head-dependent relationship types by a unified type. To increase the semantics of a relationship, we further add the age difference between two household members as a time-independent relationship property. Figure 6.2 shows an example of the group enrichment phase for group g_{1871}^b. The relationship between *Elizabeth Smith* and *Steve Smith* is added. Moreover, the age differences age_diff between persons as well as the relationship types rel_type are added to the relationships.

6.4.2 Pre-Matching

Pre-matching clusters similar records in the census datasets based on their attribute similarity and assigns a cluster label to each record. These labels are utilized to simplify subgraph matching since the labels identify similar records without further similarity computation.

Pre-matching first applies similarity function Sim_func to compare each record of R_i with each record of R_{i+1}. The similarity function specifies the attributes to be compared as well as the attribute-specific similarity function, e.g., q-gram string matching [21]. Furthermore, it uses a weighting vector ω and a required minimum similarity δ. Applying the attribute-specific similarity functions to a pair of records r_i and r_{i+1} results is a similarity vector $\vec{sim}_{(r_i,r_{i+1})}$. Using ω we determine an aggregated similarity $agg\_sim_{(r_i,r_{i+1})}$ by calculating a weighted sum of the attribute similarities:

$$agg\_sim_{(r_i,r_{i+1})} = \omega \cdot \vec{sim}_{(r_i,r_{i+1})} \qquad (6.3)$$



groupID	recordID	first name	surname	sex	age	role	address
g^b_{1871}	1871_6	john	smith	m	38	head	hippins terrace
	1871_7	elizabeth	smith	f	34	wife	
	1871_8	steve	smith	m	10	son	

Figure 6.2: Example of the group enrichment phase for group g^b_{1871}.

We then keep only the record pairs whose similarity is above the specified threshold δ as potential record matches. Furthermore, we determine the transitive closure or connected components of these match pairs (record links) to cluster together all directly and indirectly matching records. We assign to each record of a cluster a unique label, so that records of the same cluster have the same label.

Figure 6.3 shows the resulting clusters for the running example by using the attributes *first name* and *surname*, $\omega = (0.5, 0.5)$ and similarity threshold 1. Pre-matching results in the shown ten clusters where all records of a cluster share the same first name and surname. We then assign the cluster labels A, B etc. to the respective records of the clusters.

Cluster label	recordID	first name	surname
A	1871_1	john	ashworth
	1881_1	john	ashworth
	1881_9	john	ashworth
B	1871_2	elizabeth	ashworth
	1881_2	elizabeth	ashworth
	1881_10	elizabeth	ashworth
C	1871_4	william	ashworth
	1881_3	william	ashworth
	1881_11	william	ashworth
D	1871_6	john	smith
	1881_4	john	smith
E	1871_7	elizabeth	smith
	1881_5	elizabeth	smith
F	1871_8	steve	smith
	1881_6	steve	smith
G	1881_8	mary	smith
H	1871_5	john	riley
I	1871_3	alice	ashworth
K	1881_7	alice	smith

Figure 6.3: Pre-matching result for running example. Records with the same cluster label represent similar records.

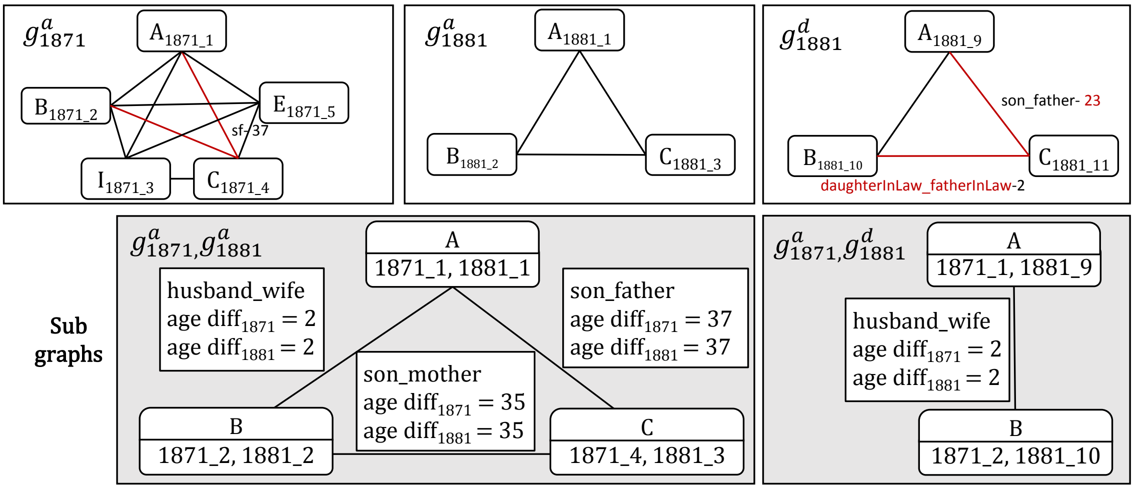


Figure 6.4: Subgraphs for group pairs (g^a_{1871}, g^b_{1881}) and (g^a_{1871}, g^d_{1881}) of the running example. For (g^a_{1871}, g^d_{1881}), the red-coloured edges are not matched due to a different relationship type or non-similar age difference.

6.4.3 Subgraph Matching

Subgraph matching looks for common subgraphs in each pair of groups g_i and g_{i+1} of $G_i \times G_{i+1}$ to determine likely group links. To avoid the computation of the cross product between G_i and G_{i+1}, subgraph matching is only applied for pairs of groups sharing at least one similar record, i.e., having the same cluster label.

The subgraph g_{sub} between two groups g_i and g_{i+1} (represented by their enriched graphs with g_i=(V_i,E_i) and g_{i+1}= (V_{i+1},E_{i+1}) consists of a set of vertices R_{sub} and a set of edges E_{sub}. Each vertex in R_{sub} represents a pair of equally labeled (i.e., similar) records v_i from V_i and v_{i+1} from V_{i+1}. Two vertices $(v1_i, v1_{i+1})$ and $(v2_i, v2_{i+1})$ of R_{sub} are connected by an edge of E_{sub} if both the old records $v1_i$, $v2_i$ and the new records $v1_{i+1}, v2_{i+1}$ of these vertices are connected within their enriched graphs of g_i and g_{i+1}, respectively. Furthermore, we require that these edges must have the same relationship type and highly similar relationship properties, in our case regarding the age differences.

Figure 6.4 illustrates subgraph matching for group g^a_{1871} from the first census dataset and the two groups g^a_{1881} and g^d_{1881} from the second dataset. For the group pair (g^a_{1871}, g^a_{1881}) we have three matching vertices with labels A, B and C. The three edges have the same relationship types and the same or very similar age differences. The second group pair (g^a_{1871}, g^d_{1881}) also shares three vertices with labels A, B and C but only one of the edges has the same relationship type and similar age difference. Hence the common subgraph is reduced to the one shown in the bottom right of Figure 6.4.

6.4.4 Selection of Group Links

Subgraph matching generates candidates for group linkage based on common subgraphs for different group pairs. There may be several linkage candidates per group in G_i and in G_{i+1} so that we have to find the best matching group pairs. The necessary selection should especially guarantee that each record of a group is only linked to one record of another group (This is not the case for the example in Figure 6.4 where we have two linkage candidates for members of group g^a_{1871}). However, a group can link to more than one group if their subgroups are disjoint.

To select for a certain group g_i the best-matching groups in G_{i+1} we consider all subgraphs g_{sub}=(R_{sub},E_{sub}) involving g_i and apply an aggregated similarity measure. This measure combines three scores capturing the record similarity (Equation 6.5), edge similarity (Equation 6.6) and the uniqueness (Equation 6.7) of a subgroup g_{sub}. The results of the similarity functions are aggregated according to Equation 6.4 whereby α determines the influence of record similarity and β represents the weight of edge similarity.

$$g\_sim = \alpha \cdot avg\_sim + \beta \cdot e\_sim + (1 - \alpha - \beta) \cdot unique \tag{6.4}$$

- *Average Record Similarity*

  For this score we determine the average of the aggregated similarities agg_sim for the record pairs of R_{sub}. These aggregated similarities are already determined during pre-matching for each record pair (see section 6.4.2) and can be obtained from the respective clusters in $\mathcal{C}$.

$$avg\_sim(g_i, g_{i+1}, g_{sub}) = \frac{\sum\limits_{(r_i,r_{i+1}) \in R_{sub}} agg\_sim_{(r_i,r_{i+1})}}{|R_{sub}|} \tag{6.5}$$

- *Edge Similarity*
  The edge similarity *e_sim* evaluates the similarity of the relationship properties *rp_sim* in the edges in a subgraph, for example the similarity of the age differences between two individuals in the older group g_i vs. the age difference in the newer group g_{i+1}. Furthermore, we apply an aggregation measure similar to the Dice-Coefficient to relate the edge similarities to the total number of relationships of the considered groups g_i and g_{i+1} thereby giving higher weight to those subgraphs covering a large portion of their relationships.

$$e\_sim(g_i, g_{i+1}, g_{sub}) = 2 \cdot \frac{\sum\limits_{e \in E_{sub}} rp\_sim(oldEdge(e), newEdge(e))}{|E_i| + |E_{i+1}|} \tag{6.6}$$

- *Uniqueness*
  If two group pairs are similar w.r.t both the average record similarity as well as the edge similarity, we like to prefer the group link between the two groups containing records that are less ambiguous than the records of other group pairs. Therefore, we define the uniqueness for a group pair based on the number of vertices of R_{sub} of g_{sub} and the aggregated number of records that are assigned to the same label like the records of R_{sub}. The uniqueness is defined as follows:

$$unique(g_i, g_{i+1}, g_{sub}) = 2 \cdot \frac{|R_{sub}|}{\sum\limits_{r_i \in R_{sub}} |label(r_i)|} \tag{6.7}$$

  The uniqueness of a group pair g_i and g_{i+1} is 1, if the labels are only assigned to the common records of g_i and g_{i+1} and there exists no other record of R_i or R_{i+1} that has the same label.

For the example of Figure 6.4, we obtain the following similarity values for the group pairs (g^a_{1871}, g^a_{1881}) and (g^a_{1871}, g^d_{1881}):

$$
\begin{aligned}
avg\_sim(g^a_{1871}, g^a_{1881}, g_{sub}) &= \frac{1+1+1}{3} = 1\\
e\_sim(g^a_{1871}, g^a_{1881}, g_{sub}) &= 2 \cdot \frac{1+1+1}{10+3} = 0.46\\
unique(g^a_{1871}, g^a_{1881}, g_{sub}) &= 2 \cdot \frac{3}{3+3+3} = 0.66\\
\\
avg\_sim(g^a_{1871}, g^d_{1881}, g_{sub}) &= \frac{1+1}{2} = 1\\
e\_sim(g^a_{1871}, g^d_{1881}, g_{sub}) &= 2 \cdot \frac{1}{10+3} = 0.15\\
unique(g^a_{1871}, g^d_{1881}, g_{sub}) &= 2 \cdot \frac{2}{3+3} = 0.66
\end{aligned}
\tag{6.8}
$$

The aggregated similarity of these values reaches a higher value for group pair (g^a_{1871}, g^a_{1881}) than for (g^a_{1871}, g^d_{1881}) due to the higher edge similarity of the former pair. As a result, we would only include group pair (g^a_{1871}, g^a_{1881}) in the group mapping and derive the record mapping only for the common subgraph of this pair.

After the determination of the introduced similarity values per subgroup, we apply Algorithm 4 for the selection of the best-matching group pairs. The algorithm follows a greedy strategy by considering subgraphs in the order of their aggregated similarity score. It also considers the disjointness of subgraphs and can determine group mappings of cardinality N:M.

In each iteration, we select the group pair with the highest group similarity from a priority queue pq. The selected pair (g_i, g_{i+1}) is added to the group mapping $\mathcal{M}^p_G$ if the overlap between the already linked records of g_i as well as g_{i+1} and the records of the record pairs of g_{sub} is empty (line 12). Thus, we ensure that a record is linked at most to one record. The linked records are represented by $linked\_R_i$ resp. $linked\_R_{i+1}$. Moreover, the records of g_i and g_{i+1} that correspond to a record pair of R_{sub} of g_{sub} are represented by the sets R^i_{sub} and R^{i+1}_{sub}. These sets are returned by *getOldRecords* and *getNewRecords* respectively for a certain subgroup g_{sub}. If a group link is added, we update sets of linked records $linked\_R_i$ and $linked\_R_{i+1}$ for g_i resp. g_{i+1} (line 14 to 17).

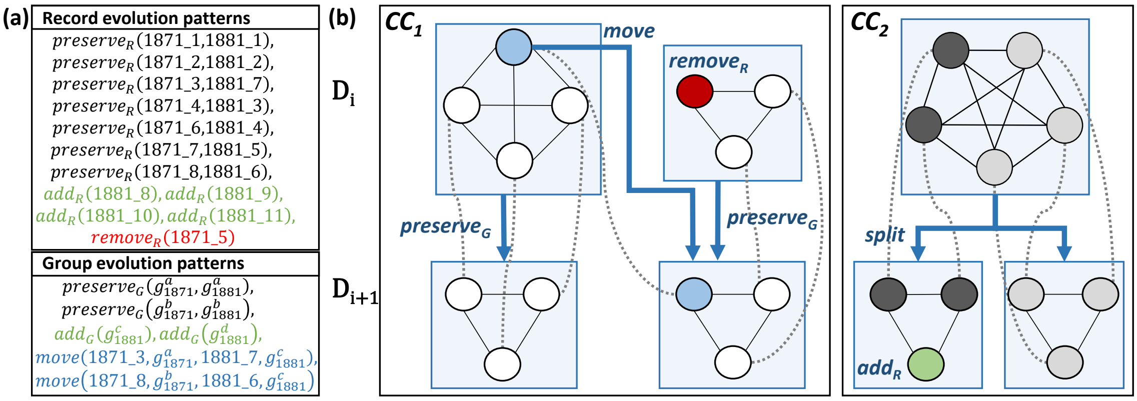
Algorithm 4: Selection of group links

```
Input:
-$Sub_G$: set of quadruples of <$g_i, g_{i+1}, g_{sub}, g\_sim$>
Output:
-$\mathcal{M}_G^p$: partial group mapping
$\mathcal{M}_G^p \leftarrow \emptyset$
$lookup \leftarrow \emptyset$
// initialize priority queue ordered by $g\_sim$
for $(g_i, g_{i+1}, g_{sub}, g\_sim) \in Sub_G$ do
    $pq \leftarrow pq.insert(g_i, g_{i+1}, g_{sub}, g\_sim)$
while $pq \neq \emptyset$ do
    $< g_i, g_{i+1}, g_{sub}, g\_sim > \leftarrow pq.max()$
    $pq \leftarrow pq.remove()$
    // sets of linked records of $g_i$ and $g_{i+1}$
    $linked\_R_i \leftarrow lookup.get(g_i)$
    $linked\_R_{i+1} \leftarrow lookup.get(g_{i+1})$
    // records of $g_i$ and $g_{i+1}$ contained in $g_{sub}$
    $R_{sub}^i \leftarrow getOldRecords(g_{sub})$
    $R_{sub}^{i+1} \leftarrow getNewRecords(g_{sub})$
    if $linked\_R_i \cap R_{sub}^i = \emptyset \wedge linked\_R_{i+1} \cap R_{sub}^{i+1} = \emptyset$ then
        $\mathcal{M}_G^p \leftarrow \mathcal{M}_G^p \cup \{(g_i, g_{i+1})\}$
        $linked\_R_i \leftarrow linked\_R_i \cup R_{sub}^i$
        $linked\_R_{i+1} \leftarrow linked\_R_{i+1} \cup R_{sub}^{i+1}$
        $lookup \leftarrow lookup.update(g_i, processed\_R_i)$
        $lookup \leftarrow lookup.update(g_{i+1}, processed\_R_{i+1})$
return $\mathcal{M}_G^p$
```

Based on the selected group matches, we are able to identify the record matches contained in the corresponding subgraph g_{sub}. The record links are included in each vertex of g_{sub} since R_{sub} is defined as a set of pairs r_i and r_{i+1}. These pairs are the most appropriate links since the related groups are linked.

6.5 Evolution analysis

We will now use the results of the temporal record and group linkage to detect changes between different census datasets in order to support the comprehensive evolution analysis of temporal census data. Such a change analysis should not be restricted to a low-level evaluation of individual links but should be realized at



(a)

Record evolution patterns

$preserve_R$(1871_1,1881_1),
$preserve_R$(1871_2,1881_2),
$preserve_R$(1871_3,1881_7),
$preserve_R$(1871_4,1881_3),
$preserve_R$(1871_6,1881_4),
$preserve_R$(1871_7,1881_5),
$preserve_R$(1871_8,1881_6),
add_R(1881_8), add_R(1881_9),
add_R(1881_10), add_R(1881_11),
$remove_R$(1871_5)

Group evolution patterns

$preserve_G(g^a_{1871}, g^a_{1881})$,
$preserve_G(g^b_{1871}, g^b_{1881})$,
$add_G(g^c_{1881})$, $add_G(g^d_{1881})$,
$move$(1871_3, g^a_{1871}, 1881_7, g^c_{1881}),
$move$(1871_8, g^b_{1871}, 1881_6, g^c_{1881})

Figure 6.5: (a) Record and group evolution patterns for the running example. (b) Evolution graph and patterns for two successive census datasets D_i and D_{i+1}. Gray dotted lines represent record links, blue arrows indicate evolution patterns between related households.

a higher, application-specific level to generate relevant and expressive change patterns. We will also include disappearing as well as newly appearing records and groups that are not reflected in the identified mappings but appear only in one of the census datasets. The analysis should further not be limited to two datasets but involve a series of successive census datasets covering longer periods of time.

In this initial study, we use the given census datasets and the determined linkage results to identify a set of basic and more complex changes for records and groups of records that can be identified with the help of so-called evolution patterns (Subsection 6.5.1). Furthermore, we propose the use of a so-called evolution graph (Subsection 6.5.2) to provide an aggregated change representation that is extensible to more than two census datasets. Such an evolution graph is a promising basis for advanced graph mining techniques, e.g., to determine frequent or unusual change scenarios.

6.5.1 Evolution Patterns

We define evolution patterns on individual records and on groups of records. There are three *record evolution patterns* called $preserve_R$, $remove_R$ and add_R. We identify these patterns by utilizing the record mapping $M_R^{i,i+1}$ as well as record sets R_i and R_{i+1} for two successive census datasets D_i and D_{i+1} as follows:

- *preserve*$_R$ is a record pair representing one individual linked between R_i and R_{i+1}.
  $\forall r_i, r_{i+1} \in R_i \times R_{i+1}$:
  $preserve_R(r_i, r_{i+1}) \leftrightarrow \exists (r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$
- *add*$_R$ denotes an individual $r_{i+1} \in R_{i+1}$ that is not linked to any record of R_i.
  $\forall r_{i+1} \in R_{i+1} : add_R(r_{i+1}) \leftrightarrow \nexists (r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$
- *remove*$_R$ denotes an individual $r_i \in D_i$ that is not linked to any record of D_{i+1}.
  $\forall r_i \in R_i : remove_R(r_i) \leftrightarrow \nexists (r_i, r_{i+1}) \in \mathcal{M}_R^{i,i+1}$

To analyze the dynamics of groups, we further define *group evolution patterns* based on changes within groups. These patterns are *add*$_G$ and *remove*$_G$ as well as the more complex patterns *preserve*$_G$, *move*, *split* and *merge*. The patterns *preserve*$_G$ and *move* both relate to pairs of linked groups but differ on whether the linked groups contain at least two preserved members (*preserve*$_G$) or only one (*move*). Each pattern is identified by utilizing the census datasets, the group mapping $\mathcal{M}_G^{i,i+1}$ and the record mapping $\mathcal{M}_R^{i,i+1}$:

- *add*$_G$ denotes a new group $g_{i+1} \in G_{i+1}$ that did not exist in D_i. Thus, the group mapping $\mathcal{M}_G^{i,i+1}$ does not contain any link with g_{i+1}.
- Similarly, *remove*$_G$ contains a group of $g_i \in G_i$ that does not exist in G_{i+1} anymore.
- *preserve*$_G$ is a group pair connected by a 1:1 link. Moreover, each group consists of at least 2 individuals satisfying the *preserved*$_R$ pattern. This condition allows us to identify preserving households across censuses. The requirement that a 'preserved' household should have at least two remaining members is influenced by real-world situations such as households where only the parents remain after their children have moved to another household.
- *move* identifies pairs of linked groups with only one member in common (determined by the *preserve*$_R$ pattern) that has moved from the old to the new group (household).

- *split* identifies a change situation between a group $g_i \in D_i$ from the old dataset and a set of groups $g^a_{i+1}, g^b_{i+1}, ..., g^k_{i+1} \in G_{i+1}$ in the new dataset, where at least two individuals of g_i must overlap with each of the groups from G_{i+1}. Note, that each individual record can only be contained in one group, i.e., $g^a_{i+1}, g^b_{i+1}, ..., g^k_{i+1}$ are disjoint.

- *merge* covers the opposite situation between a set of groups $g^a_i, g^b_i, ..., g^k_i \in G_i$ from the old dataset and one group $g_{i+1} \in G_{i+1}$ from the new dataset, where at least two individuals from groups in G_i must overlap with the merged group g_{i+1}. Each individual record can only be contained in one group, i.e., $g^a_i, g^b_i, ..., g^k_i$ are disjoint.

Figure 6.5(a) shows the corresponding record and group evolution patterns for our running example from Figure 6.1. Seven records have been preserved from D_{1871} to D_{1881}. Moreover, there are 4 record additions and one removal. According to the defined group evolution patterns, two groups have been preserved (g^a and g^b), two groups newly appeared in 1881 (add_G for g^c and g^d) and two persons, Alice (1871_3) and Steve (1871_8), moved from their parents' households (g^a_{1871} and g^b_{1871}) to their own new household g^c_{1881}.

6.5.2 Evolution Graph

Based on the evolution patterns we want to realize further comprehensive evolution analyzes for dynamically changing family structures and individual person histories. We propose the use of a so-called *evolution graph* reflecting the history of households across two or more successive census datasets. The graph $\mathcal{G}\_Evolution$ captures both the records and groups per census dataset as vertices and interconnects them across successive datasets by edges that are typed according to the identified evolution patterns (change types). Figure 6.5(b) shows a sample evolution graph and evolution patterns for two successive versions D_i and D_{i+1}. Blue boxes represent group vertices and blue arrows represent group evolution patterns, i.e., the changes between households. Two groups have been preserved and are linked via the group pattern $preserve_G$ and one household has been split into two households. One individual moved between two households that are thus connected in the evolution graph. The figure also shows the mapping between individual records (gray dotted lines) as well as a new (add_R) and a removed ($remove_R$) record without incoming/outgoing edges.

The evolution graph enables the application of several graph mining approaches such as cluster analysis, pattern matching or finding frequent subgraphs. One analysis might be to identify households that are preserved across several census periods. A second use case is to identify clusters of related households that can be used for studies of genetic diseases. In Figure 6.5(b), a simple computation of connected components on the exemplary evolution graph for two points in time leads to two components consisting of 4 (CC_1) and 3 (CC_2) households, respectively. Running such a computation for larger households graphs for many successive versions can produce longer chains of connected households, e.g., indicating relationships between many generations of families.

6.6 Evaluation

In this section, we evaluate the introduced approaches for temporal record and group linkage for different historical census datasets from the UK that have also been used in a previous study [45]. We first describe these datasets and the evaluation setup in Subsection 6.6.1. We then evaluate the linkage quality of the new approaches for different configurations (Subsection 6.6.2). In Subsection 6.6.3 we compare our approach with the results of the previous study [45] as well as with the collective record linkage approach [73]. Finally, we discuss results of an initial evolution analysis for the considered census datasets.

6.6.1 Datasets and Setup

In our evaluation, we use six census datasets collected from 1851 to 1901 in ten-year intervals from the district of Rawtenstall in North-East Lancashire in the United Kingdom. Table 6.1 shows an overview of these datasets according to the number of records and households for the different time periods. The table also shows the number of unique value combinations of the first name and surname attributes to illustrate the degree of ambiguity for these attributes. Furthermore, we report the ratio of missing attribute values. The table shows that the number of households and persons has almost doubled within the 50 years period indicating a substantial population growth. There is a high degree of name ambiguity since each combination of first name and surname is far from unique but has an average frequency of up to 2.23 (for 1851) with a highly skewed frequency distribution due to the presence of frequent surnames such as *Ashworth* and *Smith*. Up to 6.5% of the attribute values are missing, which leads to in additional difficulties for finding correct temporal links.

Table 6.1: Overview of the census datasets according to the number of records, households, unique combinations of first name and surname $|fn + sn|$ and the ratio of missing values $ratio_{mv}$.

t_i	1851	1861	1871	1881	1891	1901
$\lvert R_{t_i}\rvert$	17033	22429	26229	29051	30087	31059
$\lvert G_{t_i}\rvert$	3298	4570	5576	6025	6378	6842
$\lvert fn + sn\rvert$	7652	10198	13198	15505	17130	19910
$ratio_{mv}$	4.67%	4.19%	3.03%	4.09%	6.33%	6.51%

To evaluate the quality of the group and record mappings in terms of precision, recall and F-measure [21], we use the reference mapping determined in [45]. It covers a subset of 1250 matching households from the 1871 and 1881 datasets that consist of 6864 and 6851 members resp. These household were manually linked by experts by focusing on person records found in both datasets.

In our evaluation, we compare different settings for the similarity function considering the string similarity for five attributes and different weight vectors ω_1 and ω_2 as shown in Table 6.2. We also evaluate different similarity thresholds for pre-matching as well as different weights for determining the aggregated group similarity for selecting group links.

Table 6.2: Compared set of attributes and the corresponding weighting vector ω to identify the set of blocks $\mathcal{B}$ that are used for the subgraph matching.

Attribute	Matching method	ω_1	ω_2
First name	q-gram	0.2	0.4
Sex	exact	0.2	0.2
Surname	q-gram	0.2	0.2
Address	q-gram	0.2	0.1
Occupation	q-gram	0.2	0.1

Table 6.3: Quality of group and record mappings for different weighting vectors ω and lower bounds δ_low.

parameter	ω	ω_1				ω_2			
	δ_low	0.4	0.45	0.5	0.55	0.4	0.45	0.5	0.55
group mapping	Precision (%)	96.1	96.5	96.7	97.0	97.1	97.1	**97.3**	**97.3**
	Recall (%)	92.2	92.2	92.0	91.7	94.8	**94.8**	**94.8**	94.6
	F-measure (%)	94.1	94.3	94.3	94.2	96.0	96.0	**96.0**	95.9
record mapping	Precision (%)	96.6	96.8	96.8	96.8	97.5	97.5	**97.5**	**97.5**
	Recall (%)	91.9	91.9	91.9	91.8	93.7	93.7	**93.7**	93.7
	F-Measure (%)	94.2	94.3	94.3	94.3	95.6	95.6	**95.6**	95.5

Table 6.4: Quality of the group and record mappings for different weights α and β to select matching groups.

parameter	(α, β)	(1.0,0.0)	(0.0,1.0)	(0.5,0.5)	(0.33,0.33)	(0.2,0.7)
group mapping	Precision (%)	92.3	96.7	96.6	96.7	**97.3**
	Recall (%)	89.1	94.1	94.3	94.4	**94.8**
	F-Measure (%)	90.7	95.4	95.5	96.0	**96.0**
record mapping	Precision (%)	96.2	97.4	97.3	97.3	**97.5**
	Recall (%)	89.8	93.4	93.4	93.4	**93.7**
	F-Measure (%)	92.9	95.4	95.3	95.3	**95.6**

6.6.2 Linkage Evaluation

We first analyze the influence of different similarity functions during pre-matching and then discuss the impact of different similarity functions for selecting matching group pairs. Afterwards we study the effectiveness of incremental linkage.

Influence of pre-matching configuration

The proposed linkage approach builds on the initial record matching and clustering performed in the pre-matching step. We thus start our analysis by comparing the results for determining the attribute similarities based on the two weighting schemes ω_1 and ω_2 (Table 6.2) and different lower similarity threshold bounds δ_low. For iterative matching we use a start value $\delta\_high = 0.7$ for the similarity threshold δ and $\Delta = 0.05$ for decrementing the threshold until the minimal value δ_low is reached.

Table 6.3 shows the resulting group and record mapping quality in terms of precision, recall and F-measure for the two weighting schemes and four values of δ_low ranging from 0.4 to 0.55. We observe for all configurations high F-Measure results between 94% and 96% for both the determined record mappings and the group mappings, indicating a very high effectiveness of the proposed approach. The best F-measure results are generally achieved for $\delta\_low = 0.5$, although the differences are small for the other choices. The simple weighting scheme ω_1 giving equal weight to each of the five considered attributes is consistently outperformed by the alternate approach giving higher weight to attribute *first name* and only reduced weight for the less stable attributes *address* and *occupation*. Prematching with weight vector ω_2 thus improves F-measure by around 1.7% for the group mapping and up to around 1.3% for the record mapping.

Of course, there are many more possibilities to define the similarity function and we could also apply learning-based methods to find a near-optimal weight vector [21]. Still our results show that using the similarity function with weight vector ω_2 and $\delta\_low = 0.5$ achieve good and stable results making it an effective default configuration.

Similarity weights for selecting matching groups

We now evaluate the influence of the different weights α and β for determining the aggregated group similarity $g\_sim = \alpha \cdot avg\_sim + \beta \cdot e\_sim + (1 - \alpha - \beta) \cdot rel$ driving the selection of matching groups. Table 6.4 shows the results of the different weights. The quality of the group mapping highly depends on the edge similarity underlining the importance of considering the structural similarity within our household graphs. Without considering the edge similarity ($\beta = 0$), the F-measure for the group mapping drops to 90.7%, i.e. around 5.3% less than for the best configuration ($\alpha = 0.2, \beta = 0.7$) and also far less than when ignoring the record similarity ($\alpha = 0$). The uniqueness score can also improve the overall F-measure. For ($\alpha = 0.2, \beta = 0.7$) its weight is 0.1 which helped to achieve an improved F-measure compared to the three configurations where it is ignored (when the sum of α and β equals already 1). The best record mapping is also achieved for ($\alpha = 0.2, \beta = 0.7$) making it a good default configuration for our datasets.

Table 6.5: Quality of the group mapping and record mapping by using the iterative vs. non-iterative approach.

method		non-iterative	iterative
group mapping	Precision (%)	94.5	**97.3**
	Recall (%)	93.1	**94.8**
	F-measure (%)	93.8	**96.0**
record mapping	Precision (%)	91.8	**97.5**
	Recall (%)	93.1	**93.7**
	F-measure (%)	92.5	**95.6**

Iterative vs non-iterative linkage

We now want to analyze to what degree the iterative group and record linkage with decreasing similarity thresholds is really helpful compared to a non-iterative, one-shot approach applying only a fixed minimal similarity threshold. To evaluate such a non-iterative approach we apply similarity functions with ω_2, $\delta\_high = 0.5$ and $\delta\_low = 0.5$ resulting in only one iteration. The results are shown in Table 6.5. We observe that the iterative approach indeed outperforms the non-iterative approach with an F-Measure improvement of ≈ 2.2% for the group mapping and 3.1% for the record mapping. The improved quality mainly results from a substantially higher precision of more than 97% for both the group and record mapping. This is achieved because the iterative approach finds high-quality matches for the more restrictive thresholds while the more relaxed similarity threshold, with an increased risk of finding wrong matches, is limited to a subset of the records.

6.6.3 Comparison with Existing Approaches

We compare our approach with two previously proposed methods: the collective entity resolution approach of [73] to determine a record mapping as well as the previous group linkage approach [45] for census data.

In [73], the authors propose a collective approach that is a specialization of [8]. It initially determines seed record links by applying a high record similarity. The seed links are used to incrementally identify additional links from the neighborhood of the linked records based on their attribute similarity and relational sim-

Table 6.6: Comparison of our approach with the collective linkage approach of [73] (CL) to determine a record mapping.

method	CL	iter-sub
Precision (%)	93.5	**97.5**
Recall (%)	81.2	**93.7**
F-measure (%)	86.9	**95.6**

ilarity. The overall algorithm follows a greedy strategy that selects in each iteration the record pair with the highest similarity. The related records update their similarities according to the selected record pair. In our implementation, we use the same similarity function as in our approach (Table 6.2). Moreover, we filter all record pairs where the normalized age difference is more than 3 years[2]. To generate the seed link, we select the record links with a minimal similarity of 0.9. Table 6.6 shows the results of the record mapping obtained by collective linking. Our approach outperforms the collective approach w.r.t the record mapping quality by 8.6% for F-measure. The difference between our approach and the collective approach is that we can better link moved records with changed attribute values since we do not only link highly similar records (which is not sufficient for temporal linkage). Furthermore, our subgraph matching utilizes different relationships more comprehensively and benefits from incremental linkage.

The previous group linkage approach of [45] initially generates a highly selective record mapping consisting of 1:1 correspondences only. Based on this record mapping, the method calculates an average record similarity and an edge similarity between each group pair. Contrary to our approach, they calculate the similarities based on the initial 1:1 mapping. If correct record pairs are filtered out due to the 1:1 constraint, the approach is not able to identify these links. Hence, this filter step influences the average record similarity as well as the edge similarity, so that correct group links are not identified. Table 6.7 shows the results of the quality of the group mappings. Our approach achieves a significantly better F-measure for the group mapping compared to [45] (≈3.7%). This improvement is mainly because of a much higher recall that is limited in the previous approach mainly because of the use of the initial 1:1 mapping.

[2] In our approach, subgraph matching ensures that such age differences are not accepted.

Table 6.7: Comparison of our approach with the household linkage approach of [45] (GraphSim).

method	GraphSim	iter-sub
Precision (%)	**97.6**	97.3
Recall (%)	90.1	**94.8**
F-measure (%)	93.7	**96.0**

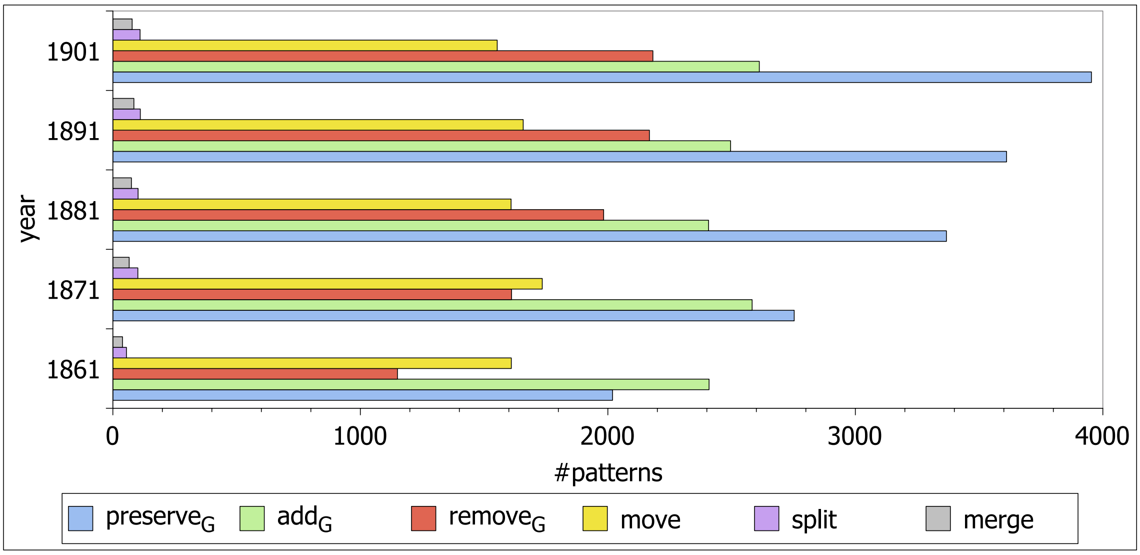


Figure 6.6: Quantitative Analysis of evolution patterns for census datasets from 1851 to 1901.

6.6.4 Analysis of Household Dynamics

Finally, we analyze the evolution of households from 1851 to 1901. For this purpose, we determine the evolution patterns for each successive census dataset pair based on the identified group and record mapping with the best parameter setting. Figure 6.6 shows the frequency of each group evolution pattern for each pair of census datasets. In general, we observe an increasing number of households since the number of add_G patterns is higher than the number of $remove_G$ patterns for each new census. Moreover, we observe an increasing number of $preserve_G$ patterns due to the general increase in the number of households over time. From 1891 to 1901, there is also a high number of $remove_G$ patterns (up to $\approx$ 2200) indicating that many households may have moved to a new region. The complex patterns such as *split* and *merge* occur only rarely with an average occurrence of $\approx$ 100 for *split* and $\approx$ 70 while the *move* patterns are more frequent ($\approx$ 1600 on average).

Table 6.8: Number of preserving households $|preserve_G|$ according to different time intervals (in years) from 1851 to 1901.

time interval	$\|preserve_G\|$
10	15705
20	7731
30	3322
40	1116
50	260

To analyze dependencies between households for the whole time period, we exploit the evolution graph and determine the largest connected component representing all households from 1851 to 1901 that are connected by group patterns. We identified the largest connected component with 17150 households over the complete interval from 1851 to 1901 thereby covering ≈52% of all households. Furthermore, we identify the number of preserved households according to different time intervals for the whole time period from 1851 to 1901. For instance, if we like to identify households that are preserved for 20 years, we define a graph pattern that consists of 2 edges with the pattern type $preserve_G$ since the difference between two census datasets is 10 years. Table 6.8 shows the number of preserved households for the different time intervals. The number of preserving households for all 10 year intervals (1851-61, 1861-71, 1871-81 etc.) represents the overall number of $preserve_G$ patterns of the quantitative analysis. Moreover, 260 household are preserved over the whole time period from 1851 to 1901.

6.7 Conclusion

In this chapter, we outlined and evaluated a new approach for temporal record and group linkage for the analysis of census data. The approach follows an iterative linkage strategy that first identifies high quality links thereby limiting the more error-prone identification of links between less similar records and groups to subsets of the input data. Group linkage is based on the identification of common subgraphs between groups such as households where we utilize the semantic relationships within groups and relationship properties such as the age differences between individuals. The evaluation showed the high effectiveness of the proposed approach that also outperforms a previous approach for linking census data.

We showed that the linkage results support a detailed evolution analysis of census data at both the level of individuals and groups. We proposed several evolution patterns to identify relevant changes including different kinds of group changes such as splits, merges and the movement of individuals from one group to another. All changes can be maintained within an evolution graph that can be used for a wide spectrum of change analysis, e.g., to identify frequent change patterns or to find connected groups over several census periods.

Nevertheless, the quality of initial links can be increased using machine learning techniques. However, these techniques need training data that is not often available. Moreover, the training data must be representative so that the resulting classification models generalize to uncertain links and generate qualitative results. Therefore, we propose an active learning approach in the following chapter, that reduce the number of training data as well as select representative links.

7

Informativeness-Based Active Learning for Entity Resolution

Preamble

This chapter is based on [24]. We propose an active learning approach for entity resolution that selects training data based on the location in the vector space instead of using intermediate classification results.

7.1 Motivation

Entity Resolution (ER) is the task of identifying pairs of records from different data sources that refer to the same real-world entities [21]. ER is a crucial step for different application domains such as census analysis, national security, and the health, life, and social sciences. The quality and usefulness of any data analysis based on linked data highly depends upon how accurate ER was conducted.

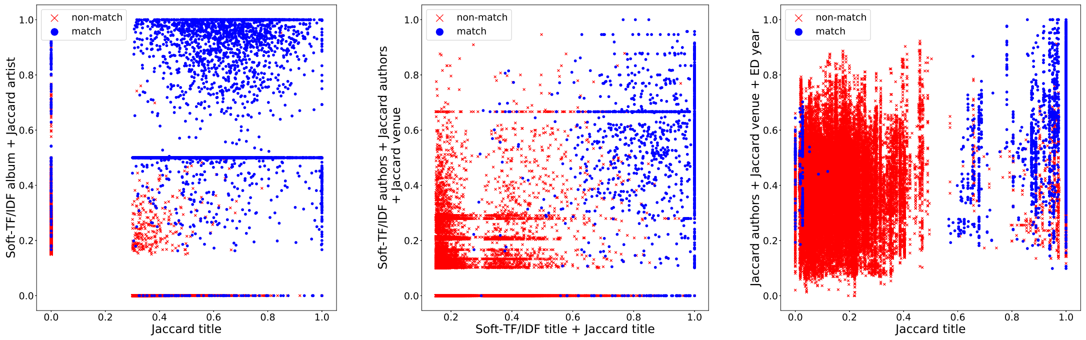
To identify pairs of records that refer to the same entity, the attributes of records are generally compared using similarity functions such as approximate string comparators [21]. A crucial part of ER is the classification of two records as a *match* (same entity) or *non-match* (different entities) based on the calculated similarities between them. Machine learning approaches [66, 123] can learn a classifier over sets of known matching and non-matching record pairs based on the similarities of their attributes as represented by a *similarity* or *weight vector*. For example, comparing first name, last name, street address, city and zipcode leads to a five-dimensional similarity vector per compared record pair [21].

To generate a classification model, labelled pairs of records are necessary. This however might require significant manual labelling efforts [138]. Moreover, the number of true matches (record pairs that refer to the same entity) is generally very small compared to the number of non-matching pairs because of the quadratic nature of the comparison space [21], and therefore the selection of labelled pairs is challenging if one wants to learn an unbiased classifier [39]. Active learning techniques promise to minimise the labelling effort as well as to select representative pairs that result in a good classifier.

Previous work in active learning for ER [3, 6, 99, 138] has focused on selecting pairs based on a certain classification model and the resulting decision boundary of the learned classifier. In this paper, we propose a novel active learning approach for ER that considers the covered similarity vector space and the relationships between similarity vectors.

The main idea of our approach is to search for new unlabelled similarity vectors around *informative* similarity vectors that already are classified as matches or non-matches. In this process, we introduce an informativeness measure for a similarity vector based on the current training dataset. The most informative vectors are then used to define a search space where new vectors are selected. We specifically make the following contributions:

- We propose an active learning technique for ER that iteratively selects new similarity vectors for manual classification by an oracle independent of any classifier using an informativeness measure. This measure is based on information entropy to characterise the relationship between vectors labelled as matches as well as non-matches. Moreover, the measure considers uncertainty so that new areas in the similarity vector space are queried.



(a) Music

(b) Google Scholar

(c) Cora

Figure 7.1: Examples of similarity vectors where the monotonicity assumption does not hold. The three plots show similarity vectors of the datasets we use in our evaluation in Section 7.5. If an axis represents more than one similarity, they are summed and normalised into [0,1].

- Our active learning technique is able to generate training data using a budget-limited human oracle [138], and it does not require any prior knowledge about true matches and non-matches.

- We evaluate our active learning technique on three datasets from different application domains. Our results show that our proposed approach outperforms a previous budget-limited active learning approach for ER [138] and achieves classification quality comparable to fully supervised approaches.

In the following we discuss work related to our approach. In Section 7.3 we formalise the problem that we aim to solve with our approach, which we describe in detail in Section 7.4. In Section 7.5 we then experimentally evaluate our approach and compare it with existing active learning as well as supervised methods for ER.

7.2 Related Work

ER is an essential part of data integration in various domains such as e-commerce, health and social science research, or national security. As a result, ER has been intensively studied [21, 67, 96, 98]. One challenge of ER is the quality of the data sources and their heterogeneity [109]. In order to overcome this problem, supervised as well as unsupervised approaches have been proposed [20, 66, 123]. Unsupervised approaches utilise clustering methods to identify groups of similar records that refer to the same entity. In contrast, supervised ER approaches require and use a training dataset consisting of verified true matches and true non-matches to build a classifier. In general, unsupervised methods perform worse than supervised approaches as shown by extensive studies [68], where supervised approaches are able to achieve high ER quality for different domains such as consumer products, bibliographic records, and census data.

A crucial part of supervised approaches is the amount and quality of data available for training, because a non-informative or not representative training dataset can result in biased, over-fitted, or inaccurate classifiers.

To overcome such issues, active learning techniques [3, 6, 99, 138] have been applied to minimise the labelling effort and to select representative record pairs for manual classification. An active learning approach is an iterative process [31] where in each iteration a number of informative and unlabelled training instances are selected that are then manually classified by a human oracle. Many active learning approaches determine informative instances using the distance between instances [134] or their entropy [119] according to a certain classification model.

Previous work in active learning for ER [3, 6] allows to specify a minimum required precision threshold, where the aim of these approaches is to then maximise the recall of the resulting classifier based on the selected record pairs. However, these approaches have the underlying assumption of monotonicity of precision which implies that a record pair with higher similarity is more likely to be a match than a pair with a lower similarity.

Recent work by Wang et al. [138] however has shown that the assumption of monotonicity does not generally hold. We validate this in Figure 7.1 which shows the distribution of true matches and non-matches for three datasets according to their similarities. As can be seen, in each dataset there are clear examples that violate the monotonicity assumption. Therefore, Wang et al. proposed a cluster based active learning approach that iteratively selects record pairs from a cluster. In each iteration, a cluster is processed by selecting a set of record pairs to be labelled by a human oracle. The labelled vectors are then added to the final training dataset if the purity of the current cluster is above a user defined threshold. Otherwise, the cluster is split into two by classifying the unlabelled vectors of the current cluster based on the current classifier. The authors showed that their approach requires less examples than earlier active learning approaches for ER while achieving similar classification accuracy.

In comparison to our proposed approach, the selected examples by Wang et al. [138], and thus the resulting training dataset, depend upon the applied classification model, and therefore the resulting ER quality can vary depending upon the classifier employed in this active learning approach.

Ngonga-Ngomo et al. [99] proposed a generation method of link specifications representing a complex match rule using genetic programming by iteratively improving a set of determined link specifications representing match rules. In each iteration, new examples are selected based on the disagreement according to the current link specification (for example, if 5 of 10 specifications classify a match for a record pair the disagreement is high). A disadvantage of this approach is that the generation of link specifications is not deterministic.

Related to active learning is crowd-sourced based ER [46, 94, 125, 139], where ambiguous or controversial matches are resolved by evaluating votes from a crowd of human evaluators. Mozafari et al. [94] proposed two such approaches, named *Uncertainty* and *MinExpError*, being applicable for applications beyond ER. The main idea of these approaches is to use non-parametric bootstraping to estimate the uncertainty of classifiers. However, crowd-sourcing techniques that rely on a large number of human resources (often non-experts) cannot be used for sensitive data, such as personal health, financial, crime, or government records, where only a small number of experts have access to the data.

In contrast to previous work, our approach is independent of the classification model used to determine informative examples, because we characterise the informativeness of similarity vectors by considering the relationships between vectors within the vector space, as well as the relationships between unlabelled and already labelled vectors. Moreover, our work does not rely upon the monotonicity assumption that does not hold for many ER problems [138].

7.3 Problem Definition

Active learning approaches aim to reduce the manual efforts required for selecting training data, while keeping the quality of ER classification at a high level [3, 6, 138]. In general, the goal of ER is to identify matches $m_i \in \mathbf{M}$ for a set of records $\mathbf{R}$ from one or multiple data sources, where each $m_i = (r_x, r_y)$, with $r_x, r_y \in \mathbf{R}$ and $r_x \neq r_y$. To determine a match for a record pair (r_x, r_y), the set of attributes $\mathbf{A} = \{A_1, ..., A_n\}$ characterising these records is used to calculate similarities $s_1, ..., s_n$ between attribute values. Similarity functions $f_j(r_x.A_j, r_y.A_j)$, with $1 \leq j \leq n$, are used to measure how similar the values in attribute A_j are. We assume each similarity function f_j maps into $[0, 1]$, where 1 means two attribute values are the same and 0 means they are completely different [21].

A similarity or weight vector $\mathbf{w} \in [0, 1]^n$ consists of the calculated n similarities between the attributes in $\mathbf{A}$. For example, the two records r_1 and r_2 characterised by the attributes $\mathbf{A} = \{surname, address\}$ with r_1*.surname="ashworth"*, r_1*.address="fern hill"* and r_2*.surname="ashwort"*, r_2*.address="fearn hill"* might results in a similarity vector $\mathbf{w} = \langle 0.74, 0.78 \rangle$ when using approximate string comparison functions such as edit distance [21].

The goal of an active learning approach is to identify a set of classified similarity vectors $\mathbf{T} \subset \mathbf{W}$ for a given set of unclassified vectors $\mathbf{W}$, where $\mathbf{T}$ consists of *matches* and *non-matches* and is used as training data to learn a classifier. Our approach considers a predefined budget b of the total number of similarity vectors that can be labelled by a human oracle. The approach selects in each iteration a predefined number k of vectors where the selection depends on the informativeness of each vector in $\mathbf{T}$ and the vector space covered by $\mathbf{T}$.

As detailed below, to measure the informativeness $info(\mathbf{w}_i, \mathbf{T})$, of a vector $\mathbf{w}_i$, we consider the relationship of $\mathbf{w}_i$ to vectors $\mathbf{w}_k \in \mathbf{T} \backslash \{\mathbf{w}_i\}$, where we calculate the similarity between two vectors $\mathbf{w}_i$ and $\mathbf{w}_k$ using the Cosine similarity defined as $sim(\mathbf{w}_i, \mathbf{w}_k) = \frac{\mathbf{w}_i \cdot \mathbf{w}_k}{||\mathbf{w}_i|| \cdot ||\mathbf{w}_k||}$. We assume that the area around a vector $\mathbf{w}_i$ consists of more informative vectors than for a vector $\mathbf{w}_k$, if $info(\mathbf{w}_i, \mathbf{T}) > info(\mathbf{w}_k, \mathbf{T})$. The area $S(\mathbf{w}_i)$ around $\mathbf{w}_i$ represents the search space for selecting new unclassified vectors, where $S(\mathbf{w}_i)$ consists of similarity vectors $\mathbf{w} \in \mathbf{W}$ and where the similarity $sim(\mathbf{w}_i, \mathbf{w})$ is above a certain threshold that is dynamically calculated according to the current training dataset $\mathbf{T}$.

Algorithm 5: Informativeness-Aware Active Learning Approach

```
Input:
- $\mathbf{W}$: Unlabelled similarity vectors
- $b$:  Total manual labelling budget
- $k$:  Number of similarity vectors to select in each iteration
Output:
- $\mathbf{T}$:  Training dataset in the form of labelled similarity vectors
1  $\mathbf{T} \leftarrow$initialSelect ($\mathbf{W}, k$)   // Select initial training dataset
2  while $|\mathbf{T}| < b$ do
3      // Identify informative similarity vectors of the current training dataset
4      $\mathbf{I} \leftarrow$identifyInformativeVectors ($\mathbf{T}$)
5      // Select unlabelled similarity vectors around informative vectors
6      $\mathbf{W}_o \leftarrow$selectVectors ($\mathbf{I}, \mathbf{W}, k, \mathbf{T}$)
7      $\mathbf{T}' \leftarrow$manualClassify ($\mathbf{W}_o$)   // Use oracle to classify selected vectors
8      $\mathbf{T} \leftarrow \mathbf{T} \cup \mathbf{T}'$ // Add newly classified vectors to the overall training
         dataset
9      $\mathbf{W} \leftarrow \mathbf{W} \setminus \mathbf{W}_o$   // Remove classified vectors from set of unlabelled
         vectors
10 return $T$
```

7.4 Informativeness-Aware Active Learning

In this section, we describe our active learning approach beginning with a high-level description. Algorithm 5 describes our informativeness-aware active learning approach for generating a training dataset **T**. This training dataset is generated by selecting a number of similarity vectors from the set of all similarity vectors **W**, where a total budget b is available for manual labelling of selected similarity vectors. The set of all (unlabelled) vectors **W** is generated by comparing record pairs based on the set of attributes **A** and appropriate similarity functions [21]. Initially, we select a number of similarity vectors $k > 1$ from **W** based on selection strategies such as *stratified sampling* or *farthest first* (line 1).

Throughout the learning process, we identify in each iteration a set of informative vectors $\mathbf{I} \subseteq \mathbf{T}$ according to the current training dataset **T**. The vectors in **I** are used to determine a search space for selecting k new vectors from **W** that are to be labelled by the oracle in the current iteration.

To identify the set **I**, we characterise the informativeness of a vector considering its relationship to all vectors already in **T** (line 4). In particular, the informativeness $info(\mathbf{w}, \mathbf{T})$ of a vector $\mathbf{w} \in \mathbf{T}$ is calculated using an entropy-based measure considering the similarities to vectors of both the same and the other class. Moreover, $info(\mathbf{w}, \mathbf{T})$ considers the potential search space around $\mathbf{w}$ with respect to the labelled vectors from $\mathbf{T}$. We describe the calculation of informativeness for similarity vectors and their selection in Subsection 7.4.2 below.

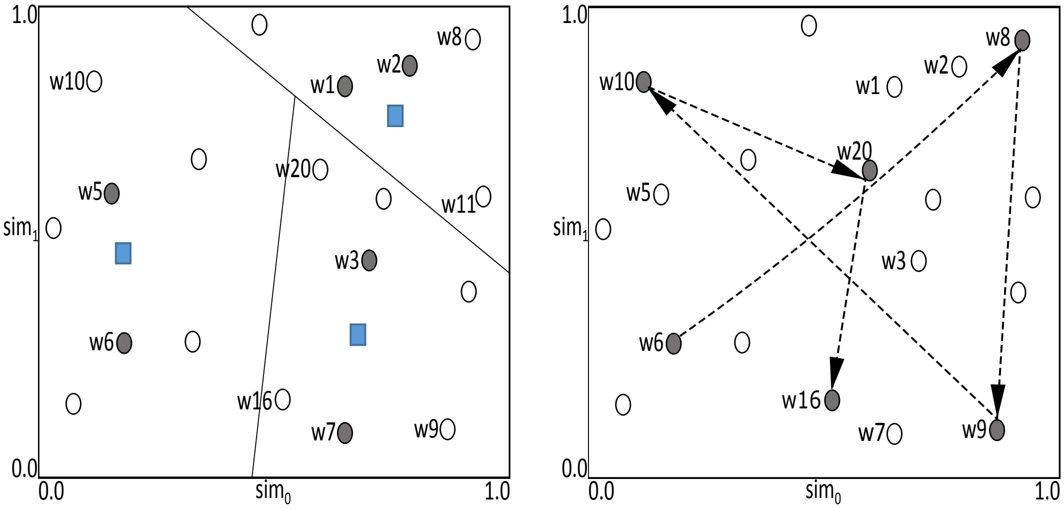
For each similarity vector in $\mathbf{I}$, we determine a search space based on its location in the similarity vector space and the location of the closest similarity vector in the opposite class as determined by the Cosine similarity. We consider each unlabelled vector contained in the search space as a candidate (line 6). The idea of the selection process is to identify similarity vectors in uncertain areas that are close to the boundary of matches and non-matches. The identified set of similarity vectors $\mathbf{W}_o$ is then manually classified by the oracle and added as $\mathbf{T}'$ to the total training dataset $\mathbf{T}$ (lines 7 and 8). The approach terminates once the number of classified similarity vectors reaches the total budget b. In the following, we describe the initial selection strategies, the computation of informativeness, and the identification of new training vectors in more detail.

7.4.1 Initial Selection

Initially, we select a set of similarity vectors from the set of all unclassified vectors $\mathbf{W}$. We propose two strategies: *stratified sampling* and *farthest first* [138].

Stratified sampling splits the set of similarity vectors $\mathbf{W}$ into several partitions $\{\mathbf{P}_1, .., \mathbf{P}_x\}$. To determine an appropriate number of partitions, x, we apply canopy clustering [86] on the unlabelled similarity vectors $\mathbf{W}$. The generated partitions are used to determine the set of k initial similarity vectors. We iteratively select similarity vectors over the x partitions, where in each iteration we select the vector $\mathbf{w}_i$ of partition $\mathbf{P}_i$ that is the closest vector to its cluster centroid, and add $\mathbf{w}_i$ to $\mathbf{T}$. After that, we remove $\mathbf{w}_i$ from partition $\mathbf{P}_i$. The process terminates once the number of selected similarity vectors is k.

On the other hand, the farthest first method [138] initially selects a similarity vector at random from $\mathbf{W}$ and adds it to $\mathbf{T}$. After that, we iteratively add another similarity vector to $\mathbf{T}$ that has the maximum distance to all vectors already in $\mathbf{T}$. We repeat this process until $\mathbf{T}$ contains k similarity vectors.



(a) Stratified sampling with $x = 3$

(b) Farthest first

Figure 7.2: Examples of initial selection strategies for $k = 6$. The grey circles represent the selected similarity vectors while squares show the centroids of each partition.

For example, in 7.2(a), stratified sampling selects the similarity vectors $w1$, $w2$, $w3$, $w5$, $w6$ and $w7$. The vector space is initially split into $x = 3$ partitions. After that, for each centroid (blue squares) of a partition we select the closest two similarity vectors. In 7.2(b), the farthest first approach randomly selects, for example, $w6$ as the first similarity vector and adds it to **T**. After that, $w8$ is selected since it is the vector farthest away from $w6$. The next selected vectors are $w9$, $w10$, $w20$, and $w16$, following the same process.

7.4.2 Informativeness of Similarity Vectors

In order to generate a representative training dataset, we propose a selection approach that considers the informativeness of similarity vectors $\mathbf{w} \in \mathbf{T}$. The goal is to determine informative classified vectors that can be used to select unclassified vectors from **W**. We describe the informativeness of a similarity vector by considering its location with respect to the vectors of the same as well as vectors from the other class in the vector space. The intuition is that we look for new vectors in the areas of classified vectors that are not outliers (i.e. are not surrounded only by vectors from the other class) but are also not easy to classify vectors (i.e. are not surrounded only by vectors from the same class).

To determine informative vectors of the current training dataset $\mathbf{T}$, we define the following measure $info(\mathbf{w}_j, \mathbf{T})$, as shown in Equation 7.1, for a classified vector $\mathbf{w}_j \in \mathbf{T}$, where sim is the Cosine similarity as described in Section 7.3. This measure is based on the entropy of a vector $\mathbf{w}_j$ according to all vectors in $\mathbf{T}$ and the uncertainty of a vector $\mathbf{w}_j$ Entropy and uncertainty are equally weighted when $\alpha = 0.5$.

$$info(\mathbf{w}_j, \mathbf{T}) = \alpha \cdot entropy(\mathbf{w}_j, \mathbf{T}) + (1 - \alpha) \cdot uncertainty(\mathbf{w}_j, \mathbf{T}) \tag{7.1}$$

Information entropy [120] can be used to describe how balanced a dataset is. In our case, the entropy of a vector $\mathbf{w}_j$ is high if it is close to vectors representing both matches as well as non matches. To determine the entropy of $\mathbf{w}_j$, we compute the aggregated similarities between $\mathbf{w}_j$ and each vector $\mathbf{w}_k$ of $\mathbf{T}_S^{w_j}$ and $\mathbf{T}_O^{w_j}$, where $\mathbf{T}_S^{w_j}$ and $\mathbf{T}_O^{w_j}$ consist of vectors that are assigned to the same class and the other class, respectively, according to $\mathbf{w}_j$, as shown in Equation 7.2.

$$\begin{aligned} entropy(\mathbf{w}_j, \mathbf{T}) \quad &= -\Big[\frac{\sum_{\mathbf{w}_k \in \mathbf{T}_S^{w_j}} sim(\mathbf{w}_j, \mathbf{w}_k)}{|\mathbf{T}|-1} \cdot log(\frac{\sum_{\mathbf{w}_k \in \mathbf{T}_S^{w_j}} sim(\mathbf{w}_j, \mathbf{w}_k)}{|\mathbf{T}|-1}) \\ &+ \frac{\sum_{\mathbf{w}_k \in \mathbf{T}_O^{w_j}} sim(\mathbf{w}_j, \mathbf{w}_k)}{|\mathbf{T}|} \cdot log(\frac{\sum_{\mathbf{w}_k \in \mathbf{T}_O^{w_j}} sim(\mathbf{w}_j, \mathbf{w}_k)}{|\mathbf{T}|})\Big] \end{aligned} \tag{7.2}$$

The uncertainty of a vector $\mathbf{w_j}$ is determined by the reciprocal of the intersection between the current training dataset $\mathbf{T}$ and the search space determined as the area between $\mathbf{w}_j$ and the closest vector of the opposite class as shown in Equation 7.3.

$$uncertainty(\mathbf{w}_j, \mathbf{T}) = \frac{1}{1 + |\mathbf{T} \cap S(\mathbf{w}_j)|} \tag{7.3}$$

For example, the entropy of $w7$ in Figure 7.3 is 0.68 calculated by Equation 7.2) utilising the aggregated similarity to vectors of the same class ($w6$ and $w5$) as $0.65 + 0.4 = 1.05$, as well as to vectors of the other class ($w1$, $w3$ and $w2$) as $0.73 + 0.91 + 0.78 = 2.42$. The intersection between the search space $S(w7)$ and the current training dataset $\mathbf{T}$ is empty and therefore $uncertainty(w7) = 1$. Con-

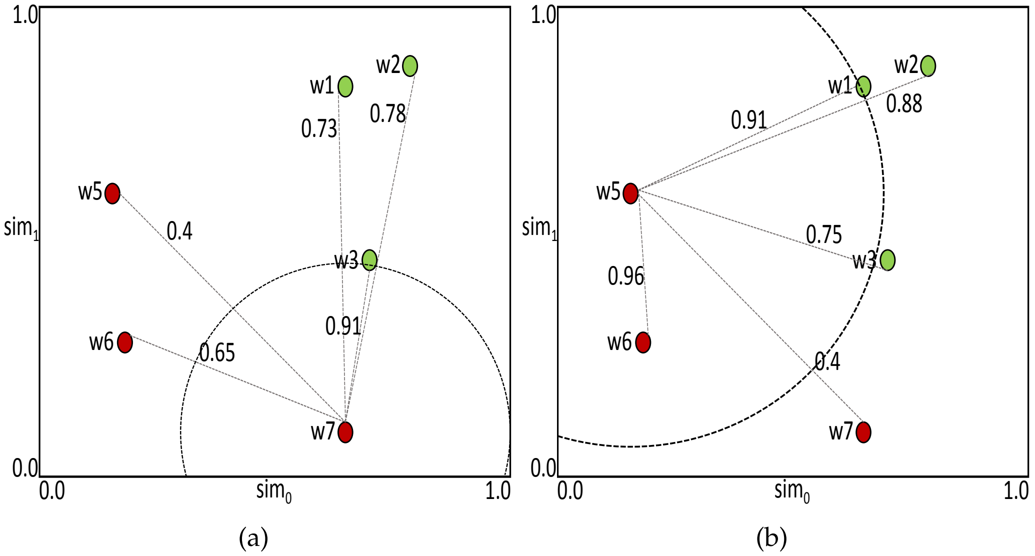


Figure 7.3: Two examples for determining the informativeness of similarity vectors $w5$ and $w7$ of $\mathbf{T}$={$w1$, $w2$, $w3$, $w5$, $w6$, $w7$}, based on the location in the vector space and the search spaces $S(w5)$ and $S(w7)$ for $w5$ and $w7$, as represented by the circles. Red coloured circles represent classified non-match similarity vectors while green coloured circles represent classified match vectors.

sequently, $info(w7)$ is equal to $0.5 \cdot 0.68 + 0.5 \cdot 1 = 0.84$. The informativeness for $w5$ is calculated similarly where its entropy is 0.697 and its uncertainty is 0.5 since $S(\mathbf{w_5}) \cap T = \{w_6\}$, and therefore $info(w5, \mathbf{T}) = 0.6$.

We add a vector $\mathbf{w}_j$ to $\mathbf{I}$ if $info(\mathbf{w}_j, \mathbf{T})$ is above the mean according to the $info$ measure for the vectors of the current training dataset $\mathbf{T}$. In our running example, the mean of $info$ according to the current training dataset is 0.61, and so we add $w7$ $(info = 0.84)$ to $\mathbf{I}$, but not $w5$. The set $\mathbf{I}$ of informative vectors is then used to select vectors of $\mathbf{W}$ to be manually classified and added to $\mathbf{T}$.

7.4.3 Training Data Selection

The selection method shown in Algorithm 6 determines for each similarity vector of $\mathbf{I}$ a set of unlabelled vectors from $\mathbf{W}$. For this, we identify for each vector $\mathbf{w}_j \in \mathbf{I}$ a search space $S(\mathbf{w}_j)$ determined by the closest vector $\mathbf{w}_c$ from the opposite class. For example, in Figure 7.4 the closest vector from the other class for $w7$ is $w3$.

The objective is to identify new vectors in uncertain areas so that in each iteration

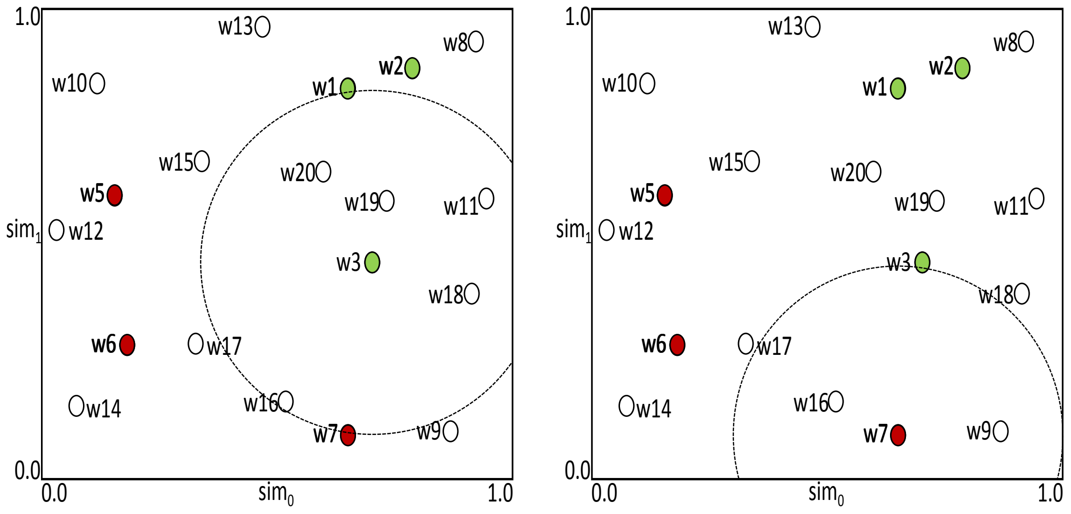
Algorithm 6: Selection Method of New Similarity Vectors

```
Input:
- $\mathbf{I}$:  Set of informative similarity vectors
- $\mathbf{T}$   Current classified training dataset
- $\mathbf{W}$: Set of unlabelled similarity vectors
- $k$:  Number of similarity vectors to be selected
Output:
- $\mathbf{W}_o$: Similarity vectors selected for manual classification by oracle
1  $\mathbf{C} = \emptyset$   // Initialise empty set of candidates
2  foreach $\mathbf{w}_j \in \mathbf{I}$ do
3      // Determine vector being closest to $w_j$ from the opposite class
4      $\mathbf{w}_c \leftarrow$ getClosest $(\mathbf{w}_j, \mathbf{T})$
5      $\delta \leftarrow sim(\mathbf{w}_j, \mathbf{w}_c)$ // Calculate threshold representing the search space of $\mathbf{w}_j$
6      foreach $\mathbf{w}_u \in \mathbf{W}$ do
7          // Add unlabelled vector if its similarity is above the threshold $\delta$
8          if $sim(\mathbf{w}_u, \mathbf{w}_j) > \delta$ then
9              $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathbf{w}_u\}$
10 // Identify the $k$ most diverse vectors from candidate set
11 $\mathbf{W}_o \leftarrow$ farthestFirstSelection $(\mathbf{C}, k)$
12 return $\mathbf{W}_o$
```

an increasingly more representative training data set $\mathbf{T}$ is generated. A vector $\mathbf{w}_u \in \mathbf{W}$ is added to the set $\mathbf{C}$ of candidates if it is contained in the search space $S(\mathbf{w}_j)$ consisting of vectors $\mathbf{w}_u$ where the similarity $sim(\mathbf{w}_j, \mathbf{w}_u)$ is larger than $sim(\mathbf{w}_j, \mathbf{w}_c)$ (line 9). At the end of the selection method, we determine the most k-diverse vectors of $\mathbf{C}$ by applying a farthest first approach (line 11).

Figure 7.4 shows an example for selecting vectors based on w_3 and w_7. The selection method selects all vectors as candidates into $\mathbf{C}$ that are in the search spaces $S(w7)$ and $S(w3)$, shown as circles around $w3$ and $w7$. Consequently, the combined candidate set, $\mathbf{C}$, based on $w7$ and $w3$ consists of the similarity vectors $w9$, $w11$, $w16$, $w18$, $w19$ and $w20$.

The identified set of similarity vectors $\mathbf{W}_o$ are then manually classified by an oracle and added to $\mathbf{T}$ (Algorithm 5, line 8). The updated training dataset is used in the next iteration to identify a new set of informative vectors. This loop ends once the number of manually classified similarity vectors reaches the budget b.



(a) (b)

Figure 7.4: Two examples of selecting new similarity vectors according to the search spaces $S(w3)$ and $S(w7)$ represented as circles, where $w3$ and $w7$ are the informative vectors. Red and green coloured circles represent classified vectors.

7.4.4 Complexity Analysis

We now briefly discuss the complexity of our proposed approach. Because of the independence of our approach with regard to the actual classification model used, its complexity only depends upon the number of unlabelled similarity vectors, $\mathbf{W}$, the total budget $\mathbf{b}$, and the number k of similarity vectors to be selected in each iteration. In each iteration, we compute the similarities between all pairs of vectors in the current training dataset, $\mathbf{T}$, resulting in a complexity of $O(|\mathbf{T}|^2)$. Moreover, we identify for each informative similarity vector of $\mathbf{I}$ the closest unlabelled similarity vectors in $\mathbf{W}$, a process which requires $|\mathbf{W}| \cdot |\mathbf{I}|$ comparisons where $|\mathbf{I}| \leq |\mathbf{T}|$ holds. At the end of each iteration, we determine the k most diverse similarity vectors of $\mathbf{C}$, where $|\mathbf{C}| \leq |\mathbf{W}|$, resulting in a complexity $O(k \cdot |\mathbf{C}|)$. Overall, the complexity to determine similarity vectors for one iteration is $O(|\mathbf{T}|^2 + |\mathbf{W}| \cdot |\mathbf{I}| + k \cdot |\mathbf{C}|)$, with $|\mathbf{I}| \leq |\mathbf{T}|$ and $|\mathbf{C}| \leq |\mathbf{W}|$. The number of iterations is bound by k and b as b/k.

Table 7.1: Overview of evaluated datasets.

dataset	Number of records	$\lvert\mathbf{W}\rvert$	Match:Non-match	Attributes	$n = \lvert\mathbf{w}\rvert$
Cora	1,295	286,141	1:16	Title, authors, year, venue	4
Google Scholar	2,616 / 64,263	472,790	1:89	Title, authors, year,venue	6
Music	19,375	251,715	1:16	Title, artist, album, year, language, number	7

7.5 Experiments and Results

We evaluated our active learning approach using three datasets as summarised in Table 7.1. The Cora and Google Scholar (GS) [68] datasets contain publication records that are to be linked, where the GS dataset consists of matches between DBLP and GS. The Music dataset contains records from the Music-Brainz database[1]. This dataset is corrupted [56] and consists of five sources with duplicates for 50% of the original records. To avoid the comparison of the full Cartesian product of vectors, we applied blocking [21] and filtering [69].

The ratios between matches and non-matches (with blocking and filtering applied) shown in Table 7.1 highlight the imbalance of these datasets and emphasize the challenges of selecting a representative training dataset. The similarity vectors (of dimension n) were calculated using string comparison functions on the different attributes shown in Table 7.1, such as q-gram based Jaccard and Soft-TF/IDF [21]. To classify the similarity vectors as matches and non-matches, we used the decision tree classifier implemented in the Weka toolkit [42].

Our proposed active learning approach is implemented in Java 1.8 and we ran all experiments on a desktop machine equipped with an Intel Core i7-4470 CPU with 8x3.40 GHz CPUs, and 32 GBytes of main memory.

We evaluated different parameter settings for our approach. As initialisation method we used *farthest first*, *stratified sampling* and *random selection*, set $\alpha = [0.3, 0.4, 0.5, 0.6, 0.7]$ to weight the *entropy* and *uncertainty* in Equation 7.1 when determining informative similarity vectors, set the number of selected vectors in each iteration as $k = [30, 35, 40, 45, 50]$, and the total budget $b = [200, 500, 1000, 2000, 5000]$. We set default values as $\alpha = 0.5$, $k = 30$, $b = 1000$ and *farthest first* as the initialisation method, because we obtained good results with these settings for all three datasets based on preliminary experiments.

[1] Available at: https://musicbrainz.org

We compared our approach with the two basic active learning approaches *Smallest Margin* [134] and *Entropy* [119], the *Uncertainty* selection approach [94], as well as the only budget limited active learning approach for ER we are aware of (named *Clu-AL*) [138]. We do not compare our approach with *MinExpError* [94] because this approach does not scale well for large budgets. Furthermore, we compared our approach with both fully supervised decision tree and support vector machine (using RBF and linear kernels) classifiers, as also used for comparison in previous work on active learning for ER [138].

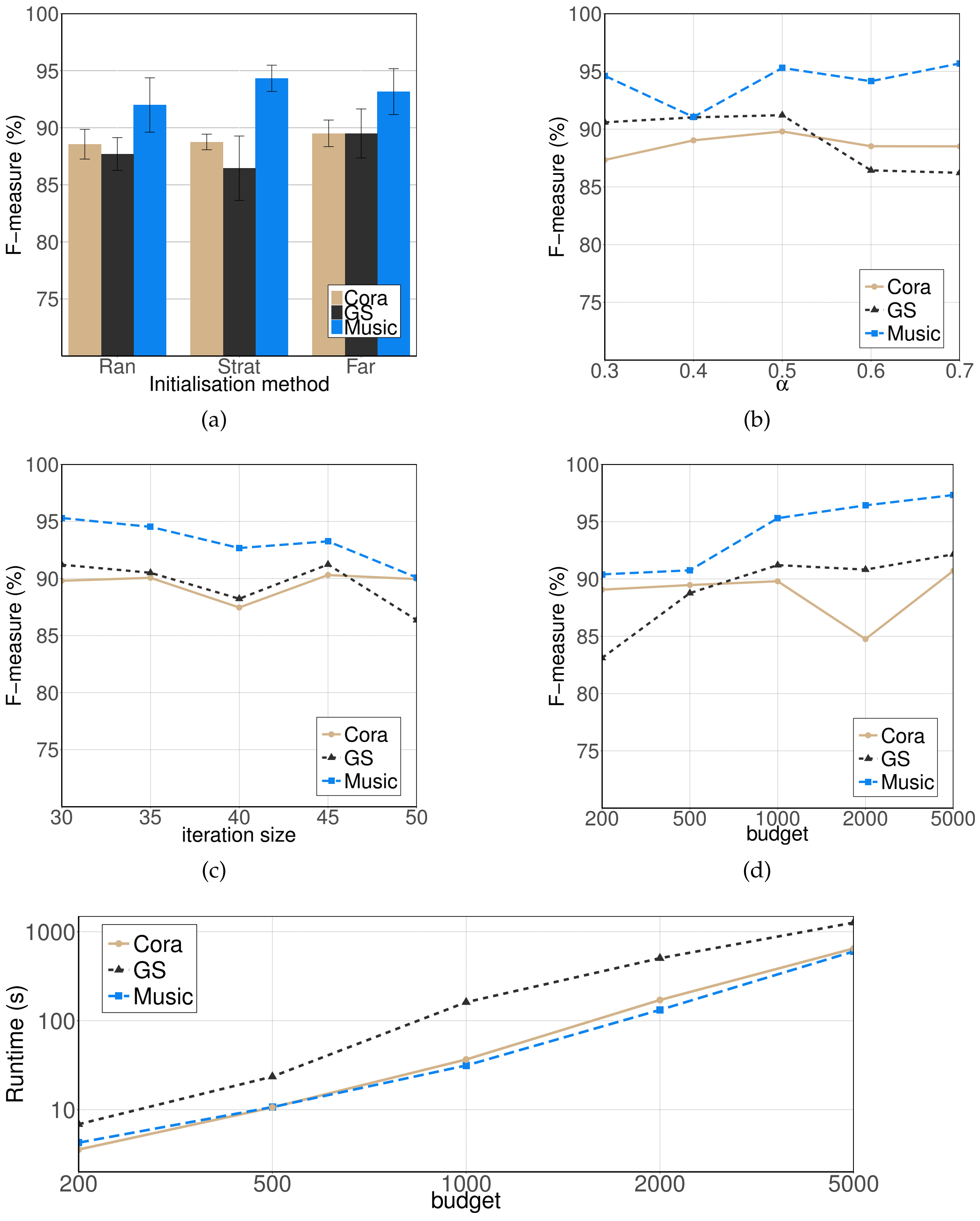
To allow a comparative evaluation of our proposed approach with these earlier approaches we use the F-measure [51]. We acknowledge that there are issues when this measure is used to comparatively evaluate different ER classifiers, however there is currently no accepted alternative to the F-measure we are aware of.

7.5.1 Parameter Evaluation

7.5(a) shows the obtained ER classification quality for different initialisation methods averaged over different iteration sizes k. Farthest first slightly outperforms stratified sampling and random selection by 0.75% and 0.95%, respectively, for the Cora dataset, and by 3.1% and 1.8% for Google Scholar. On the other hand, Farthest first achieves a lower F-Measure by 1.17% compared to stratified sampling for the Music dataset. The small differences in F-measure results for the different initial selection strategies show that our main selection strategy based on the search space of informative vectors performs effectively independent of the initial set of similarity vectors.

As can be seen in 7.5(b), changes for the weight parameter α only slightly influence the ER classification quality, between 2% to 4%, for the three datasets. For the Cora dataset we observe a decreasing quality for $\alpha > 0.5$. With an α weight over 0.5 our approach prioritises the *entropy* of a vector more than the *uncertainty*, and therefore the approach mainly selects vectors as informative that are located in-between true matches and non-matches.

For all three datasets, the F-measure slightly decreases with a higher number of selected similarity vectors, k, per iteration as shown in 7.5(c). This indicates



(a)

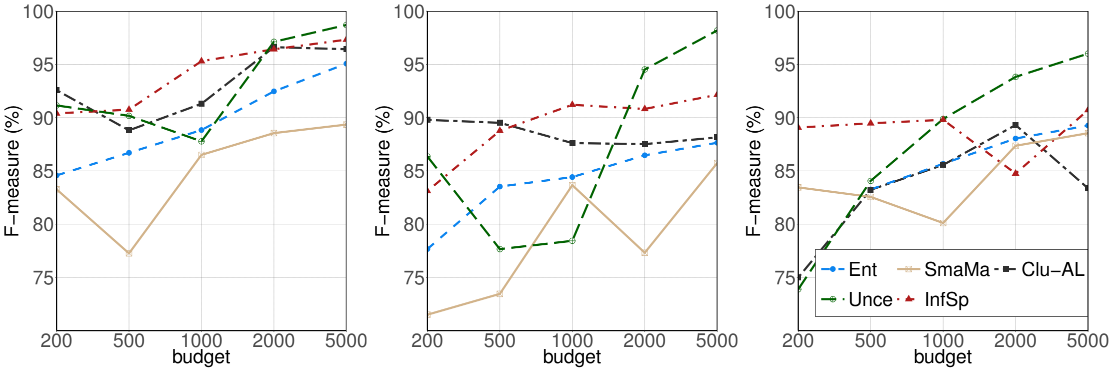
(b)

(c)

(d)

(e)

Figure 7.5: Classification F-measure results for (a) different initialisation methods, (b) different values for weight parameter α of $info$, (c) different numbers of similarity vectors per iteration k, (d) different total budgets b, and (e) runtime for different total budgets b.



(a) Music (b) Google Schoolar (c) Cora

Figure 7.6: F-measure results of our approach (named *InfoSpace-AL*, InfSp) as compared with the other active learning approaches *Entropy* (Entr) [119], *Smallest Margin* (SmaMa) [134], *Clu-AL* [138] and *Uncertainty* (Unce) [94].

that a higher number of selected similarity vectors increases the probability for selecting non-informative vectors. An increasing budget generally leads to an improvement of F-measure results as shown in 7.5(d). Even for a small budget of $b = 200$, for all three datasets our approach achieves F-measure results of above 80%, with an increase up to 97% for the Music dataset as more informative vectors are added to the training set. The runtime scales quadratically with respect to the total budget as shown in 7.5(e), however, all runtimes are below 200 seconds for budgets up to $b = 1,000$.

7.5.2 Comparison with Existing Approaches

We compare our active learning approach, named *InfoSpace-AL*, with the active learning approaches *Smallest Margin*, *Entropy*, and *Uncertainty*, as well as the clustering based active learning approach *Clu-AL* [138]. We also compare our approach with supervised approaches using fully supervised SVM and decision tree classifiers. To compare the different active learning approaches, we experimentally determined a suitable number of similarity vectors to select in each iteration, k, for each approach separately over all datasets. We use the following values for k: *Smallest Margin*: 45, *Entropy*: 50, *Uncertainty*: 45, and *InfoSpace-AL*: 30. The *Clu-AL* approach follows an adaptive strategy for determining the number of similarity vectors it selects in each iteration.

Table 7.2: F-measure results of our approach (InfoSpace-AL) as compared with fully supervised classifiers (SVM and DTree) for a budget of $b = 1,000$.

dataset	Dtree	SVM	**InfoSpace-AL**
Google Scholar	88.63%	91.44%	91.21%
Cora	84.09%	82.22%	89.80%
Music	96.80%	96.90%	95.30%

Figure 7.6 shows the F-Measure of the considered approaches according to different budgets b. *InfoSpace-AL* is the only approach that, for a small budget, achieves an F-Measure above 80% for all three datasets. *Smallest Margin* and *Uncertainty* result in a high variance with an increasing budget, where the F-Measure achieved by *Uncertainty* is reduced by up to 8.7% from a budget of $b = 200$ to $b = 500$. In contrast, *InfoSpace-AL* achieves more stable F-Measure results compared to *Uncertainty* even for small budgets of $200 \leq b \leq 1,000$. *InfoSpace-AL* and *Clu-AL* both achieve high F-Measure results for each dataset for small budgets of $b = 500$ and $b = 1,000$. However, we observe that *Uncertainty* achieves high F-Measure values above 90% for each dataset if the budget is above $b = 2,000$. To summarise, our approach achieves results comparable to *Clu-AL* and *Uncertainty*, and it is one of the best performing approaches for small budgets of up-to $b = 1,000$.

To evaluate the two supervised approaches, we applied 10-fold cross validation. Our approach achieves comparable results compared to the fully supervised approaches as shown in Table 7.2. Our informativeness-based active learning approach outperforms the supervised approaches by around 5.7% in F-Measure for the Cora dataset. On the other hand, the supervised approaches achieve higher F-Measure results for the Google Scholar and Music datasets compared to our active learning approach. However, we emphasize that our approach achieves these comparable results with a moderate manual classification effort, so that the labelling effort is reduced by around 99% compared to a fully supervised classifier that requires much larger training datasets which are commonly not available in real-world ER applications.

7.6 Conclusion

We have proposed an active learning approach for entity resolution (ER) that iteratively selects similarity vectors into a training dataset based on the informativeness of vectors for a current training dataset. Unlike with existing active learning approaches for ER, the main advantage of our approach is that it is independent of any intermediate classification results since it determines the search space for new vectors based on a defined informativeness measure considering the location of vectors in the vector space, as well as the uncertainty of the search space. In each iteration, our approach selects new vectors according to the most informative vectors. The evaluation showed that our approach can achieve results comparable to fully supervised approaches where much larger training datasets are required to achieve a high ER quality compared to our budget limited approach. Moreover, our approach outperforms a previous state-of-art active learning method for ER that is also based on a limited budget for the number of manual classifications possible. Furthermore, our approach does also not rely on the assumption of monotonicity of precision [138].

For future work we aim to investigate adaptive methods for determining an optimal number k of selected similarity vectors in each iteration such that the probability for selecting non-informative similarity vectors is minimised. We also plan to investigate filtering methods that initially reduce the set of vectors $\mathbf{W}$ to avoid the selection of non-informative vectors. Moreover, we like to integrate metric space approaches to improve the efficiency of the approach for determining new unlabelled similarity vectors.

Part IV

Conclusion and Outlook

8 Conclusion and Outlook

8.1 Conclusion

This dissertation focuses on approaches for improving data integration tasks in the medical domain and other domains where entity resolution is needed. The introductory discussion showed the importance of annotations as well as entity resolution but also mentioned the challenges being not addressed by current methods so far. The second part proposed different methods for improving the annotation process for medical forms. The third part focused on techniques for improving entity resolution for graph structure and temporal data as well as the linkage quality using machine learning.

8.1.1 Entity Linking of medical documents

The current research concentrates little on annotating medical forms, especially case report forms being essential for examining clinical trials. The majority of approaches utilize dictionary-based techniques with exact string matches. To overcome data quality problems such as typos or unknown mentions that not occur in the dictionary, we developed AnnoMap. This tool provides a set of similarity functions and different options for combining them to determine appropriate candidates for document fragments. To finally select the annotations for a document fragment, the group selection was developed. This strategy selects the best candidate for a document fragment that has multiple ones. Nevertheless, the search space is enormous to link document fragments to the corresponding concepts due to the size of ontologies. Moreover, each concept is described by several synonyms and names so that it is challenging to identify the relevant synonyms for the annotation process. The developed approach for reusing annotations reduces the number of comparisons and increases the quality of annotation mappings. The approach generates compact representatives for each concept that is linked to already annotated document fragments. The annotation process utilizes the generated representatives to link the not annotated documents. The approach extends the group selection utilizing the graph structure from the ontology and the co-occurring concepts from the annotated documents. The graph structure is utilized to compute graph based measurements for each annotation candidate. The evaluation showed that the reuse of annotations and the graph-based selection strategy improved the annotation results compared to the basic annotation process and MetaMap.

In addition to the reuse of annotated documents, the results of various annotation tools can also be reused. The method considers the identified annotations from each method as well as the calculated confidence values. The combination of results used a machine-learning approach, where each annotation is characterized by a vector containing the computed confidence values. The evaluation showed that the results from MetaMap and cTAKES could be improved by the machine learning-based combination approach.

8.1.2 Techniques for improving Entity resolution results

Besides the enrichment of data, entity resolution is an essential task to enable data analysis. Especially when analyzing census data, methods need to be able to identify personal records over different periods that represent the same person.

A further feature of historical census data is the information in which household a person lived. This information can be used to build a graph for each household where the edges are between persons from the same one. However, there are only a few methods that make use of these graphs. The developed approach uses the graphs to reduce the search space of possible matches. The core idea is to determine similar subgraphs so that the search for matches for persons is limited to corresponding households.

Nevertheless, the quality of the result depends highly on the pre-matching step since the next steps utilize the resulting matches to identify the same subgraphs. The pre-matching uses a manual defined similarity function so that the effort for the definition is time-consuming, and the result is error pruned. Machine learning techniques can determine classification models based on training data to improve this step. The developed approach aims to reduce the number of classified links and to generate sounded classifiers. In detail, the method utilized the vector space of similarity vectors to determine informative ones that are selected for creating a model. The identification of an informative vector distinguishes the approach from another one that uses intermediate classification results. The evaluation showed that the developed approach outperforms previous budget limited approaches and achieves similar results like supervised methods with less labeling effort.

8.2 Outlook

This dissertation focused on improving annotation processes for medical forms, and entity resolution methods represent an essential contribution to data integration efforts in the life sciences as well as building knowledge graphs for generic domains. The increasing amount of data in the medical area, such as case report forms for clinical trials, requires an integrated view to effectively interlink results from different clinical studies and reuse existing forms for creating new ones. Further approaches can incorporate the developed methods and concepts, such as a reuse repository for documents, annotations, annotation clusters, domains, and tools with quality result metadata.

Moreover, the increasing number of various data sources restricts comprehensive information retrieval and data analysis. Methods that provide strategies to link records from different sources are essential to create integrated knowledge bases, such as knowledge graphs. The developed methods contribute to existing approaches to improve the results. Nevertheless, new research directions can be integrated, such as graph embeddings as well as node embeddings for the developed group linkage method. In the next section, we describe the possibilities for further development and improvement.

Reuse Repository for Annotation Managment

The reuse approach proposed in Chapter 4 can be extended by maintaining the domains of documents, the used tools, results, and quality metadata as well as topic depending classification models. The idea is to utilize specific annotation clusters, tools, and classifiers for certain domains. An extension would be the identification of topics for a document. For instance, case report forms and electronic health records about heart diseases are probably annotated with the same concepts. Depending on the determined topics, the appropriate tools can be determined based on the stored tool results for the same or similar domain. The idea is to select tools that are known to perform well for the identified field. The chosen tools generate the annotations considering the area. If the reuse repository consists of sufficient representative annotations, they can be used for creating classification models like in Chapter 5. Moreover, the approach to determine informative vectors (Chapter 7) can be integrated to reduce the number of annotations to build a model.

Besides the improvement of the annotation process, the retrieval of annotated documents is essential to create new forms or to integrate existing results. New approaches can utilize the annotated documents to generate knowledge graphs based on annotations so that two documents are connected if they share joint annotations. Moreover, each document can be represented as a graph if the annotations occur close to each other. The graph structure allows the usage of graph query languages, such as Cypher, so that patterns can be queried. For instance, a CRF for the clinical study consists of annotations about a particular disease and the tested drug. Furthermore, an electronic health record of a patient is annotated with the diagnosis and specific illness. A retrieval approach can query that subgraph by specifying a graph pattern with a disease, drug, and diagnosis concepts as well as the types of documents to retrieve the clinical study.

The extended repository combines the different aspects of domain-specific annotations, annotation clusters, tools, and classifiers so that a method can derive an appropriate configuration for processing an unannotated document.

Moreover, a graph structure would enable complex graph queries to get specific results.

Mulitlingual Annotation processes

The majority of datasets for annotation or entity linking benchmarks are in English so that approaches perform well. Nevertheless, only a few methods can handle non-English documents shown in a recent study [80]. Hence, novel approaches can focus on solving multilingual documents. For instance, a method translates all documents to Englisch and annotates the translations. However, the study in [80] observed that the annotation quality based on the translated documents decreases. Further directions are the usage of hidden representations using neural networks that embed concept and text representation in a shared vector space for different languages. The goal is that for instance, an English ontology is linkable to a German document.

Parallelization of the Annotation process

This thesis mainly focused on the quality of annotations. Nevertheless, the growing amount of documents requires efficient solutions that can process thousands of documents in a short time. A solution could be the parallelization of the proposed methods using parallel frameworks like Flink or Spark. These frameworks are built on predefined functions that allow the definition of workflows. The idea is to represent each proposed step with these functions to process multiple documents in parallel.

Temporal Analysis

The developed approach to link census data considers temporal and graph-structured aspects. Current research [116] shows that temporal analysis for graphs supports the understanding of evolving graphs. However, the study of graphs at different times requires the linking of records. The developed approach identifies new and deleted nodes by their non-existence in a match result. This requires more sophisticated procedures that can determine, for example, whether a node is new or removed based on the given features. Furthermore, edges representing not a "same as" relationship, such as Move, can also be inferred using the node features, edge features, and temporal aspects. The temporal features can be used to compute trajectories with a probability for attribute value changes of a record.

Neural Network-based Collective Entity Resolution

Current approaches utilize the graph structure as context information. On the assumption that two nodes represent the same if the neighborhoods are similar. For instance, two publications are probably the same if the authors are the same. However, the final result depends on the order of selected links for computing the neighborhood similarity. A wrong chosen link leads to a wrong neighborhood similarity and hence to a malicious link. Node embeddings provide such information, to consider attribute similarity and neighborhood similarity at the same time. Recently, a lot of work offers techniques to generate node embeddings using random walks [106, 47], or Graph Convolutional Neural networks [62]. New approaches can combine traditional collective entity resolution approaches with embedding approaches to compute more meaningful similarities. Moreover, graph embeddings can be used to represent subgraphs, such as households like in the method proposed in Chapter 6.

The listed opportunities for further work show the potential for improvement by novel methods. Our contributions for annotating medical forms as well as linking temporal and graph structured data can be used in the appropriate domain and can be extended with further machine learning techniques.

Bibliography

[1] TIES-Text Information Extraction System, 2017.

[2] ABEDI, V., ZAND, R., YEASIN, M., FAISAL, F. An automated framework for hypotheses generation using literature. *BioData Min. 5* (2012), 13.

[3] ARASU, A., GÖTZ, M., KAUSHIK, R. On active learning of record matching packages. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010* (2010), A. K. Elmagarmid and D. Agrawal, Eds., ACM, pp. 783–794.

[4] ARONSON, A. R., LANG, F. An overview of metamap: historical perspective and recent advances. *J. Am. Medical Informatics Assoc. 17*, 3 (2010), 229–236.

[5] AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, R., IVES, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web* (Berlin, Heidelberg, 2007), K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds., Springer Berlin Heidelberg, pp. 722–735.

[6] BELLARE, K., IYENGAR, S., PARAMESWARAN, A. G., RASTOGI, V. Active sampling for entity matching. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012* (2012), Q. Yang, D. Agarwal, and J. Pei, Eds., ACM, pp. 1131–1139.

[7] BENDER, O., OCH, F. J., NEY, H. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* (2003), pp. 148–151.

[8] Bhattacharya, I., Getoor, L. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data 1*, 1 (2007), 5.

[9] Bodenreider, O. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research 32*, Database-Issue (2004), 267–270.

[10] Bramesfeld A, W. G. Cross-Sectoral Quality Assurance. *Social Code Book V. Public Health Forum* (2014), 14.e1–14.e3.

[11] Breil, B., Kenneweg, J., Fritz, F., et al. Multilingual medical data models in ODM format–a novel form-based approach to semantic interoperability between routine health-care and clinical research. *Appl Clin Inf 3* (2012), 276–289.

[12] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[13] Bunescu, R., Pasca, M. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy* (April 2006), pp. 9–16.

[14] Campos, D., Matos, S., Lewin, I., Oliveira, J. L., Rebholz-Schuhmann, D. Harmonization of gene/protein annotations: towards a gold standard medline. *Bioinformatics 28*, 9 (2012), 1253–1261.

[15] Campos, D., Matos, S., Oliveira, J. Biomedical named entity recognition: A survey of machine-learning tools. In *Theory and Applications for Advanced Text Mining*, S. Sakurai, Ed. InTech, Rijeka, 2012, ch. 08.

[16] Campos, D., Matos, S., Oliveira, J. Current methodologies for biomedical named entity recognition. *Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data* (2013), 839–868.

[17] Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., Trani, S. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (New York, NY, USA, 2013), CIKM '13, Association for Computing Machinery, p. 139–148.

[18] CHIANG, Y., DOAN, A., NAUGHTON, J. F. Modeling entity evolution for temporal record matching. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014* (2014), C. E. Dyreson, F. Li, and M. T. Özsu, Eds., ACM, pp. 1175–1186.

[19] CHIANG, Y., DOAN, A., NAUGHTON, J. F. Tracking entities in the dynamic world: A fast algorithm for matching temporal records. *Proc. VLDB Endow. 7*, 6 (2014), 469–480.

[20] CHRISTEN, P. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008* (2008), Y. Li, B. Liu, and S. Sarawagi, Eds., ACM, pp. 151–159.

[21] CHRISTEN, P. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Springer, 2012.

[22] CHRISTEN, P. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Trans. Knowl. Data Eng. 24*, 9 (2012), 1537–1555.

[23] CHRISTEN, P., GAYLER, R. W. Adaptive temporal entity resolution on dynamic databases. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II* (2013), J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds., vol. 7819 of *Lecture Notes in Computer Science*, Springer, pp. 558–569.

[24] CHRISTEN, V., CHRISTEN, P., RAHM, E. Informativeness-based active learning for entity resolution. In *Machine Learning and Knowledge Discovery in Databases* (Cham, 3 2020), P. Cellier and K. Driessens, Eds., Springer International Publishing, pp. 125–141.

[25] CHRISTEN, V., GROSS, A., FISHER, J., WANG, Q., CHRISTEN, P., RAHM, E. Temporal group linkage and evolution analysis for census data. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017* (3 2017), V. Markl, S. Orlando, B. Mitschang, P. Andritsos, K. Sattler, and S. Breß, Eds., OpenProceedings.org, pp. 620–631.

[26] CHRISTEN, V., GROSS, A., RAHM, E. A reuse-based annotation approach for medical documents. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I* (2016), P. T. Groth, E. Simperl, A. J. G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck, and Y. Gil, Eds., vol. 9981 of *Lecture Notes in Computer Science*, pp. 135–150.

[27] CHRISTEN, V., GROSS, A., VARGHESE, J., DUGAS, M., RAHM, E. Annotating medical forms using UMLS. In *Data Integration in the Life Sciences - 11th International Conference, DILS 2015, Los Angeles, CA, USA, July 9-10, 2015, Proceedings* (2015), N. Ashish and J. L. Ambite, Eds., vol. 9162 of *Lecture Notes in Computer Science*, Springer, pp. 55–69.

[28] CHRISTEN, V., LIN, Y., GROSS, A., CARDOSO, S. D., PRUSKI, C., SILVEIRA, M. D., RAHM, E. A learning-based approach to combine medical annotation results. In *Data Integration in the Life Sciences - 13th International Conference, DILS 2018, Hannover, Germany, November 20-21, 2018, Proceedings* (11 2018), S. Auer and M. Vidal, Eds., vol. 11371 of *Lecture Notes in Computer Science*, Springer, pp. 135–143.

[29] CUCERZAN, S. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic* (2007), J. Eisner, Ed., ACL, pp. 708–716.

[30] DAI, M., ET AL. An efficient solution for mapping free text to ontology terms. *AMIA Summit on Translational Bioinformatics 21* (2008).

[31] DASGUPTA, S. Two faces of active learning. *Theor. Comput. Sci. 412*, 19 (2011), 1767–1781.

[32] DONG, X. L., KEMENTSIETSIDIS, A., TAN, W.-C. A time machine for information: Looking back to look forward. *SIGMOD Rec. 45*, 2 (Sept. 2016), 23–32.

[33] DONNELLY, K. SNOMED-CT: The Advanced Terminology and Coding System for eHealth. *Studies in Health Technology and Informatics–Medical and Care Compunetics 3 121* (2006), 279–290.

[34] DREDZE, M., MCNAMEE, P., RAO, D., GERBER, A., FININ, T. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (Beijing, China, Aug. 2010), Coling 2010 Organizing Committee, pp. 277–285.

[35] DUGAS, M. Missing Semantic Annotation in Databases. The Root Cause for Data Integration and Migration Problems in Information Systems. *Methods of Information in Medicine 53*, 6 (2014), 516–517.

[36] DUGAS, M., FRITZ, F., KRUMM, R., BREIL, B. Automated UMLS-based comparison of medical forms. *PloS one 8*, 7 (2013).

[37] DUGAS, M., NEUHAUS, P., MEIDT, A., DOODS, J., STORCK, M., BRULAND, P., VARGHESE, J. Portal of medical data models: information infrastructure for medical research and healthcare. *Database 2016* (2016).

[38] ELMAGARMID, A. K., IPEIROTIS, P. G., VERYKIOS, V. S. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng. 19*, 1 (2007), 1–16.

[39] ERTEKIN, S., HUANG, J., BOTTOU, L., GILES, C. L. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007* (2007), M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, **O**. H. Olsen, and A. O. Falcão, Eds., ACM, pp. 127–136.

[40] EUZENAT, J., SHVAIKO, P. *Ontology matching*. Springer, 2007.

[41] FELLEGI, I. P., SUNTER, A. B. A theory for record linkage. *Journal of the American Statistical Association 64*, 328 (1969), 1183–1210.

[42] FRANK, E., HALL, M. A., HOLMES, G., KIRKBY, R., PFAHRINGER, B., WITTEN, I. H., TRIGG, L. Weka-a machine learning workbench for data mining. In *Data Mining and Knowledge Discovery Handbook, 2nd ed*, O. Maimon and L. Rokach, Eds. Springer, 2010, pp. 1269–1277.

[43] FRANKE, M., SEHILI, Z., GLADBACH, M., RAHM, E. Post-processing methods for high quality privacy-preserving record linkage. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2018 International Workshops, DPM 2018 and CBT 2018, Barcelona, Spain, September 6-7, 2018, Proceedings* (2018), J. García-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, Eds., vol. 11025 of *Lecture Notes in Computer Science*, Springer, pp. 263–278.

[44] FU, Z., CHRISTEN, P., BOOT, M. Automatic cleaning and linking of historical census data using household information. In *Workshop on Domain Driven Data Mining, held at IEEE ICDM* (Vancouver, 2011).

[45] FU, Z., CHRISTEN, P., ZHOU, J. A graph matching method for historical census household linkage. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I* (2014), V. S. Tseng, T. B. Ho, Z. Zhou, A. L. P. Chen, and H. Kao, Eds., vol. 8443 of *Lecture Notes in Computer Science*, Springer, pp. 485–496.

[46] GOKHALE, C., DAS, S., DOAN, A., NAUGHTON, J. F., RAMPALLI, N., SHAVLIK, J. W., ZHU, X. Corleone: hands-off crowdsourcing for entity matching. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014* (2014), C. E. Dyreson, F. Li, and M. T. Özsu, Eds., ACM, pp. 601–612.

[47] GROVER, A., LESKOVEC, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016* (2016), B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds., ACM, pp. 855–864.

[48] GRUBER, T. R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation* (Deventer, The Netherlands, 1993), N. Guarino and R. Poli, Eds., Kluwer Academic Publishers.

[49] GUO, S., CHANG, M.-W., KICIMAN, E. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Atlanta, Georgia, June 2013), Association for Computational Linguistics, pp. 1020–1030.

[50] HAN, X., SUN, L., ZHAO, J. Collective entity linking in web text: a graph-based method. In *SIGIR '11* (2011).

[51] HAND, D., CHRISTEN, P. A note on using the f-measure for evaluating record linkage algorithms. *Stat. Comput. 28*, 3 (2018), 539–547.

[52] HAO, T., RUSANOV, A., BOLAND, M. R., WENG, C. Clustering clinical trials with similar eligibility criteria features. *J. Biomed. Informatics 52* (2014), 112–120.

[53] HARTUNG, M., GROSS, A., RAHM, E. Conto-diff: generation of complex evolution mappings for life science ontologies. *J. Biomed. Informatics 46*, 1 (2013), 15–32.

[54] HARTUNG, M., TERWILLIGER, J. F., RAHM, E. Recent advances in schema and ontology evolution. In *Schema Matching and Mapping*, Z. Bellahsene, A. Bonifati, and E. Rahm, Eds., Data-Centric Systems and Applications. Springer, 2011, pp. 149–190.

[55] HERZOG, T. N., SCHEUREN, F. J., WINKLER, W. E. *Data quality and record linkage techniques*. Springer, 2007.

[56] HILDEBRANDT, K., PANSE, F., WILCKE, N., RITTER, N. Large-scale data pollution with apache spark. *IEEE Transactions on Big Data* (2017).

[57] HOFFART, J., YOSEF, M. A., BORDINO, I., FÜRSTENAU, H., PINKAL, M., SPANIOL, M., TANEVA, B., THATER, S., WEIKUM, G. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, UK., July 2011), Association for Computational Linguistics, pp. 782–792.

[58] HUMPHREY, S. M., ROGERS, W. J., KILICOGLU, H., DEMNER-FUSHMAN, D., RINDFLESCH, T. C. Word sense disambiguation by selecting the best semantic type based on Journal Descriptor Indexing: Preliminary experiment. *Journal of the American Society for Information Science and Technology 57*, 1 (2006), 96–113.

[59] HUNTLEY, R. P., SAWFORD, T., MUTOWO-MEULLENET, P., SHYPITSYNA, A., BONILLA, C., MARTIN, M. J., O'DONOVAN, C. The GOA database: Gene ontology annotation updates for 2015. *Nucleic Acids Research 43*, Database-Issue (2015), 1057–1063.

[60] KALASHNIKOV, D. V., MEHROTRA, S. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Trans. Database Syst. 31*, 2 (2006), 716–767.

[61] KALAYDJIEVA, L., GRESHAM, D., CALAFELL, F. Genetic studies of the roma (gypsies): A review. *BMC Medical Genetics 2* (02 2001).

[62] KIPF, T. N., WELLING, M. Semi-supervised classification with graph convolutional networks. *CoRR abs/1609.02907* (2016).

[63] KIRSTEN, T., GROSS, A., HARTUNG, M., RAHM, E. GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *J. Biomedical Semantics 2* (2011), 6.

[64] KOLB, L., THOR, A., RAHM, E. Load balancing for mapreduce-based entity resolution. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012* (2012), A. Kementsietsidis and M. A. V. Salles, Eds., IEEE Computer Society, pp. 618–629.

[65] KOLB, L., THOR, A., RAHM, E. Multi-pass sorted neighborhood blocking with mapreduce. *Computer Science-Research and Development 27*, 1 (2012), 45–63.

[66] KÖPCKE, H., RAHM, E. Training selection for tuning entity matching. In *Proceedings of the International Workshop on Quality in Databases and Management of Uncertain Data, Auckland, New Zealand, August 2008* (2008), P. Missier, X. Lin, A. de Keijzer, and M. van Keulen, Eds., pp. 3–12.

[67] KÖPCKE, H., RAHM, E. Frameworks for entity matching: A comparison. *Data Knowl. Eng. 69*, 2 (2010), 197–210.

[68] KÖPCKE, H., THOR, A., RAHM, E. Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow. 3*, 1 (2010), 484–493.

[69] KÖPCKE, H., THOR, A., RAHM, E. Learning-based approaches for matching web data entities. *IEEE Internet Comput. 14*, 4 (2010), 23–31.

[70] KOUKI, P., PUJARA, J., MARCUM, C., KOEHLY, L., GETOOR, L. Collective entity resolution in multi-relational familial networks. *Knowledge and Information Systems 61*, 3 (2019), 1547–1581.

[71] KULKARNI, S., SINGH, A., RAMAKRISHNAN, G., CHAKRABARTI, S. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009* (2009), J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, Eds., ACM, pp. 457–466.

[72] KUM, H., KRISHNAMURTHY, A. K., MACHANAVAJJHALA, A., AHALT, S. C. Social genome: Putting big data to work for population informatics. *IEEE Computer 47*, 1 (2014), 56–63.

[73] LACOSTE-JULIEN, S., PALLA, K., DAVIES, A., KASNECI, G., GRAEPEL, T., GHAHRAMANI, Z. Sigma: simple greedy matching for aligning large knowledge bases. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013* (2013), I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthurusamy, Eds., ACM, pp. 572–580.

[74] LASSILA, O., MCGUINNESS, D. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science 6*, 5 (2001), 2001.

[75] LEPENDU, P., IYER, S., FAIRON, C., SHAH, N. H. Annotation analysis for testing drug safety signals using unstructured clinical notes. *J. Biomedical Semantics 3*, S-1 (2012), S5.

[76] LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (1966), vol. 10, pp. 707–710.

[77] LI, F., LEE, M., HSU, W., TAN, W. Linking temporal records for profiling entities. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015* (2015), T. K. Sellis, S. B. Davidson, and Z. G. Ives, Eds., ACM, pp. 593–605.

[78] LI, J., CARDIE, C. Timeline generation: tracking individuals on twitter. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014* (2014), C. Chung, A. Z. Broder, K. Shim, and T. Suel, Eds., ACM, pp. 643–652.

[79] LI, P., DONG, X. L., MAURINO, A., SRIVASTAVA, D. Linking temporal records. *Proc. VLDB Endow. 4*, 11 (2011), 956–967.

[80] LIN, Y., CHRISTEN, V., GROSS., A., KIRSTEN., T., CARDOSO., S. D., PRUSKI., C., SILVEIRA., M. D., RAHM., E. Evaluating cross-lingual semantic annotation for medical forms. In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5 HEALTHINF: HEALTHINF,* (2 2020), INSTICC, SciTePress, pp. 145–155.

[81] LIN, Y.-C., CHRISTEN, V., GROSS, A., CARDOSO, S. D., PRUSKI, C., DA SILVEIRA, M., RAHM, E. *Evaluating and Improving Annotation Tools for Medical Forms*. Springer International Publishing, Cham, 11 2017, pp. 1–16.

[82] LINGREN, T., DELÉGER, L., MOLNÁR, K., ZHAI, H., MEINZEN-DERR, J., KAISER, M., STOUTENBOROUGH, L., LI, Q., SOLTI, I. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *J. Am. Medical Informatics Assoc. 21*, 3 (2014), 406–413.

[83] LOWE, H. J., BARNETT, G. O. Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches. *Journal of the American Medical Association (JAMA) 271*, 14 (1994), 1103–1108.

[84] LUO, Z., DUFFY, R., JOHNSON, S., WENG, C. Corpus-based approach to creating a semantic lexicon for clinical research eligibility criteria from UMLS. *AMIA Summits on Translational Science Proceedings 2010* (2010), 26.

[85] MANNING, C. D., RAGHAVAN, P., SCHÜTZE, H. *Introduction to information retrieval*. Cambridge University Press, 2008.

[86] MCCALLUM, A., NIGAM, K., UNGAR, L. H. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000* (2000), R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, Eds., ACM, pp. 169–178.

[87] MCCRAY, A. T., SRINIVASAN, S., BROWNE, A. C. Lexical methods for managing variation in biomedical terminologies. In *Proceedings of the Annual Symposium on Computer Application in Medical Care* (1994), American Medical Informatics Association, p. 235.

[88] MIHALCEA, R., CSOMAI, A. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007* (2007), M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, **O**. H. Olsen, and A. O. Falcão, Eds., ACM, pp. 233–242.

[89] MILIAN, K., HOEKSTRA, R., BUCUR, A. I. D., TEN TEIJE, A., VAN HARMELEN, F., PAULISSEN, J. Enhancing reuse of structured eligibility criteria and supporting their relaxation. *J. Biomed. Informatics 56* (2015), 205–219.

[90] MILNE, D., WITTEN, I. H. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (New York, NY, USA, 2008), CIKM '08, Association for Computing Machinery, p. 509–518.

[91] MOCERI, V. M., KUKULL, W. A., EMANUAL, I., VAN BELLE, G., STARR, J. R., SCHELLENBERG, G. D., MCCORMICK, W. C., BOWEN, J. D., TERI, L., LARSON, E. B. Using census data and birth certificates to reconstruct the early-life socioeconomic environment and the relation to the development of alzheimer's disease. *Epidemiology 12*, 4 (2001), 383–389.

[92] MONAHAN, S., LEHMANN, J., NYBERG, T., PLYMALE, J., JUNG, A. Cross-lingual cross-document coreference with entity linking. In *TAC* (2011), NIST.

[93] Morwal, S., Jahan, N., Chopra, D. Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) 1*, 4 (2012), 15–23.

[94] Mozafari, B., Sarkar, P., Franklin, M. J., Jordan, M. I., Madden, S. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proc. VLDB Endow. 8*, 2 (2014), 125–136.

[95] Nadeau, D., Sekine, S. A survey of named entity recognition and classification. *Linguisticae Investigationes 30*, 1 (January 2007), 3–26. Publisher: John Benjamins Publishing Company.

[96] Naumann, F., Herschel, M. *An Introduction to Duplicate Detection*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[97] Nentwig, M., Gross, A., Rahm, E. Holistic entity clustering for linked data. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* (Dec 2016), pp. 194–201.

[98] Nentwig, M., Hartung, M., Ngomo, A. N., Rahm, E. A survey of current link discovery frameworks. *Semantic Web 8*, 3 (2017), 419–436.

[99] Ngomo, A. N., Lyko, K. EAGLE: efficient active learning of link specifications using genetic programming. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings* (2012), E. Simperl, P. Cimiano, A. Polleres, Ó. Corcho, and V. Presutti, Eds., vol. 7295 of *Lecture Notes in Computer Science*, Springer, pp. 149–163.

[100] Ngonga Ngomo, A.-C., Sherif, M. A., Lyko, K. Unsupervised link discovery through knowledge base repair. In *The Semantic Web: Trends and Challenges* (Cham, 2014), V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, and A. Tordai, Eds., Springer International Publishing, pp. 380–394.

[101] NOY, N. F., MUSEN, M. A. PROMPTDIFF: A fixed-point algorithm for comparing ontology versions. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada* (2002), R. Dechter, M. J. Kearns, and R. S. Sutton, Eds., AAAI Press / The MIT Press, pp. 744–750.

[102] NUAIMI, K. A., MOHAMED, N., NUAIMI, M. A., AL-JAROODI, J. A survey of load balancing in cloud computing: Challenges and algorithms. In *Second Symposium on Network Cloud Computing and Applications, NCCA 2012, London, United Kingdom, December 3-4, 2012* (2012), IEEE Computer Society, pp. 137–142.

[103] ODELL, M., RUSSELL, R. The soundex coding system. *US Patents 1261167* (1918).

[104] OGREN, P. V., SAVOVA, G. K., CHUTE, C. G. Constructing evaluation corpora for automated clinical named entity recognition. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco* (2008), European Language Resources Association.

[105] PAPADAKIS, G., KOUTRIKA, G., PALPANAS, T., NEJDL, W. Meta-blocking: Taking entity resolutionto the next level. *IEEE Transactions on Knowledge and Data Engineering 26*, 8 (Aug 2014), 1946–1960.

[106] PEROZZI, B., AL-RFOU, R., SKIENA, S. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014* (2014), S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, Eds., ACM, pp. 701–710.

[107] PESQUITA, C., FARIA, D., FALCÃO, A. O., LORD, P. W., COUTO, F. M. Semantic similarity in biomedical ontologies. *PLoS Computational Biology 5*, 7 (2009).

[108] RAHM, E. Towards large-scale schema and ontology matching. In *Schema Matching and Mapping*, Z. Bellahsene, A. Bonifati, and E. Rahm, Eds., Data-Centric Systems and Applications. Springer, 2011, pp. 3–27.

[109] RAHM, E., DO, H. H. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull. 23*, 4 (2000), 3–13.

[110] RAMADAN, B., CHRISTEN, P., LIANG, H. Dynamic sorted neighborhood indexing for real-time entity resolution. In *Databases Theory and Applications - 25th Australasian Database Conference, ADC 2014, Brisbane, QLD, Australia, July 14-16, 2014. Proceedings* (2014), H. Wang and M. A. Sharaf, Eds., vol. 8506 of *Lecture Notes in Computer Science*, Springer, pp. 1–12.

[111] RASTOGI, V., DALVI, N. N., GAROFALAKIS, M. N. Large-scale collective entity matching. *Proc. VLDB Endow. 4*, 4 (2011), 208–218.

[112] RATINOV, L., ROTH, D., DOWNEY, D., ANDERSON, M. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (USA, 2011), HLT '11, Association for Computational Linguistics, p. 1375–1384.

[113] REN, K., LAI, A. M., MUKHOPADHYAY, A., ET AL. Effectively processing medical term queries on the UMLS Metathesaurus by layered dynamic programming. *BMC Medical Genomics 7*, Suppl 1 (2014).

[114] RICHARDS, L., ANTONIE, L., AREIBI, S., GREWAL, G. W., INWOOD, K., ROSS, J. A. Comparing classifiers in historical census linkage. In *2014 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2014, Shenzhen, China, December 14, 2014* (2014), Z. Zhou, W. Wang, R. Kumar, H. Toivonen, J. Pei, J. Z. Huang, and X. Wu, Eds., IEEE Computer Society, pp. 1086–1094.

[115] ROBERTS, A., GAIZAUSKAS, R. J., HEPPLE, M., DEMETRIOU, G., GUO, Y., ROBERTS, I., SETZER, A. Building a semantically annotated corpus of clinical texts. *J. Biomed. Informatics 42*, 5 (2009), 950–966.

[116] ROST, C., THOR, A., RAHM, E. Temporal graph analysis using gradoop. In *BTW 2019 – Workshopband* (2019), H. Meyer, N. Ritter, A. Thor, D. Nicklas, A. Heuer, and M. Klettke, Eds., Gesellschaft für Informatik, Bonn, pp. 109–118.

[117] SAEEDI, A., PEUKERT, E., RAHM, E. Using link features for entity clustering in knowledge graphs. In *The Semantic Web* (Cham, 2018), A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., Springer International Publishing, pp. 576–592.

[118] SAVOVA, G. K., MASANZ, J. J., OGREN, P. V., ZHENG, J., SOHN, S., KIPPER-SCHULER, K. C., CHUTE, C. G. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association 17*, 5 (2010), 507–513.

[119] SETTLES, B. Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[120] SHANNON, C. E. A mathematical theory of communication. *Bell Syst. Tech. J. 27*, 4 (1948), 623–656.

[121] SHEN, W., WANG, J., HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng. 27*, 2 (2015), 443–460.

[122] SHEN, W., WANG, J., LUO, P., WANG, M. Linden: linking named entities with knowledge base via semantic knowledge. In *WWW* (2012).

[123] SHERIF, M. A., NGOMO, A. N., LEHMANN, J. Wombat - A generalization approach for automatic link discovery. In *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I* (2017), E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, Eds., vol. 10249 of *Lecture Notes in Computer Science*, pp. 103–119.

[124] SIMONINI, G., BERGAMASCHI, S., JAGADISH, H. V. Blast: A loosely schema-aware meta-blocking approach for entity resolution. *Proc. VLDB Endow. 9*, 12 (Aug. 2016), 1173–1184.

[125] SINGH, R., MEDURI, V. V., ELMAGARMID, A. K., MADDEN, S., PAPOTTI, P., QUIANÉ-RUIZ, J., SOLAR-LEZAMA, A., TANG, N. Synthesizing entity matching rules by examples. *Proc. VLDB Endow. 11*, 2 (2017), 189–202.

[126] SOHN, S., KOCHER, J.-P. A., CHUTE, C. G., SAVOVA, G. K. Drug side effect extraction from clinical narratives of psychiatry and psychology patients. *Journal of the American Medical Informatics Association 18*, Supplement 1 (2011), i144–i149.

[127] SOHN, S., SAVOVA, G. K. Mayo clinic smoking status classification system: extensions and improvements. *AMIA Annual Symposium Proceedings* (2009), 619–623.

[128] STEORTS, R. C., VENTURA, S. L., SADINLE, M., FIENBERG, S. E. A comparison of blocking methods for record linkage. In *Privacy in Statistical Databases* (Cham, 2014), J. Domingo-Ferrer, Ed., Springer International Publishing, pp. 253–268.

[129] STOJANOVIC, L., MAEDCHE, A., MOTIK, B., STOJANOVIC, N. User-driven ontology evolution management. In *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings* (2002), A. Gómez-Pérez and V. R. Benjamins, Eds., vol. 2473 of *Lecture Notes in Computer Science*, Springer, pp. 285–300.

[130] SUCHANEK, F. M., KASNECI, G., WEIKUM, G. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web* (New York, NY, USA, 2007), WWW '07, ACM, pp. 697–706.

[131] TANENBLATT, M. A., CODEN, A., SOMINSKY, I. L. The conceptmapper approach to named entity recognition. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta* (2010), N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds., European Language Resources Association.

[132] THOR, A., RAHM, E. MOMA - A mapping-based object matching system. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings* (2007), www.cidrdb.org, pp. 247–258.

[133] TODOROVIC, B. T., RANCIC, S. R., MARKOVIC, I. M., MULALIC, E. H., ILIC, V. M. Named entity recognition and classification using context hidden markov model. In *2008 9th Symposium on Neural Network Applications in Electrical Engineering* (Sep. 2008), pp. 43–46.

[134] TSAI, M.-H., HO, C.-H., LIN, C.-J. Active learning strategies using SVMs. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (Barcelona, 2010), IEEE, pp. 1–8.

[135] TSEYTLIN, E., MITCHELL, K. J., LEGOWSKI, E., CORRIGAN, J., CHAVAN, G., JACOBSON, R. S. NOBLE - flexible concept recognition for large-scale biomedical natural language processing. *BMC Bioinform. 17* (2016), 32.

[136] VARGHESE, J., DUGAS, M. Frequency Analysis of Medical Concepts in Clinical Trials and their Coverage in MeSH and SNOMED-CT. *Methods of Information in Medicine 53*, 6 (2014).

[137] VERITY, D., MARR, J., OHNO, S., WALLACE, G., STANFORD, M. Behçet's disease, the silk road and hla-b51: historical and geographical perspectives. *Tissue Antigens 54*, 3 (1999), 213–220.

[138] WANG, Q., VATSALAN, D., CHRISTEN, P. Efficient interactive training selection for large-scale entity resolution. In *PAKDD* (Vietnam, 2015).

[139] WANG, S., XIAO, X., LEE, C.-H. Crowd-based deduplication: An adaptive approach. In *ACM SIGMOD* (Melbourne, 2015), pp. 1263–1277.

[140] WANG, Y., ZHU, M., QU, L., SPANIOL, M., WEIKUM, G. Timely YAGO: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings* (2010), I. Manolescu, S. Spaccapietra, J. Teubner, M. Kitsuregawa, A. Léger, F. Naumann, A. Ailamaki, and F. Özcan, Eds., vol. 426 of *ACM International Conference Proceeding Series*, ACM, pp. 697–700.

[141] WEIKUM, G., NTARMOS, N., SPANIOL, M., TRIANTAFILLOU, P., BENCZÚR, A. A., KIRKPATRICK, S., RIGAUX, P., WILLIAMSON, M. Longitudinal analytics on web archive data: It's about time! In *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings* (2011), www.cidrdb.org, pp. 199–202.

[142] WHETZEL, P. L., NOY, N. F., SHAH, N. H., ALEXANDER, P. R., NYULAS, C., TUDORACHE, T., MUSEN, M. A. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Research 39*, Web-Server-Issue (2011), 541–545.

[143] WILKINSON, M. D., DUMONTIER, M., AALBERSBERG, I. J., APPLETON, G., AXTON, M., BAAK, A., BLOMBERG, N., BOITEN, J.-W., DA SILVA SANTOS, L. B., BOURNE, P. E., ET AL. The fair guiding principles for scientific data management and stewardship. *Scientific Data 3* (3 2016), 160018–.

[144] ZHANG, W., TAN, C. L., SIM, Y. C., SU, J. NUS-I2R: learning a combined system for entity linking. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010* (2010), NIST.

[145] ZHENG, J., CHAPMAN, W. W., MILLER, T. A., LIN, C., CROWLEY, R. S., SAVOVA, G. K. A system for coreference resolution for the clinical narrative. *Journal of the American Medical Informatics Association 19*, 4 (2012), 660.

[146] ZOU, Q., ET AL. Indexfinder: a knowledge-based method for indexing clinical texts. *Proc. AMIA Annual Symp.* (2003), 763–767.

[147] ZWICKLBAUER, S., SEIFERT, C., GRANITZER, M. From general to specialized domain: Analyzing three crucial problems of biomedical entity disambiguation. In *Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part I* (2015), Q. Chen, A. Hameurlain, F. Toumani, R. R. Wagner, and H. Decker, Eds., vol. 9261 of *Lecture Notes in Computer Science*, Springer, pp. 76–93.