Knowledge Discovery Considering Domain Literature and Ontologies : Application to Rare Diseases

Mohsen Hassan

▶ **To cite this version:**

Mohsen Hassan. Knowledge Discovery Considering Domain Literature and Ontologies : Application to Rare Diseases. Computation and Language [cs.CL]. Université de Lorraine, 2017. English. NNT : 2017LORR0092 . tel-01678860v2

HAL Id: tel-01678860

https://theses.hal.science/tel-01678860v2

Submitted on 18 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Knowledge Discovery Considering Domain Literature and Ontologies: Application to Rare Diseases

THÈSE

présentée et soutenue publiquement le 11 Juillet 2017

pour l'obtention du

Doctorat de l'Université de Lorraine

(Spécialite informatique)

par

Mohsen Hassan

Composition du jury

Président :	François CHAROY	Professeur, Université de Lorraine
Rapporteurs :	Mathieu ROCHE	Chercheur, HDR, CIRAD
	Nathalie PERNELLE	Maître de conférences, HDR, Université Paris-Sud
Examinateurs :	Christel VRAIN	Professeur, Université d'Orléans
	Marie-Christine JAULENT	Directeur de recherche, INSERM
	Peggy CELLIER	Maître de conférences, INSA Rennes
Directeur :	Yannick TOUSSAINT	Professeur, Université of Lorraine
Co-Directeur :	Adrien COULET	Maître de conférences, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Acknowledgement

I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully and completing this thesis.

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Yannick TOUSSAINT and Dr. Adrien COULET for the continuous support of my Ph.D. work and for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I appreciate a lot their care with every point in this thesis work. This lets me learn a lot from them and increase my skills. I could not have imagined having better supervisors and mentors for my Ph.D. study.

Besides my supervisors, I would like to thank Christel VRAIN, François CHAROY, Marie-Christine JAULENT, Mathieu ROCHE, Nathalie PERNELLE and Peggy CELLIER who kindly accepted being members of my defense committee. I would like to thank them alot for their detailed report, encouragement, insightful comments, questions and valuable suggestions.

I want to thank all members of Orpailleur team for their friendly, positive, and helpful attitude. I thank Amedeo Napoli and Chedy Raïsi for sharing their scientific experience.

I would like to thank my family, my parents and my parents-in-law for supporting me throughout my life. Special thanks to my wife Rania who has been a constant source of love, support and encouragement during the challenges of my PhD and life. Special thanks to my mother and my father who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve. My father learned me a lot until the last moment in his life. He left our world two days before my PhD defense. It was hard times for me. But, his last call made me insist to do a good defense as this may make him happy in his new world. Our current world does not only take our lovely relatives but sometimes it gives us the future. During my PhD journey, my two sons Ahmed and Mohamed were born. They encouraged me a lot to make them proud of their father.

I would like also to thank Mr. Bernard. He is my French teacher who teaches our family the French language once we stayed at Nancy-France and I can't forget to thank all French people too. Their smiles during their work and their kind help especially for the foreigners make you feel that you are not far from your home.

This work would not have been possible without the financial support of two ANR grants: Hybrid and PractiKPharma. Finally, I would like to thank Inria and University of Lorraine for the hospitality and the continuous support.

To those I love, especially to my parents, parents-in-law, my wife, my sons and my brothers.

Contents

Introduction

1 Data and Knowledge Resources for RD and their Limitations

1.1 Rare Diseases (RD) and Orphanet

A Rare Disease (RD), also known as an orphan disease, is by definition a disease that affects a small percentage of the population. A disease is considered as rare when it affects less than 1 in 2,000 persons. Even if RDs are rare, their number is large: There are between 6,000 and 8,000 known RDs. In addition, they are often chronic and life-threatening. Their cumulative number and severity are two reasons that make their study important. But several factors are making their study difficult: (1) the lack of access to correct diagnosis; (2) the lack of information about the disease online or in public databases; (3) the lack of scientific knowledge about the disease. These problems lead to misdiagnosis or delay in the diagnosis; and providing comprehensive information of good quality and extracting knowledge about RDs from available resources could help at facilitating diagnosis and developing therapeutic procedures for RDs.

RDs are characterized by a wide diversity of phenotypes that vary not only from one disease to another but also from one patient to another suffering from the same disease. The extraction of information about the phenotypes of RD is of particular importance since it provides a fine-grained description of diseases, which could be used to guide medical diagnosis and clinical care. Medical diagnosis is the process of identifying which disease explains a patient's symptoms and signs. Classically, symptoms and clinical signs are observable phenomenon that arises from and accompanies a particular disease and serves as an indication of it. Usually, a symptom is subjective, observed by the patient, and cannot be measured directly, whereas a clinical sign is objectively observable. For example, "A light headache" is a symptom because it is only ever detected by the patient, while "High blood sugar" is a sign because it is measured in a medical laboratory. Because the complexity of RD, it is hard to know precisely what feature is a cause or an indicator of the disease. This is why in the case of RD we propose to consider phenotypes, a generalization of symptoms and signs, which are observable phenomenon broadly associated with the disease.

Orphanet is a web portal that is an institutional source of information related to RDs, mainly funded by the INSERM. The Orphanet encyclopedia is a part of it that provides natural language summaries for many RDs. These summaries serve as international references about the definition of RD. They contain various kinds of medical information about the RDs such as the clinical description, etiology, epidemiology, diagnostic methods, disease management and treatments. In addition, they provide a list of the phenotypes associated with the RD. An example of Orphanet summary is provided in Appendix A. Orphanet summaries are written and updated manually by experts of each RD in collaboration with the Orphanet team. This manual process is based on an extensive and manual review of literature, consequently tedious and time consuming. As the knowledge may evolve quickly in the field, even high quality summaries may no longer be up-to-date rapidly. This is what is observed when some summaries are compared with the most recent literature or compared with Orphadata, a freely-accessible dataset related to RDs and part of Orphanet, which update rate and process are different and more reactive to novelties.

It is consequently tempting to develop data mining approaches on available resources such as Orphadata and the literature to propose completing RD summaries.

1.2 Knowledge Discovery from Databases (KDD) and Texts (KDT)

Knowledge Discovery from Databases (KDD) is the process of discovering useful knowledge from data. Fayyad *et al.* [FPSS96] defined it as follows: "KDD focuses on the overall process of knowledge discovery from data, including how the data are stored and accessed, how algorithms can be scaled to massive data sets, how results can be interpreted and visualized, and how the overall man-machine interaction can usefully be modeled and supported". Indeed, KDD is an iterative and interactive process. It is iterative because the results of each KDD step may be used to refine previous steps. It is interactive because it is controlled by a domain expert that guides the extraction process.

The KDD process may be seen as a process turning data into information and then knowledge. Before introducing the various KDD steps, we first propose to clarify the distinction between Data, Information and Knowledge as defined in [SA00]. *Data* are raw facts and uninterpreted signals that have no meaning. They can be any alphanumeric characters such as text, numbers and symbols. Given the following sentence as an example, "DMD is characterized by muscle weakness", it can be considered as data, consisting of the strings of characters and other symbols that form this sentence. *Information* is data equipped with meaning. In other words, information is data that has been processed into a form that provides some meaning. For example, "DMD" is not just a set of characters, it may be interpreted by a reader as a disease that could affect a person. *Knowledge* is the understanding of data, information and rules that is acquired through experience or learning. It is used to perform tasks and infer new information. For instance, considering the following sentence "BMD is characterized by muscle weakness", we could infer that the two diseases "DMD" and "BMD" have the same phenotype "muscle weakness" and thus we could expect some common treatments for both of them.

Figure 1 illustrates the KDD process as defined by Fayyad *et al.* in 1996. The steps of the KDD process are:

Data selection: It is the step during which one selects the appropriate datasets for mining. This step depends on the kind of knowledge one desires to extract. This selection step may involve domain experts and can be done manually or automatically by retrieving related data from large repositories.

Preprocessing: The preprocessing step aims at cleaning the collected data obtained from the selection step. It is a particularly useful when dealing with noise or missing values in the dataset. **Transformation**: The transformation step aims at preparing the dataset in a format, adapted to the desired data mining method. In other words, in this step one selects data and syntax and expresses the data in terms of this syntax. This step may include data reduction and projection in order to highlight useful features that may represent the data. **Data Mining** This step uses prepared data to discover previously unknown or interesting patterns. This is the "core step" of the KDD process. Different data mining algorithms are available depending on the kind of considered data and the goal of the KDD process.

Interpretation/Evaluation In this step, one desires to evaluate the information extracted by the mining step. Experts in a domain may be involved in this process. Also, visualization techniques can be used to represent the discovered patterns to help experts in understanding and interpreting the mining results. Indeed, data mining results may be large and not obvious to interpret.

The results of a KDD process rely on the available data. Databases could be incomplete, and this is particularly the case in databases manually curated such as Orphanet. In our case Orphanet is incomplete in comparison to what exist in scientific literature.

In many domains such as the biomedical research, the literature is a major source of information. It contains a very large amount of information, which may be novel and useful in comparison to what is available in structured or semi-structured databases such as Orphadata and Orphanet. Unfortunately the size of the literature is frequently too large to be fully considered manually [LvI10] and an automatic approach is desired to consider exhaustively

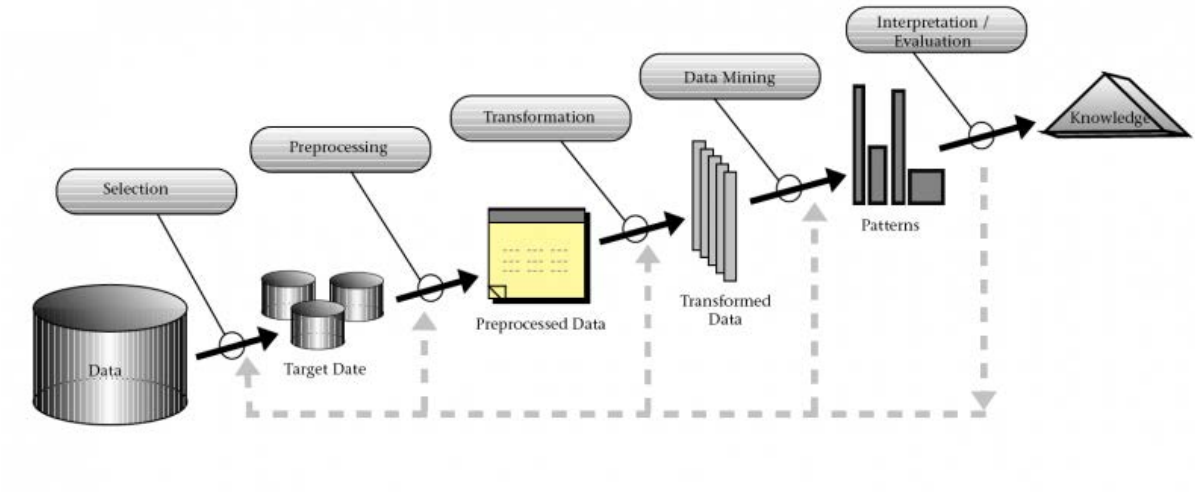


Figure 1: An overview of the steps that compose the KDD process [FPSS96].

this information. Therefore, it is of interest to use text mining approaches for completing these databases.

Similarly to data mining, text mining can be seen as the application of algorithms and methods from the fields of machine learning and statistics to texts with the goal of finding useful patterns. For this purpose it is necessary to pre-process the texts accordingly. Information extraction methods, Natural Language Processing (NLP) techniques and other pre-processing steps are required in order to prepare texts. Then data mining algorithms can be applied to prepared texts to extract patterns and knowledge.

Similarly to KDD, we define knowledge discovery in text (KDT) as a KDD applied to textual data when the core step of data mining is replaced by text mining. Text mining is similar to data mining, but the used data sources are different. Data Mining uses any kind of data whereas text mining uses only unstructured or semi-structured documents, *i.e.,* text-based documents. Because of the particularity of natural language used in text-based documents, specific preparation steps and adapted mining algorithms may be applied to them.

Discovering knowledge from textual documents is not a straightforward task. KDT uses NLP techniques for dealing with irregular and implicitly structured representation of texts. NLP aims at transforming this kind of representations into structured representation. For example, this can be achieved by representing texts by a set of features, called feature-based representation. Indeed, texts may be represented at different levels: words, a bag of words, sequences of words, syntactic trees, graphs such as dependency graphs; and they may be enriched by some linguistic features: part of speech, syntactic or semantic features. Then, an appropriate mining algorithm may be applied to this novel representation for knowledge discovery.

1.3 Classifications of Diseases and RD

Classifications are useful in health care because they help in the processing, the management, the integration, the visualization and the retrieval of the medical information. Diseases may be classified by etiology (cause), pathogenesis (mechanism by which the disease is caused), or by phenotype(s). Diseases can be also classified according to the organ system involved, which is complicated since many diseases affect more than one organ. A classical classification of human diseases is the World Health Organization's ICD (International Classification of Diseases). It is considered as the oldest and a classical classification of diseases [ICD07]. Other existing

classifications such as Human Disease Ontology (DO), Orphanet Rare Disease ontology (ORDO) and Orphanet classification are presented in the next paragraphs.

International Classification of Diseases (ICD) The ICD is designed as a health care classification system, providing a system of diagnostic codes for classifying diseases, including a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease [ICD07]. It is used for statistical purposes and as a coding system in medical databases. One of the limitations of the ICD is that it does not help recognizing the outcome of the course of treatment [KPM+08]. Also, the final diagnosis that is coded by ICD may require several visits of the patient and rarely accomplished during the first visit.

Human Disease Ontology (DO) DO is an ontology for describing the classification of human diseases organized from a clinical perspective of disease etiology and location [KAF+15]. DO was developed to create a single structure for the classification of diseases to unify the representation of diseases into an ontology from the many and varied existing terminologies and vocabularies. DO aims at providing a clear definition for each disease. This enables its consistent use and application for annotating biomedical data. DO is integrated with other terminological resources such as MeSH, ICD, NCI's thesaurus, SNOMED and OMIM.

Orphanet Rare Disease Ontology (ORDO) ORDO provides a structured vocabulary for RDs capturing relationships between RDs, genes and other relevant features. It could be a useful resource for the computational analysis of RDs. ORDO is developed from the Orphanet database [Orp15b]. It integrates a nosology (*i.e.,* a classification of RDs), data (*e.g.,* gene-disease relationships) and connections with other terminologies (MeSH, SNOMED CT, UMLS, MedDRA, ICD10) and databases (OMIM, UniProtKB, HGNC, ensembl, Reactome, IUPHAR, Geantlas).

Orphanet Classifications In addition to ORDO, Orphanet offers 33 classifications of RDs in hierarchical representations each focusing on one organ system (*e.g.,* Rare cardiac diseases, Rare neurological diseases) [Orp15b]. A RD in Orphanet, depending on their clinical presentation, may be included in several classifications. These classifications contain only the diagnostic criteria that are reviewed by experts in the medical domain and published in the scientific literature.

All these resources of disease classifications are useful resources aiming to provide disease information to health-care professionals in order to contribute improving the diagnosis and the treatment of these diseases. Although these classifications are of high-quality thanks to the involvement of experts curating their content, they are not replacing to a health care professional. These classifications are not exhaustive and they present distinct views and aspects of diseases. In addition, the continuous update of these resources is a nontrivial and tedious task which consumes a lot of efforts and time. The availability of an automatic method for classifying diseases based on their phenotypic description could help this process.

2 Objectives of the Thesis

In the previous section we presented some issues related to RDs. These issues motivate the focus and the objectives of this thesis. Here we list these issues as follows:

- The available databases that list D-P relationships (*e.g.,* Orphanet, Orphadata) are incomplete and not up-to-date in comparison with what exists in the biomedical literature.
- The manual update of these databases is a hard and consuming task.

- Extracting information from literature is not a straightforward task.
- Classifications of RDs are incomplete and do not consider the phenotypic description of RD.

3 Contributions

In this thesis, we contribute to the domain of KDT and knowledge representation applied to RD by proposing novel methods and applications.

3.1 On the Methodology Level

First contribution: We proposed a hybrid approach, named SPARE*, for extracting D-P relationships from text. This approach combines two main components: *(1)* a pattern based method and *(2)* a machine learning method. The pattern-based method, called SPARE (without the *), aims at learning syntactic patterns and then applying them to identify relationships from text. Different parameters have been associated with learned patterns to identify high quality ones, adequate for relationship extraction. The machine learning method is based on Support Vector Machines (SVM). SVM uses sets of linguistic features extracted from sentences that contain the two interesting entities and checks the existence of a relationship between these two entities. Then, SVM is combined with SPARE to improve the results of the relationship extraction task. Indeed, different machine learning methods (*e.g.,* rule-based methods, decision tree methods, Naïve Bayes, SVM) have been experimented to finally choose the more suitable for our task.

Second contribution: we proposed an approach for identifying new entities from texts that are unrecognized previously by a Named Entity Recognition (NER) tool. To achieve this, we reused, but relaxed, our syntactic patterns that are generated from SPARE for identifying entity candidates from text. To validate the correctness of the candidates and assess their novelty, we compared candidates to concepts existing ontologies. The comparison between candidates and existing ontologies is achieved by using a compositional semantics space and mapping rules for finding the most appropriate mapping.

Third contribution: We employed the pattern structures, which are an extension of Formal Concept Analysis, for proposing an ontology enrichment method. Here, pattern structures enable to respect and consider an existing ontology, while classifying objects in regards to their description in a database. Then, we generated a lattice that contains a very large number of concepts that are candidates to enrich our initial ontology. Because we can not propose every concept to experts, we provided several methods for selecting most interesting concepts among all those of the lattice.

3.2 On the Application Level

- We applied SPARE* for:
  1. the extraction of D-P relationships from the biomedical literature, where diseases and phenotypes are identified in biomedical articles by a NER tool;
  2. the identification of RD phenotypes that are not identified by a NER tool.
- SPARE* has been used for suggesting completions to the Orphanet encyclopedia and Orphadata. For this, we both extracted D-P relationships and identified unreferenced phenotypes from PubMed abstracts. Then, we mapped the result of one extraction to what exists in Orphanet and Orphadata.

- Finally, we applied our ontology enrichment method to a RD ontology using the phenotypic descriptions of diseases provided by Orphadata as a source of enrichment. The resulting lattice provides new RD classes, from which we selected the most interesting ones to suggest new classes to add to the ontologies.

4 Thesis Layout

The thesis is structured as follows:

Chapter 1: Biomedical Resources about Rare Diseases and Phenotypes. This chapter presents some data and knowledge resources about RDs and phenotypes. It particularly focuses on those we use through this thesis such as Orphanet, Orphadata, OMIM, the Human Phenotype Ontology (HPO) and UMLS. Then, it presents different text corpora that could be used for knowledge discovery from text in this domain. Finally, it discusses the roles and limits of the resources presented.

Chapter 2: Data Mining and Classification Approaches. This chapter gives an overview of data mining and classification approaches. First, we introduce different pattern mining techniques such as itemset mining, association rule, sequence and graph mining. Second, we propose a non-exhaustive survey on classification approaches, including symbolic and numerical approaches. Finally, we provide details on Formal Concept Analysis and Pattern Structures, two symbolic classification approaches.

Chapter 3: Knowledge Discovery from Text and its Biomedical Applications. This chapter introduces the process of Knowledge Discovery from Text (KDT) and its related applications in the biomedical domain. It particularly presents the use of NLP for text mining. It also presents how text can be differently represented and the characteristics of each representation. We introduce some useful applications of text mining such as information extraction and ontology construction. Then, we propose a focus on the Relationship Extraction task and its applications in the biomedical domain. Finally, we propose a recall on the main evaluation metrics used in text mining.

Chapter 4: A Hybrid Method for Disease-Phenotype Relationship Extraction. This chapter presents the problem of extracting D-P relationships from biomedical texts. Then, it presents the original method we proposed, called SPARE*. This method is hybrid because it combines both a pattern-based method and a machine learning method. The pattern-based method, called SPARE (standing for Syntactic PAttern for Relationship Extraction), learns syntactic patterns using the shortest paths between a disease and a phenotype in dependency graphs of sentences. These syntactic patterns are then used to extract new D-P relationships from biomedical texts. Several experiments over biomedical texts have been conducted to configure SPARE* in order to be efficient in terms of $F\text{-}measure$.

Chapter 5: Recognizing Complex Phenotypes Using Syntactic Patterns. The chapter 5 presents our approach for recognizing new entities using syntactic patterns. This approach relies on the relaxation of patterns learned with SPARE*. To validate the correctness of discovered phenotypes and assess their novelty, they are compared with phenotypes listed in HPO. This comparison uses a mapping approach that relies on a compositional semantic space model and a set of manually defined mapping rules. The results show the feasibility of our approach for discovering new phenotypes that were not already referenced in phenotype databases and ontologies and may involve complex phenotype descriptions. We present here an application of SPARE* that consists in mining the scientific literature for completing the RD phenotype description proposed in 2 reference databases: Orphanet and Orphadata.

Chapter 6: Using Text Mining and Pattern Structures for Classification and Ontology Enrichment. In this chapter, we use the KDT results and pattern structures for providing a new RD classification based on their phenotypic descriptions and for enriching an existing RD ontology by suggesting new and potentially interesting (*i.e,* if they are useful in RD diagnosis) RD classes. The lattice generated by pattern structures provides a new RD classification based on both the initial ontology and their phenotypic descriptions. This classification may suggest

new RD classes (*i.e.,* concepts of the lattice) to enrich the original RD classification. Because their number is large, we propose different methods to select a reduced set of concepts.

Chapter 7: Conclusion and Perspectives. This chapter concludes the document with a summary of my contributions and a perspective of this work.

1

Biomedical Resources about Rare Diseases and Phenotypes

Contents

This chapter presents public resources about RDs and phenotypes. These resources could be used for extracting information and could help in extracting knowledge about the biomedical domain. Section 1.1 presents the databases and ontologies of RDs and phenotypes. Section 1.2 presents MEDLINE, which is a source of publications in the biomedical domain, and presents some available annotated corpora that contain disease or phenotype annotations. Finally, section 1.3 concludes and discusses the role of these resources in this thesis.

1.1 Data and Knowledge Resources

This section presents the databases and ontologies that list RDs, phenotypes and Disease-Phenotype (D-P) relationships.

1.1.1 Orphanet

Orphanet [Orp15b] is a web portal providing information about orphan drugs and RDs. It is accessible online in seven languages (English, French, German, Italian, Portuguese, Spanish and Dutch). It supplies information about RDs, including their phenotypes, genes and orphan drugs (*i.e.,* drugs for RDs). Orphanet provides a diagnosis assistance service that enables retrieving information about diseases by searching clinical signs using a controlled vocabulary. At present Orphanet is partly filled manually and describes over 3,000 RDs.

The Orphanet encyclopedia, which is a part of Orphanet, provides natural language summaries for these RDs. Orphanet summaries are written and updated manually by experts of the domain in collaboration with the Orphanet team. This manual process results from a regular and manual review of the related literature and is a tedious and time consuming task. Figure A.2 of Appendix A shows an example of Orphanet summary for the Kennedy disease. This summary is available in the Orphanet website [Orp15b] by searching for the disease name "Kennedy disease" or by searching with its Orpha number 481. This summary is structured into different sections, which may or may not be present: *(1)* disease definition, *(2)* epidemiology, *(3)* clinical description, *(4)* etiology, *(5)* diagnostic methods, *(6)* differential diagnosis, *(7)* antenatal diagnosis, *(8)* genetic counseling, *(9)* management and treatment and *(10)* prognosis. Orphanet summaries contain usually a list of phenotypes associated with the disease. For instance, the summary of Kennedy disease mentions phenotypes such as *"proximal and bulbar muscle wasting", "wasting of the limb"*, *"bulbar muscles"*, *"dysarthria"*, *"dysphonia"*, *"hanging jaw"*, *"tongue wasting"*, *"chewing difficulty"*, *"impaired mobility"*, *"Intellectual decline is minimal to none"*, *"unable to swallow"* and *"unable to breathe"*. Phenotypes are mainly mentioned in the disease definition, clinical description and etiology sections.

1.1.2 Orphadata

Orphadata [Orp15a] provides datasets related to RDs and orphan drugs. These datasets are freely-accessible on the Orphadata website [Orp15a]. It includes:

- An inventory of RDs, cross-referenced with other terminological resources such as OMIM, ICD-10, UMLS, MeSH and MedDRa;
- A list of phenotypes, *i.e.,* a list of signs and symptoms, that are cross-referenced with other terminological resources such as HPO and PhenoDB;
- Phenotypes associated with RDs;
- Genes associated with RDs;
- Orphanet classifications that are based on published expert classifications;
- The Orphanet Rare Disease Ontology (ORDO);
- Epidemiological data (class of prevalence, average age of onset and average age at death) extracted from the literature.

Orpahdata is a valuable resource for D-P relationship extraction task. It lists 8,644 RDs, 1,273 phenotypes and 52,503 D-P relationships that could be used for extracting D-P relationships from text. Among these 8,644 RDs, only 2,689 (31.11%) are associated with phenotypes. For instance, Orphadata lists 37 phenotypes for the Kennedy disease: *"Gynecomastia"*, *"breast"*, *"mammary gland enlargement"*, *"hyperplasia"*, *"Impotence"*, *"painful erection"*, *"priapism"*, *"erection troubles"*, *"Abnormal gait"*, *"Movement disorder"*, *"Hypotonia"*, *"Areflexia"*, *"hyporeflexia"*, *"Speech troubles"*, *"aphasia"*, *"dysphasia"*, *"echolalia"*, *"mutism"*, *"logorrhea"*, *"dysprosodia"*, *"Muscle hypotrophy"*, *"atrophy"*, *"dystrophy"*, *"agenesis"*, *"amyotrophy"*, *"X-linked recessive inheritance"*, *"Small"*, *"atrophic"*, *"hypoplastic testes"*, *"monorchism"*, *"microorchidism"*, *"anorchia"*, *"Insulin-independent"*, *"type 2 diabetes"*, *"Hyperlipidemia"*, *"hypercholesterolemia"* and *"hypertriglyceridemia"*. It is hard to map these phenotypes to those mentioned in the Orphanet encyclopedia because in the later phenotypes are in a natural language description of the disease, thus not normalized.

Orphadata phenotypes are organized in an ontology that follows a tree structure. In this simple ontology, a tree node represents a phenotype and an edge represents a $is\text{-}a$ relationship between two phenotypes. In the rest of

the thesis, we will refer to phenotypes in an ontology as a phenotype class and hierarchical relationships between phenotype classes (*i.e.* simple superclass-subclass relationships) as subsumption relations, denoted by $\leqslant$. Given two classes c_1 and c_2, $c_1 \leqslant c_2$ means that c_2 subsumes c_1 and that every element of the class c_1 also belongs to the class c_2.

Figure 1.1 shows an excerpt from the Orphadata phenotype ontology. The phenotype classes "Hallux valgus", "Broad/bifid big toe" and "Dorsiflexed great toe", are subsumed by "Big toe anomaly (excluding absence)" phenotype class. "Big toe anomaly (excluding absence)" is subsumed by "Foot anomalies", which is subsumed by "Lower limb segmental anomalies". Finally, "'Lower limb segmental anomalies" is subsumed by "All", which is the root of all Orphadata phenotype classes.

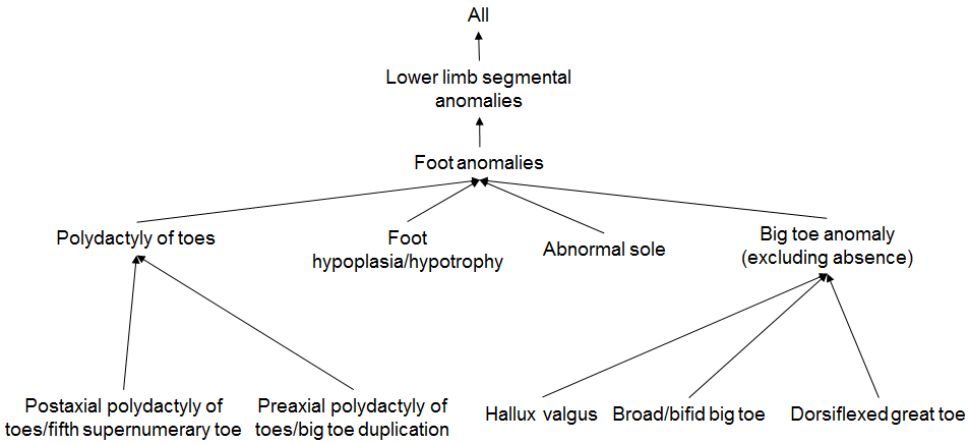


Figure 1.1: An excerpt from Orphadata phenotype ontology

Orphadata phenotype ontology is a knowledge representation for phenotypes that could be utilized as a resource for annotating medical text or normalizing discovered phenotypes with their phenotype classes.

1.1.3 OMIM

OMIM (Online Mendelian Inheritance in Man) [ABS+15] is a database of genetic diseases, associated genes and phenotypes of these diseases. OMIM is publicly available at [OMI15] and its data is regularly updated. It provides full-text summaries that contain information on many known genetic disorders and over 15,000 genes. OMIM focuses on the relationship between phenotype and genotype. Disease descriptions include a list of associated phenotypes named "clinical synopsis". For instance, OMIM shows a list of phenotypes for the Kennedy disease (OMIM ID = 313200), including *"Dysarthria"*, *"Dysphagia"*, *"Muscle cramping"* and *"Atrophy and weakness of limb musculature"*.

At present (May 2017) OMIM describes 23,929 diseases. Among those, 23,910 diseases are associated with phenotypes. OMIM contains 432,760 D-P relationships, *i.e.* the average number of phenotypes associated with OMIM disease is 18. Also, OMIM diseases have cross-references with Orphadata diseases, which could enable one to associate Orphadata diseases to its phenotypes listed in OMIM. Only 4,856 (56.18%) Orphadata diseases are associated with OMIM diseases and then could be associated with their phenotypes. For example, the Wilson disease (Orpha ID = 905) is not associated with any Orphadata phenotype. However, it is associated with OMIM (via OMIM ID = 277900) that assigns a list of phenotypes to Wilson disease such as *"Hepatic coma"*, *"Liver failure"*, *"High liver copper"*, *"Osteoporosis"* and *"Joint hypermobility"*.

1.1.4 HPO

HPO is the Human Phenotype Ontology [KDM+14], it provides a controlled vocabulary of phenotypic abnormalities related to human diseases. HPO is developed using different resources including the big medical literature, Orphanet, DECIPHER[1] and OMIM. In May 2017, HPO contains approximately 11,000 phenotype classes. Similarly to the Orphadata phenotype ontology, HPO organizes its phenotype classes in an hierarchical structure. This structure is a Directed Acyclic Graph (DAG), where a graph node represents a phenotype class and an edge may be seen as a subsumption relationship between two phenotype classes. The number of levels (or maximal depth) of the HPO ontology is 15. The average number of children for each class is 3, where the maximum number of children for a class is 31. The number of leaves, *i.e.,* number of classes that do not have any child, is 6,681.

Figure 1.2 shows an excerpt from the HPO ontology. In the figure, the phenotype class "Facial myokymia" is subsumed by both "Myokymia" and "Abnormality of facial musculature", where "Myokymia" is subsumed by "Abnormality of movement". All HPO phenotype classes are subsumed by "Phenotypic abnormality", which indeed is subsumed by "All". The "All" class is the root of all HPO classes and subsumes 5 different classes including: *(1)* Frequency, *(2)* Mortality/Aging, *(3)* Mode of inheritance, *(4)* Clinical modifier and *(5)* Phenotype abnormality.

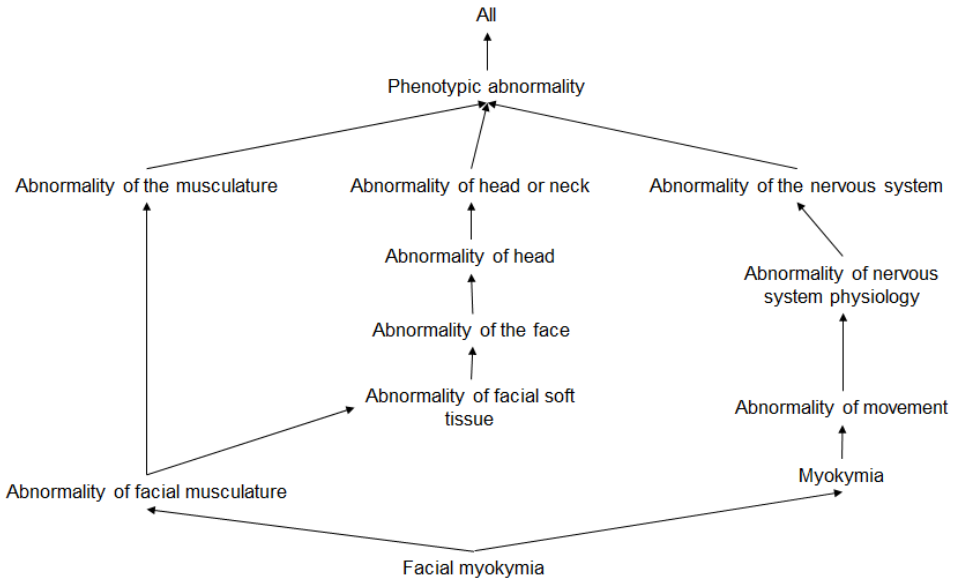


Figure 1.2: An excerpt from the HPO ontology

HPO also contains a large set of annotations for about 4,000 common diseases. These annotations associate HPO phenotypes to 2,687 Orphadata diseases and 3,892 OMIM diseases. The average numbers of HPO phenotypes annotating an Orphadata disease and OMIM disease are 24.5 and 17.75 respectively. HPO data are freely available at [HPO15].

1.1.5 UMLS

The UMLS [Bod04], Unified Medical Language System, contains health and biomedical vocabularies, and provides a mapping structure among these vocabularies. It can be considered as a comprehensive resource of

[1]DECIPHER database contains data from 21684 patients that is publicly available at `https://decipher.sanger.ac.uk/` and is used to share and compare phenotypic and genotypic data.

a large number of biomedical concepts. The UMLS consists of three knowledge resources:

- The Metathesaurus: it includes over 1 million biomedical concepts from many vocabularies such as CPT®, ICD-10-CM, LOINC®, MeSH®, RxNorm and SNOMED CT®. Each concept in the UMLS Metathesaurus is associated with a Concept Unique Identifier (or CUI).
- The Semantic Network: it encompasses broad categories named semantic types and their relationships (semantic relations). There are about 133 unique semantic types for both entities (*e.g.,* Organism, Anatomical Structure, Sign or Symptom, Clinical Attribute) and events (*e.g.,* Diagnostic Procedure, Biologic Function, Disease or Syndrome), and 54 unique relationships (*e.g.,* "physically related to", "is part of", "may-cause", "is caused by"). Each UMLS concept (CUI) in the UMLS Metathesaurus is associated with one semantic type or more from the UMLS semantic network.
- SPECIALIST Lexicon: it is a database of lexicographic information (Medical English lexicon) of biomedical terms. It contains over 200,000 terms and holds the criteria for the organization of the terms and concepts in the Metathesaurus. The SPECIALIST Lexicon is used by lexical tools to help in developing Natural Language Processing (NLP) applications.

In addition to these knowledge sources, some supporting tools have been provided by the National Library of Medicine such as MetaMap, MetamorphoSys, lvg and Knowledge Source Server. For instance, MetaMap [AL10] is a tool that recognizes the UMLS concepts in biomedical texts. It is based on symbolic, NLP, and computational-linguistic techniques for biomedical concept recognition. In Example 1.1.1, we applied MetaMap to recognize the UMLS concepts mentioned in the text. The recognized concepts are between the following tags <UMLSConcept> and </UMLSConcept>. In the start tag, <UMLSConcept>, the CUI of the concept and its semantic type are attached to the attributes 'CUI' and 'ST' respectively. MetaMap recognizes several disease concepts (ST='dsyn') such as "Familial Mediterranean fever", "peritonitis", "pleuritis", "arthritis" and "erysipelas", and one sign or symptom (ST='sosy'), "erythema". MetaMap fails here at recognizing "recurrent episodes of fever" as a phenotype.

Ex. 1.1.1

[*from PMID:23400211*] "<UMLSConcept CUI='C0031069' ST='dsyn' > Familial Mediterranean fever </UMLSConcept> (FMF) is an autosomal recessive disease characterized by recurrent episodes of fever accompanied by <UMLSConcept CUI='C0031154' ST='dsyn'> peritonitis </UMLSConcept>, <UMLSConcept CUI='C0032231' ST='dsyn'> pleuritis </UMLSConcept>, <UMLSConcept CUI='C0003864' ST='dsyn'> arthritis </UMLSConcept>, or <UMLSConcept CUI='C0014733' ST='dsyn'> erysipelas </UMLSConcept> -like <UMLSConcept CUI='C0041834' ST='sosy'> erythema </UMLSConcept>."

1.2 Corpora

1.2.1 MEDLINE

MEDLINE (Medical Literature Analysis and Retrieval System Online) is a bibliographic database that contains publications and abstracts from biomedical and health care journals. MEDLINE consists of more than 23 million publications from journal articles (full texts from more than 1,400 journals and search citations from more than 5,600 biomedical journals) in life sciences with a concentration on biomedicine. A characteristic feature of MEDLINE is that its records are indexed with concepts of MeSH (Medical Subject Headings), which is a large vocabulary from the UMLS. The number of MEDLINE publications increases rapidly. The chart in Figure 1.3 presents the number of indexed publications added to MEDLINE during each year from 1995 to 2016 [MED17]. The amount of MEDLINE publications is a valuable resource for a lot of biomedical text mining applications such as Information Retrieval and Relationship Extraction.

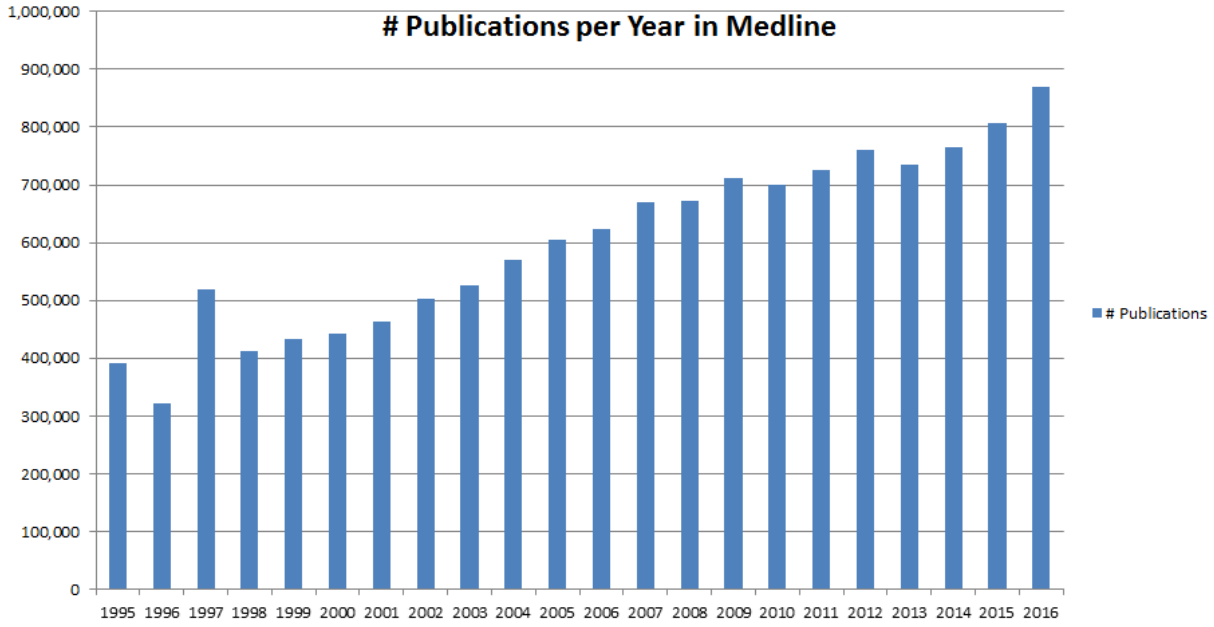


Figure 1.3: The chart shows the number of indexed publications added to MEDLINE during each year from 1995 to 2016. Data are re-plotted from [MED17]

Engines such as PubMed and Entrez have been designed to search MEDLINE. For instance, PubMed provides free access to MEDLINE and links to full-text articles via an easy-to-use interface and a search engine [LW07]. PubMed utilizes an Automatic Term Mapping feature to search for the following: *(1)* subjects using the MeSH terms, *(2)* journals and *(3)* authors. It first starts to search for subjects. When a match is found, the mapping process is complete and does not continue on to the next step. If there is no a match, then it starts searching for journals and so on till a match is found or the search stops. PubMed is publicly available online [PUB15].

1.2.2 Some Annotated Corpora

Annotated corpora are necessary resources for supervised learning approaches. For example, a machine learning approach such as SVM requires an annotated corpus for training the classification model. In this thesis we aim at extracting D-P relationships. So, we are looking for corpora that are annotated with the following entities: diseases and phenotypes, and with the relationships between them. There are few corpora that are freely available online and annotated with these entities such Arizona Disease Corpus (AZDC) [DL12], NCBI disease corpus [DLL14], PhenoCHF [ATBNA15] and BioText [RH04]. Although, they are not annotated with D-P relationships, they provide annotation guidelines that could be standard for annotating new corpora.

AZDC contains 2,783 sentences from 793 PubMed abstracts that are manually annotated with 3,224 diseases mentions (1,202 unique mentions). These mentions are mapped to UMLS concepts with relevant semantic types (*e.g.,* Acquired Abnormality, Anatomical Abnormality, Disease of Syndrome, Injury or Poisoning, Sign or Symptom). The NCBI disease corpus extends AZDC corpus to cover all 6,881 sentences of the same 793 PubMed abstracts to annotate 6,892 disease mentions (790 unique mentions). The semi-automatic annotation has been employed for annotating the NCBI disease corpus by using automatic classifier as the basis of the annotation process. Then, 14 annotators, two-annotators per document (randomly paired), reviewed and completed the annotations manually. The annotations guidelines of NCBI are similar to those of AZDC. Example 1.2.1 is

Corpus	Source	Entities	Relationships	Annotation	Link
AZDC	793 PubMed abstracts	D	-	Manually	[AZD15]
NCBI	793 PubMed abstracts	D	-	Semi-automatic	[NCB15]
PhenoCHF	300 clinical records + 10 full papers	P, O	links between considered medical terms	Manually	[Phe15]
BioText	40 abstracts + 100 titles from MEDLINE	D, T	<D,T>	Manually	[Bio15]

Table 1.1: A summary of the 4 introduced annotated corpora in terms of their source, the annotated entities and relationships they contain and a link to the corpus. In the third column, D stands for Disease, P for Phenotype, O for organ and T for Treatment. While in the fourth column "Relationships", <D,T> means a disease-treatment relationship.

an instance of annotated sentence from the NCBI corpus. In this example, the "Complement C7 deficiency" is annotated as a disease and is located between the tags <category="SpecificDisease"> and </category>.

Ex. 1.2.1 "<category="SpecificDisease"> Complement C7 deficiency </category>: seven further molecular defects and their associated marker haplotypes.

While AZDC and NCBI corpora contain disease mentions, PhenoCHF contains phenotype mentions that are manually annotated by domain experts. PhenoCHF corpus is consisting of documents from narrative reports (300 discharge summaries) and documents from literature articles (10 full-text papers) that are related to congestive heart failure. It annotates medical terms that denote phenotypic information about Congestive Heart Failure (CHF) disease such as causes, risk factors and clinical signs/symptom, and organs (*i.e.,* body parts). It also annotates the relationships, such as negation or causality, between any two of these annotated terms. Some of these relationships are linked implicitly to CHF disease because this corpus considers only this disease. However, PhenoCHP is limited to a single disease (CHF); it shows favorable results that would encourage the efforts toward further additions and enhancements for extracting phenotypic information for other diseases.

BioText corpus, provided by the BioCreAtIvE challenge [Bio15], consists of 40 abstracts and 100 titles obtained from MEDLINE. It is annotated manually with the diseases and treatment mentions. In addition, it is annotated with semantic relationships between diseases and treatments. It does not annotate phenotypes and consequently does not annotate D-P relationships. However, it is a useful resource that shows the annotation guidelines for annotating relationships between two entities. For instance, example 1.2.2 shows the annotation of the relationship between the disease "sore throat" and the treatment "Antibiotics". When a relation is found between a disease and a treatment, the disease and the treatment are located between the following tags respectively: *(1)* <TREAT> </TREAT> and *(2)* <DIS> </DIS>. If there is no a relation between them, then they are located between the following tags respectively: *(1)* <TREATONLY> </TREATONLY> and *(2)* <DISONLY> </DISONLY> as shown in Examples 1.2.3 and 1.2.4.

Ex. 1.2.2 " <TREAT> Antibiotics </TREAT> prescribed for <DIS> sore throat </DIS> during the previous year had an additional effect (hazard ratio 1.69 , 1.20 to 2.37) .

Ex. 1.2.3 " <TREATONLY> Heterologous vaccines : </TREATONLY> proponent sparks some interest.

Ex. 1.2.4 " <DISONLY> Chronic pancreatitis </DISONLY> and <DISONLY> carcinoma of the pancreas </DISONLY>

Table 1.1 presents a summary of the 4 introduced annotated corpora (AZDC, NCBI, PhenoCHF and BioText) in terms of their textual source, the annotated entities and relationships they contain and a link to the corpus.

1.3 Conclusion and Discussion

The availability of biomedical data and knowledge resources about RDs and their phenotypes is useful in the medical diagnosis. These resources could also be useful for biomedical knowledge discovery since they can either serve as an input or serve as a guide to the KDD process. In this chapter, we presented some public resources that we will use in this thesis such as Orphanet, Orphadata, OMIM, HPO and UMLS. Orphanet aims to provide RD information to health-care professionals, patients, and their relatives, in order to improve the diagnosis, care and treatment of these diseases. Orphanet provides summaries for the most of them, which mention RD phenotypes. These summaries are written and updated manually by experts, what requires a lot of time and efforts. Orphadata, OMIM and HPO list *D-P* relationships. However, they are incomplete in comparison with what is in the literature. Therefore, a method for extracting automatically these relationships from literature is of interest.

MEDLINE is a valuable resource of biomedical literature for biomedical researchers. The millions of MEDLINE publications contain valuable information (*e.g.,* D-P relationships). The manual extraction of this information from a very large set of documents is a difficult and costly task. Therefore, an automatic method based on linguistics and data mining techniques may help extracting such kind of information.

There are few corpora that have been annotated with disease mentions (*e.g.,* AZDC, NCBI disease corpus, BioText) or with phenotype mentions (*e.g.,* PhenoCHF). Although they do not provide annotations of D-P relationships, they provide helpful guidelines for annotating a corpus. PhenoCHF provides relationships between medical terms representing phenotypic information such as negation and causality. But, these are related and limited to only one single disease, which is CHF disease.

Starting from this point, a textual corpus related to RDs could be retrieved from MEDLINE via PubMed. This corpus should be annotated with the desired annotations (*e.g.,* diseases, phenotypes, D-P relationships). The corpus annotation could be done manually or the basis of other annotation tools (*e.g.,* MetaMap). For instance, MetaMap could be used to annotate a corpus with UMLS concepts (*e.g.,* diseases and phenotypes). Then, the annotations of D-P relationships could be added manually. The resulted corpus is used for learning patterns that define the relationships between diseases and phenotypes. Then, these patterns are applied to the literature for extracting new D-P relationships. Finally, the discovered D-P relationships are compared to what exist in the D-P databases and ontologies such Orphadata, OMIM and HPO to define their validity and novelty. The novel extractions could be suggested as enrichment for the content of databases such as Orphadata and Orphanet summaries.

2

Data Mining and Classification Approaches

Contents

Data mining is the process of extracting previously unknown or interesting patterns from large databases [FPSS96]. There is a huge amount of data that is being collected and warehoused such as Web data, e-commerce, purchases and Bank transactions. Processing this large volume of data using traditional database techniques is not an easy task and may be time consuming for decision makers. Data mining helps at finding hidden patterns in large data that could be used to discover previously unknown relationships and predict future behavior, which may be useful for decision makers.

Data mining can be categorized in two main groups: descriptive data mining and predictive data mining [Mcc06]. Descriptive data mining looks for human-interpretable patterns that describe the data such as clustering, Association Rule Mining and more generally pattern discovery. Predictive data mining is based on mathematical models, that are usually learned from a training dataset to predict unknown or future values. Examples of such approaches are classification, regression or Deviation Detection tasks.

We introduce in this chapter different techniques for pattern discovery and different classification approaches in sections 2.1 and 2.2 respectively. Section 2.3 focuses on particular classification approaches: Formal Concept Analysis and Pattern Structures. Finally, section 2.4 ends this chapter with the discussion and conclusion.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Wipes, Egg
3	Milk, Diaper, Wipes, Coke
4	Bread, Milk, Diaper, Wipes
5	Bread, Milk, Diaper, Coke

Table 2.1: Example of transactions in a market basket database

2.1 Pattern Discovery

Pattern discovery aims at discovering interesting patterns describing regularities in a database. Pattern discovery algorithms can be applied to various types of data such as transaction databases, sequential databases, streams, spatial data, trees, graphs, etc.

The definition of what is an interesting pattern is fuzzy and varies depending on the application of the mining task. Some authors define an interesting pattern as a pattern that appears frequently in a database [MTV94, AMS$^+$96]. But others are interested in rare patterns [SNV07], patterns with a high confidence [AIS93] or stable patterns (*i.e.,* patterns that still stand when data change slightly) [BKN14].

2.1.1 Itemset Mining

Definition 1 *(Itemset)*
Given a finite set of items $I = \{i_1, i_2, \ldots, i_n\}$, an itemset $X = \{X_1, X_2, \ldots, X_n\}$ is a non-empty subset of I where k is the size of the itemset x, noted $|X| = k$.

Itemset mining was first proposed by Agrawal *et al.* in 1993 [AIS93] for supermarket transaction. It aims at discovering a set of items (*e.g.,* products, actions) that occur together. Let's introduce a simple example for illustrating the basic concepts of itemset mining. Table 2.1 shows a set of shopping transactions, known in the literature as "market-basket" model of data. This table consists of two columns: one for transaction ID (TID) and one for the list of items (*e.g.,* products) bought in each transaction. {Bread, Milk, Diaper} is an example of itemset made of products that may be bought together. k-itemset is an itemset that contains k items, where k is the size of the itemset. In our example, {Bread, Milk, Diaper} is a 3-itemset as it contains 3 items 'Bread', 'Milk' and 'Diaper'. The support of an itemset is the fraction of transactions that contains this itemset. For example, the frequency of {Bread, Milk, Diaper} is 2 because it is included in two transactions (with TIDs 4 and 5). The total number of transaction in this example is 5, then the support of thisitemset is $2/5$.

Frequent Itemsets

Definition 2 *(Frequent Itemset)*
Given a transnational database and minimal support threshold $min\text{-}sup$. An itemset X is frequent if and only if $support(X) \geqslant min\text{-}sup$.

Frequent itemset mining, also known as frequent pattern mining, focuses on selecting only itemsets that are more frequent to a specific threshold. It aims at retrieving the itemsets whose supports are greater than or equal to a minimum support, denoted $min\text{-}sup$. For example, if we consider $min\text{-}sup = 3/5$ and the example provided in Table 2.1, then a frequent itemset mining algorithm should retrieve all itemsets with a support equal or higher to $3/5$: The following itemsets are retrieved:

- Frequent 1-itemsets:
  {Bread} with $support = 4/5$, {Milk} with $support = 4/5$, {Diaper} with $support = 4/5$ and {Wipes} with $support = 3/5$
- Frequent 2-itemsets:
  {Bread, Milk} with $support = 3/5$, {Bread, Diaper} with $support = 3/5$, {Milk, Diaper}with $support = 3/5$, {Diaper, Wipes} with $support = 3/5$
- Frequent 3-itemsets: none

A simple algorithm to find the frequent itemsets is Apriori [AS94a]. Apriori starts by first considering the frequent itemsets of length 1 (*i.e.,* 1-itemsets). Then, the frequent itemsets of length 1 are used to generate candidates of length 2, which support are tested to be higher or equal to $min\text{-}sup$. For example, {Bread} is a frequent 1-itemset as its support is $4/5$, which is greater than $min\text{-}sup = 3/5$. As {Bread} is frequent, it is used to generate candidates for 2-itemsets such as {Bread, Milk}, {Bread, Diaper}, {Bread, Wipes}, {Bread, Egg}, {Bread, Coke}. By checking the database for support computing, their supports are $3/5, 3/5, 2/5, 1/5$ and $2/5$ respectively. {Bread, Milk} and {Bread, Diaper} are frequent 2-itemsets because their supports are equal to $min\text{-}sup$, while {Bread, Wipes}, {Bread, Egg} and {Bread, Coke} are infrequent because their supports are less than $min\text{-}sup$ and consequently they are filtered out. Then, in a similar way, the frequent 2-itemsets {Bread, Milk} and {Bread, Diaper} are used to generate candidates for frequent 3-itemsets and so on. This simple algorithm suffers from a poor performance as it needs a large number of database scans. This happens when the candidate generation generates large numbers of subsets and thus requires to compute the support of each subset. Several faster algorithms have been proposed to overcome this issue such as [BMUT97,DH07,MD14,QGYH14,LYSZ16].

Closed Itemsets

Definition 3 *(Superset)*
Superset of an itemset is an itemset that includes it, and has only one additional element.

Definition 4 *(Closed Itemset)*
Given a database and a minimal support threshold $min\text{-}sup$. An itemset is closed if it is frequent and none of its immediate supersets have the same support.

To identify closed itemset, a naïve approach is to first extract all frequent itemsets. Then, closed itemsets may be completed by checking the support of supersets of each frequent itemset. Considering our running example, $\{Wipes\}$ and $\{Diaper, Wipes\}$ are two frequent itemsets with $support = 3/5$. As $\{Diaper, Wipes\}$ is an immediate superset of $\{Wipes\}$ and as they have the same support, then, $\{Wipes\}$ is not a closed itemset. Differently, $\{Bread, Milk\}$ is a superset of $\{Milk\}$ and their supports are respectively $3/5$ and $4/5$. In this case, $\{Milk\}$ is a closed itemset. Using the dataset example of Table 2.1, one can generate the following closed itemsets: $\{Bread\}$, $\{Milk\}$, $\{Diaper\}$, $\{Bread, Milk\}$, $\{Bread, Diaper\}$, $\{Milk, Diaper\}$ and $\{Diaper, Wipes\}$.

Maximal Itemsets

Definition 5 *(Maximal Itemset)*
Given a transnational database and a minimal support threshold $min\text{-}sup$, an itemset is maximal if it is closed and none of its immediate supersets is closed.

A maximal itemset is a closed itemset that is not included in any other closed itemset. In other words, a closed itemset is also maximal if it has not an immediate closed superset. To identify maximal itemset, one may extract all closed itemsets; then find those that have no immediate superset closed. Maximal itemsets in our running example are: $\{Bread, Milk\}$, $\{Bread, Diaper\}$, $\{Milk, Diaper\}$ and $\{Diaper, Wipes\}$.

Itemset	Support	Frequent	Closed	Maximal
{Bread}	$4/5$	✓	✓	
{Milk}	$4/5$	✓	✓	
{Diaper}	$4/5$	✓	✓	
{Wipes}	$3/5$	✓		
{Bread, Milk}	$3/5$	✓	✓	✓
{Bread, Diaper}	$3/5$	✓	✓	✓
{Milk, Diaper}	$3/5$	✓	✓	✓
{Diaper, Wipes}	$3/5$	✓	✓	✓

Table 2.2: The closed itemsets, their support and confidence, which are discovered by an Association Rule Mining algorithm using $min\text{-}sup = 3/5$ and $min\text{-}conf = 3/5$

Relationship between Frequent, Closed and Maximal Itemsets

Table 2.2 presents the frequent, closed and maximal itemset generated from data in Table 2.1 when considering $min\text{-}sup = 3/5$. Both closed and maximal itemsets are subsets of frequent itemsets but maximal itemsets are a more compact representation because they are a subset of closed itemsets. Figure 2.1 presents the relationship between frequent, closed and maximal Itemsets. As the number of frequent itemsets grows highly when the database grows, it is necessary to generate a reduced set of itemsets. The maximal and closed frequent itemsets are two reduced representations that are subsets of the frequent itemsets and have the same representative of all frequent ones *i.e.,* they can regenerate all frequent itemsets.

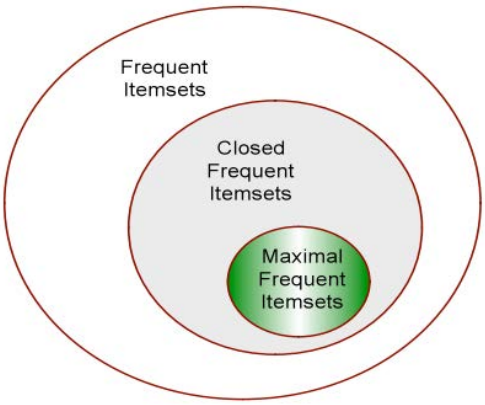


Figure 2.1: The relationship between Frequent, Closed and Maximal Itemsets [TMK05].

2.1.2 Association Rules Mining

Association Rule Mining was introduced by Agrawal *et al.* [AIS93] for finding implication relationship between frequent itemsets. It was first applied to the mining of supermarket transactions, such as itemset mining. Association Rule Mining aims at finding rules that associate the occurrence of an item to the occurrences of other items in the transaction. These rules, named association rules (AR), are presented as a collection of if-then rules. The form of an association rule is $A \rightarrow B$, where A, the antecedent of the rule, and B, the consequent, are sets of items. The interpretation of such a rule is that if all the items in A appear, then the items of B may appear as well, with a certain confidence. For instance, the rule $\{Bread\} \rightarrow \{Milk\}$ can be obtained from Table 2.1.

Association Rule	Support	Confidence
$Bread \rightarrow Milk$	3/5	3/4
$Milk \rightarrow Bread$	3/5	3/4
$Bread \rightarrow Diaper$	3/5	3/4
$Diaper \rightarrow Bread$	3/5	3/4
$Milk \rightarrow Diaper$	3/5	3/4
$Diaper \rightarrow Milk$	3/5	3/4
$Diaper \rightarrow Wipes$	3/5	3/4
$Wipes \rightarrow Diaper$	3/5	1

Table 2.3: The association rules, including their support and confidence, that are discovered by an Association Rule Mining algorithm using $min\text{-}sup = 3/5$ and $min\text{-}conf = 3/5$.

This rule means that when someone buy $\{Bread\}$, she/he is also buying $\{Milk\}$.

To measure how the association is strong, different measures such as the $support$ and $confidence$ can be used. The support of an association rule is the ratio of the number of occurrences of the itemset $A \cup B$ over the total number of transactions. The confidence of an association rule is the number of itemsets containing the antecedent and consequent of the rule divided by the number of itemsets containing its antecedent. Formally, support and confidence are defined by:

$$support(A \rightarrow B) = \frac{|A \cup B|}{\text{total number of transactions}} \tag{2.1}$$

$$confidence(A \rightarrow B) = \frac{|A \cup B|}{|A|} \tag{2.2}$$

Let's consider the following association rule $\{Bread\} \rightarrow \{Milk\}$. Its support, $support(\{Bread\} \rightarrow \{Milk\})$, is the support of the itemset $\{Bread, Milk\}$ or $support(\{Bread, Milk\}$, which is $3/5$ according to the transaction database provided in Table 2.1. Its confidence, $confidence(\{Bread\} \rightarrow \{Milk\})$, is $support(\{Bread\} \cup \{Milk\})/support(\{Bread\})$, or 3/4.

Minimal $support$ and $confidence$ thresholds are user defined values that are fixed to select only a subset of rules. Frequent association rule mining searches for the subset of AR that have a support greater than or equal to the threshold denoted $min\text{-}sup$. Valid association rule mining searches for the rules that have a support greater than or equal to $min\text{-}sup$ and a confidence greater than or equal to a specific confidence denoted $min\text{-}conf$. Using $min\text{-}sup = 3/5$ and $min\text{-}conf = 3/5$, Table 2.3 presents all valid ARs that could be discovered form the database presented in Table 2.1.

A naïve algorithm can mine valid rules by first retrieving all frequent itemsets that satisfy the $min\text{-}sup$ threshold. Then, it generates rules from the frequent itemsets by dividing each itemset into $Antecedent$ and $Consequent$. For example, if an itemset C may be divided in two subsets A and B (*i.e.,* $C = A \cup B$), then we can take A as an $Antecedent$ and B =$((A \cup B) - A)$ as a $Consequent$. Then, the rule $A \rightarrow B$ may be constructed. Finally, the confidence of each rule is computed according to the equation 2.2; and then only rules with $confidence \geq min\text{-}conf$ are selected as valid association rules. This naïve algorithm needs to access the database frequently, what results in poor performances. Therefore, more efficient and faster algorithms have been proposed in the literature such as [AS94b, PCY95, RKK15, SDG15].

TID	Items
$S1$	$\langle\{a\},\{i\}\rangle$
$S2$	$\langle\{a,b\},\{c\},\{d,f,g\}\rangle$
$S3$	$\langle\{c\},\{e\},\{g\}\rangle$
$S4$	$\langle\{c\},\{d,g\},\{i\}\rangle$
$S5$	$\langle\{i\}\rangle$

Table 2.4: An Example of Sequential Database

2.1.3 Sequence Mining

A sequential database consists of ordered items or events. A sequence in a sequential database is an ordered list of itemsets, denoted $\langle e_1, e_2, \ldots e_n\rangle$. Sequences are type of data structures that can be found in many domains such as marketing (sequences of customer transactions), bioinformatics (sequences of nucleotides in DNA, medical informatics(sequence of drug treatments), NLP (sequences of words in a sentence), etc. An example of sequential database may be a list of customer transactions ordered by increasing transaction data. Table 2.4 shows an example of such a sequential database consisting of 5 customer transactions $S1$ to $S5$. Each transaction is a sequence of itemsets. For instance, $S2 = \langle\{a,b\},\{c\},\{d,f,g\}\rangle$ consists of an order of the 3 itemsets $\{a,b\}$, $\{c\}$ and $\{d,f,g\}$.

A sequential pattern is a subsequence that appears in several sequences of a sequential database. It is associated with a support, which is the number of time the pattern occurs over the number of transactions. For example, the sequential pattern $\langle\{c\},\{g\}\rangle$ appears in three sequences ($S1$, $S2$ and $S3$) in the sequential database presented in Table 2.4. It indicates that customers who bought product $\{c\}$, bought product $\{g\}$ next 3 times over 5 transactions. The support of this pattern is consequently $3/5$.

Definition 6 *(Subsequences)*
A sequence $B = \langle b_1, b_2, \ldots, b_m\rangle$ is a subsequence of another sequence $A\langle a_1, a_2, \ldots a_n\rangle$ if there exist integers $1 \leqslant j_1 < j_2 < \ldots < j_m \leqslant n$ such that $b_1 \subseteq a_{j_1}, b_2 \subseteq a_{j_2}, \ldots, b_n \subseteq a_{j_n}$.

Definition 7 *(Frequent Sequential Pattern)*
Given a sequential database and minimal support threshold $min\text{-}sup$. The sequence S is called a frequent sequential pattern if and only if $support(S) \geqslant min\text{-}sup$.

Sequence Mining, or Sequential Pattern Mining, aims at finding the frequent sequence patterns in a sequential databases. It is similar to frequent itemsets mining, but with consideration of an order.

Several algorithms have been proposed for finding sequential patterns from a sequential database such as GSP [SA96], SPADE [Zak01], FreeSpan [HPMA$^+$00], PrefixSpan [PHMA$^+$04] or CloSpan [YHA03]. These algorithms take as input a sequence database and a minimum support threshold ($min\text{-}sup$). GSP, SPADE, FreeSpan and PrefixSpan generate all frequent sequential patterns having a support greater than or equal to $min\text{-}sup$. CloSpan algorithm mines the closed sequential patterns. GSP and SPADE are apriori-based approaches. FreeSpan and PrefixSpan are Pattern-growth approaches. Apriori-based approach uses frequent patterns of length-(k-1) to generate candidates patterns of length-k. Initially, every item in the database is a candidate of length-1. At each level (*i.e.,* sequences of length-k), a database scan is required to compute the support of each candidate sequence. Then, the frequent patterns of length-k are selected based on their support. These length-k frequent sequences are used to generate candidate sequences of length-(k+1). This process is repeated until no frequent sequence or no candidate can be found. In Pattern-growth approaches, sequence databases are recursively projected into a set of smaller projected databases. Sequential patterns are grown in each projected database by exploring only locally frequent sequences.

Raïssi and Plantevit [RP08] proposed an algorithm, named MDSDS, for mining multidimensional sequential patterns in streaming data. MDSDS searches for the most specific multidimensional items, which are items

defined on a set of n dimensions. Then, PrefixSpan [PHMA+04] is used to find the sequences containing only these items. Egho *et al.* [ERI+12] proposed a sequential pattern mining approach for Healthcare Trajectory, that also mines multidimensional itemset sequential patterns. They presented a new algorithm in [EJR+13], named MMISP (Mining Multidimensional Itemsets Sequential Patterns), which relies on external taxonomic knowledge to enrich the mining process and provides results with appropriate levels of granularity. MMISP is mainly based on transferring the multidimensional itemsets sequential database into a classical sequential database.

2.1.4 Graph Mining

Definition 8 *(Graph)*
A graph G is defined as a pair (V, E) where V is a set of vertices and E is a set of edges connecting vertices such as $E \subseteq V \times V$.

Definition 9 *(Directed Graph)*
A graph is a directed graph also called digraph if each edge is an ordered pair of vertices, where $\{v_i, v_j\} \neq \{v_j, v_i\}$.

A graph is a *labeled graph* when vertices and edges are associated with labels. A Directed Acyclic Graph (DAG) is a directed graph containing no directed cycles. Acyclic means that there is no path that connects a vertex to itself.

Frequent Subgraph Mining

Graph Mining is the process of searching for relevant information from data structured in the form of graphs. Graph mining has been applied to the mining of biochemical structures [WNK10] and social networking [TL10]. One simple approach in graph mining is to search for frequent subgraphs, *i.e.,* subgraphs with a support greater than or equal to a threshold named min_sup [KK01].

Definition 10 *(Subgraphs)*
$S = (SV, SE)$ is a subgraph of $G(V, E)$, denoted $S \subseteq G$, if $SV \subseteq V$ and $SE \subseteq E$.

Definition 11 *(Subgraph Support)*
The frequency of a subgraph S_i is the number of its occurrences in $\mathcal{G}$, where $\mathcal{G}$ is the collections of subgraphs of a graph G. The support of S_i is $\frac{|S_i|}{|\mathcal{G}|}$.

Definition 12 *(Frequent Subgraph Mining)*
Given a graph collection $\mathcal{G} = \{G_1, G_2, ..., G_k\}$, with $G_i = (V_i, E_i)$, and a minimum support min_sup, the Frequent Subgraph Mining task (denoted FSM) extracts the collection of subgraphs $\mathcal{S} = \{S_1, ..., S_n\}$ that occurs in $\mathcal{G}$, with a $support(S_i) \geqslant min\_sup$.

Frequent Subgraph Mining (FSM) is a graph mining method that extracts frequently occurring subgraphs either from a single large graph or a set of graphs [KK01]. FSM algorithms are mainly based on two distinct approaches: *Apriori*-based and *pattern growth*-based approaches. *Apriori*-based graph mining algorithms share similarities with Apriori-based frequent itemset mining algorithms [AS94a]. In their case, the search for frequent subgraphs starts with graphs with no edge. At each iteration, the size of the newly discovered frequent substructures is increased by one by joining two subgraphs from the previous iteration. AGM [IWM00], FSG [KK01] and FFSM [HWP03] are examples of Apriori-based algorithms. The *pattern-growth* mining algorithms extend a frequent graph by trying to add successively a new edge to every possible position. If the new graph is frequent, a new frequent graph can

be expended; if it is not frequent an alternative edge is tried to be added. gSpan [YH02], CloseGraph [YH03]and Gaston [NK05] are examples of pattern-growth algorithms.

Mining subgraph patterns from graph databases is a challenging task because of graph related operations, such as subgraph isomorphism, which generally have a higher complexity than the corresponding operations on itemsets or sequences. Subgraph isomorphism is a NP-complete problem (*i.e.,* no polynomial algorithm can solve it), and its running time is exponential [Epp99]. So, efficient graph mining algorithms need optimized techniques to determine whether a subgraph pattern may be generated or not and pruning techniques to reduce the complexity of testing subgraph patterns.

An example of FSM: gSpan

gSpan is a FSM algorithm that processes undirected labeled graphs. Given a collection of such graphs, gSpan returns the set of frequent subgraphs and their support without candidate generation. Avoiding candidate generation improves the performance by avoiding testing false candidates. gSpan generates a Tree Search Space (TSS) that is composed of all trees and subtrees that rely on the collection of graphs. gSpan represents each tree of the TSS using a specific encoding, named *minimum Depth-First Search (DFS) Code*. This code is unique for each tree because it is constructed following the unique DFS traversal that follows the lexicographic order of vertex labels. gSpan follows a pattern-growth mining approach, *i.e.,* expends at each iteration a frequent graph with a new edge, trying every potential position. An issue with this approach is that the same graph can be discovered several times from different frequent graphs. gSpan avoids this problem by introducing a *right-most extension technique*, where edge extensions only take place on a specific position determined by DFS Codes. This enables gSpan to discover frequent subgraphs efficiently without generating useless candidates.

2.2 Classification Approaches

Classification is a data mining task that aims at associating sets of data instances to classes. It aims at creating a classifier that predicts the class based on several input variables. A classification method is classically based on two steps: (1) a learning step for building a classifier and (2) a classification step for applying the classifier to future or unknown instances.

The goal of this section is to provide an introduction of different classification techniques and to position and detail the methods we used in this thesis. In particular, we distinguish in this thesis symbolic approaches (*e.g.,* Rule-based (Decision Trees, Association Rules), FCA-based classification) to numerical approaches (Probabilistic-based approaches (Naïve Bayes – Bayesian Networks), Instance-based Classifiers (*e.g.,* IBK – k-Nearest Neighbors), Support Vector Machines, Neural Networks).

2.2.1 Symbolic Approaches

Symbolic Approaches could be used to automatically extract rules or patterns from the data. These rules or patterns are easy to understand and to interpret, in regards to numerical approaches. Examples of symbolic approaches include Rule-based classification, Decision Trees and Lattice-based classification.

Rule-based Approaches

Rule-based classifiers make use of a set of IF-THEN rules for classification. A rule can be expressed as: IF *condition* THEN *conclusion*, or *condition* $\rightarrow$ *conclusion*. The IF part of the rule is sometimes called the antecedent, while the THEN part is called the consequent. The condition is a conjunction of attributes. In a classification task, the consequent is the class prediction itself. ZeroR and OneR are naïve examples of Rule-based

classifiers. ZeroR is the simplest classification method which relies on the target class and ignores all predictors. It constructs a frequency table of the target class values and selects the most frequent class as the prediction. ZeroR classifier simply predicts a new instance as belonging to the majority class. Although ZeroR does not have predictability power, it could be used as a baseline classifier to compare with other classification methods. OneR (one-attribute-rule) classifier [Hol93] searches for the attribute that makes fewest prediction errors. It generates one rule for each attribute and then selects the rule that predicts the right class with the smallest number of errors. To create a rule for an attribute, a frequency table is constructed for each value against the target.

Association rules have been predominantly used for data exploration and description tasks. However, they can also be used for prediction tasks. The use of association rules for classification was first proposed by Liu *et al.* [LHM98]. They adapted the Apriori's algorithm to generate association rules that are then used to build a classification model. The adapted algorithm is called CBA (Classification Based on Associations). CBR generates association rules that have the particularity of having only one attribute in the consequent, which is the target class. These association rules are called class association rules (CARs). Association rules have been applied successfully to different classification applications such as document classification [YL05], classification of web documents, classification of mammography images [ZAC02], classification of spatial data [CAM04], recommendation systems [LAR02], and text categorization [CYZH05].

Decision Trees [Qui86] are non-parametric supervised learning methods and are commonly used in data mining for classification or regression. They are used to predict the class of a target variable by learning simple decision rules from the data features. A decision tree includes a root node, branches and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. A decision tree classifies data instances by starting at the root of the tree and moving through it until a leaf node. It could be seen as a set of if-then-else decision rules, that are organized to make them easy to use and to interpret. C4.5 is one of the common algorithms developed by Quinlan [Qui93] to build decision trees using a set of training data and the concept of information entropy. At each node in the tree, C4.5 selects the attribute of the data that best splits its set of samples into subsets that could be classified by one class or the other. The splitting criterion is based on the normalized information gain. The attribute that gives the highest normalized information gain is selected. Then, these steps are repeated on the next smaller sublists.

FCA-based Approaches

Formal Concept Analysis (FCA) is another symbolic approach that is detailed in subsection 2.3. It may be seen as an unsupervised classification approach and has been applied for classification tasks [OPG13]. Nguyen et al. [NVHT12] proposed a lattice-based approach that uses a lattice generated by FCA for mining CARs. The lattice structure helps to check easily if a rule generated from a lattice node is redundant or not by comparing it with all its parent nodes. Asses *et al.* [ABB$^+$12] proposed a hybrid method based on FCA and Emerging Patterns for the classification of biological inhibitors. This method uses FCA for building a concept lattice and then finding the concepts whose extents determine classes of objects sharing the same labels. Jumping Emerging patterns (JEPs) are used as a supervised method to predict the class of unknown objects [KW11]. JEPs are generated from the lattice, where a JEP is the intent of a formal concept where all objects in the concept extent are in the same class.

2.2.2 Numerical Approaches

Probabilistic Approaches

Bayesian Networks, also known as Probabilistic Networks, are probabilistic classifiers based on Bayes' theorem [FGG97, Nea03]. They specify joint conditional probability distributions by defining the conditional independencies between subsets of attributes.

Naive Bayes classifiers are the simple form of Bayesian Networks. Naive Bayes is a conditional probability model that estimates the probability for a given tuple (including values of several attributes) to belong to a class. For example, given an instance to be classified, represented by a vector $X = (x_1, \ldots, x_n)$ representing n features where each feature x_i represents an independent variable. Naive Bayes model calculates the probabilities of assigning the instance X to each class. Using Bayes' theorem, the conditional probability of a Naive Bayes classifier is expressed as:

$$P(C_i|X) = \frac{P(C_i)P(X|C_i)}{p(X)} = \frac{P(C_i \cap X)}{p(X)} \tag{2.3}$$

$P(C_i|X)$ is the probability of assigning the instance X to a class C_i. $P(C_i \cap X)$ is the probability of observing both of C_i and X together. $P(X)$ is the probability of observing X.

Finally, Naive Bayes suggests that X belongs to a class C_i iff the probability $P(Ci|X)$ is the highest among all the $P(C_k|X)$ for all the k classes. Naive Bayes is easy to implement, it involves significant computation or many observations because it requires either to compute or to know prior knowledge on probability distribution considered.

Instance-based Learning

Instance-based learning is a family of classification algorithms that compares new instances with instances stored in the training data. K-Nearest Neighbors (k-NN), an example that is commonly known in data mining [AKA91]. It stores training instances and classifies new instances based on a distance or a similarity measure. The classification is done by comparing the feature vector of a new instance against the feature vectors of all training instances. It finds the k training examples that are closest to the new instance example. A distance measure, *i.e.*, Euclidean distance, is used to assign weights to the neighbors based on their distances from the query instance. Finally, the new instance is classified by a majority vote of its neighbors. In k-NN, k is a positive integer, typically small. If k = 1, then the instance is simply assigned to the class of that single nearest neighbor. Figure 2.2 gives an examples of k-NN classification when $k = 3$ and $k = 7$.

k-NN is non-parametric, meaning that it does not make any assumption on the underlying data distribution. k-NN is a lazy method, which means it does not need to generalize from training data. It does not need a training phase, it only needs to store all the training data and the computation is done in the testing phase.

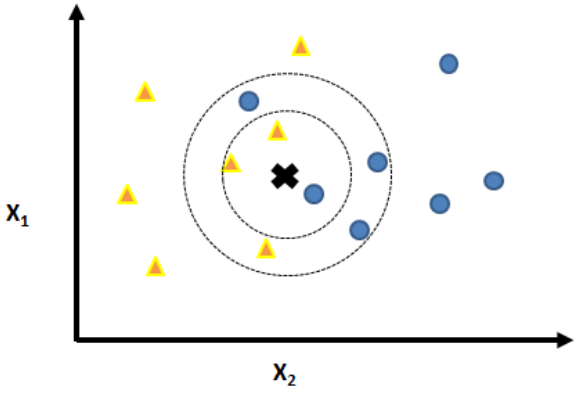


Figure 2.2: Two examples of k-NN classification in two dimensions. The test instance (black cross) should be classified either to the blue circle class or to the yellow triangle class. If $k = 3$, it is assigned to the yellow triangle class because there are 2 yellow triangles and only 1 blue circles inside the inner dashed circle. If $k = 7$, it is assigned to the blue circle class because there are 4 blue circles vs. 3 yellow triangles inside the outer dashed circle.

Support Vector Machines

Support Vector Machines (SVM) are supervised machine learning models that may be used for classification [FM06] or regression analysis [DBK+97]. A SVM model is a representation of the training examples as points in space. The examples from different categories are divided by a clear gap that is as wide as possible. In other words, SVM performs classification by finding the hyperplane that maximizes the margin between sets of instances. New examples are then assigned to the same space and associated with a category based on which side of the gap they fall on. In addition to perform linear classification, SVM can efficiently perform a nonlinear classification using what is called the kernel trick, by implicitly mapping their inputs into high-dimensional feature spaces. Indeed, kernel functions convert nonlinear separable data into linear separable data. Figure 2.3 gives an example of SVM classification, where the blue line presents the margin that best separates the blue circles and yellow triangles.

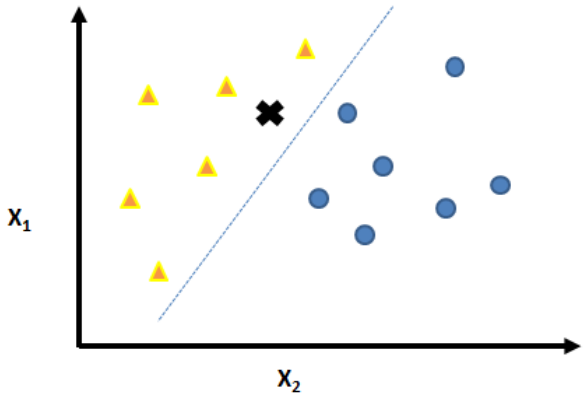


Figure 2.3: Example of SVM classification in two dimensions. The test instance (black cross) is classified as belonging to the yellow circle class as it located on the left side of the margin (*i.e.,* the blue dashed-line).

In this section, we presented various approaches for classification. We categorized these approaches into two main categories: symbolic approaches and numerical approaches. Symbolic approaches utilize the structural aspect of data and use structural or symbolic representation of data. Numerical approaches utilize the numerical aspect of data and use statistical techniques. Symbolic approaches are easy to interpret as they provide human-readable results, while numerical approaches provide poor explanations of their results, and may cause difficulties into interpretation. Deep learning, which is based on Neural Networks, is a numerical approach that relies on learning representations of data [GBC16]. It replaces handcrafted features with efficient algorithms for learning the best features automatically. It is a promising approach in machine learning and could be applied for classification tasks [KSH12] that already show its efficiency for several text mining tasks [CWB+11]. In this thesis, we did not use deep learning approaches and we leave it for the future extension of our work.

2.3 Formal Concept Analysis and Pattern Structure

FCA is a mathematical framework used for data analysis that may be used for descriptive data mining tasks [GW99]. FCA helps to analyze data described in a *formal context*, *i.e.,* a set of binary relationships between a particular set of objects and a particular set of attributes. The formal context may be seen as a binary table and is used to build a lattice of *formal concepts*. Pattern structures [GK01] are an extension of FCA enabling dealing directly with data more complex than binary tables. FCA and pattern structure have been used for knowledge discovery and classification [KK12a, CR96, CR04, ABNS15]. This section introduces the basics of FCA [GW99] and pattern structure [GK01].

	egg	feather	teeth	fly	swim	lung
Ostrich	X	X				X
Canary	X	X		X		X
Duck	X	X		X	X	X
Shark	X		X		X	
Crocodile	X		X		X	X

Table 2.5: Formal Context Example.

2.3.1 Classical Setting of FCA

Definition 13 *(Formal Context)*
A formal context is a triple (G, M, I), composed of two sets G and M and a binary relationship I. G is a set of objects, M is a set of attributes and I is relationships between G and M, $I \subseteq G \times M$. $(g, m) \in I$ iff an object $g \in G$ has an attribute $m \in M$.

Table 2.5 shows an example of formal context presented in a binary table or cross table (where crosses represent that an object has an attribute). Rows are objects from G and columns are attributes from M. In our example, the set of objects is $\{Ostrich, Canary, Duck, Shark, Crocodile\}$ and the set of attributes is $\{egg, feather, teeth, fly, swim, lung\}$. Every cross in a table cell represents an element of the binary relation I. For example, the cross between the object "$Canary$" and the attribute "fly" means that the object "$Canary$" has this attribute "fly".

Definition 14 *(Galois Connection Operators)*
Given a formal context $K = (G, M, I)$, the Galois connection between G and M is defined as:
$A' = \{m \in M : \forall g \in A, (g, m) \in I\}$, is the set of attributes that all objects in A have in common
$B' = \{g \in G : \forall m \in B, (g, m) \in I\}$, is the set of objects that have all the attributes of B.

From this formal context, FCA enables: *(1)* extracting formal concepts and *(2)* building a concept hierarchy of these formal concepts, commonly named a concept lattice. To extract formal concepts, two derivation operators known as Galois connection operators are used. A formal concept (A, B) consists of two sets A, a set of objects called a concept extent, and B, a set of attributes called a concept intent.

Definition 15 *(Formal Concept)*
A pair $(A, B), A \subseteq G, B \subseteq M$, is a formal concept iff $A' = B$ and $B' = A$. A is called the extent and B the intent of the formal concept.

The formal concepts in the lattice are partially ordered by inclusion of extents (or dually by inclusion of intents). Figure 2.4 shows the constructed concept lattice that is generated from the formal context of Table 2.5. Given two formal concepts $(A1, B1)$ and $(A2, B2)$, $(A1, B1) \leq (A2, B2)$ iff $A1 \subseteq A2$ (or dually $B2 \subseteq B1$). For instance, the formal concept ({Crocodile, Duck, Ostrich}, {egg, swim}) denoted '3' has for extent {Crocodile, Duck, Ostrich} and for intent {egg, swim}. This means that "Crocodile", "Duck" and "Ostrich" have attributes "egg" and "swim" in common. The concept ({Canary, Duck}, {lung, egg, feather, fly}), denoted '6', is a subconcept of a '3', because its extent is a subset of the extent of '3' (and dually the intent of '3' is a subset of the intent of '6').

2.3.2 Data Scaling for Many-Valued Context

In many real-world examples, attributes are not limited to binary values, but may be assigned to many different values, named many-valued attributes. For instance, an attribute "height" may be assigned to one of these values $\{short, medium, tall\}$. Contexts that contain many-valued attributes are named many-valued contexts.

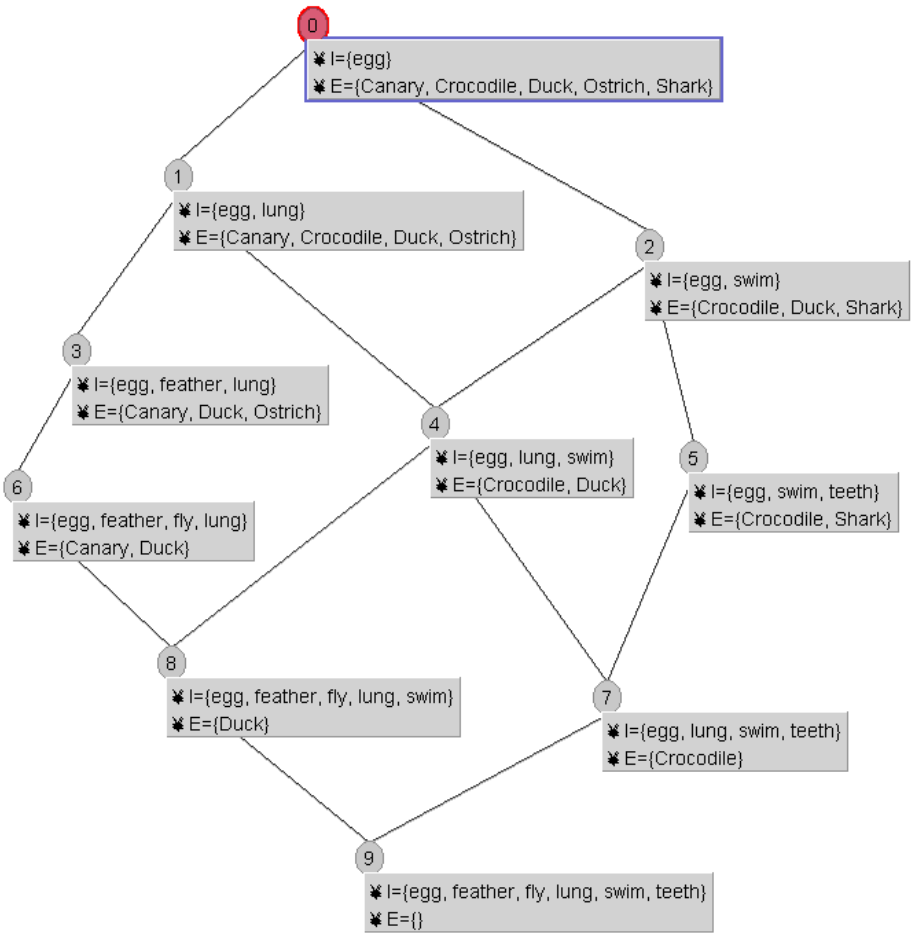


Figure 2.4: Concept lattice constructed from the formal context introduced in Table 2.5. Each node is a formal concept composed of its extent (set of objects) and its intent (set of attributes).

Definition 16 *(A Many-Valued Context)*
A many-valued context is a quadruple (G, M, V, I)*, composed of three sets* G*,* M *and* V *and a relationship* I*.* G *is a set of objects,* M *is a set of many-valued attributes,* V *is a set of attribute values and* I *is a ternary relationship between* G*,* M *and* V*,* $I \subseteq G \times M \times V$*.* $(g, m, v) \in I$*, means that object* g *has value* v *for attribute* m*.*

Scaling Many Valued Contexts

To use FCA with many-valued context, one may transform her/his context into a binary one. This transformation is called conceptual scaling [GW89]. It is achieved by turning each many-valued attribute into several binary attributes. For example, the many-valued attribute "height" could be converted into three binary attributes "short", "medium" and "tall". In some cases, this transformation is not straightforward, and necessitates some choices to be made. Different types of conceptual scaling have been described such as nominal, ordinal and interordinal scaling [GW89].

Let's have an example to illustrate this concept step by step. Table 2.6 presents an example of many-valued context. It describes persons {Adam, Eva, Dora, Zidane, Ali} using the following multi-valued attributes sex, age and height. First, we transfer this multi-valued context into a binary context using conceptual scaling for transferring each multi-valued attributed into binary attributes. For example, the "sex" attribute is transferred into

	sex	age	height
Adam	M	29	tall
Eva	F	22	medium
Dora	F	41	short
Zidane	M	44	tall
Ali	M	32	medium

Table 2.6: Example of a *many-valued* context.

	sex		age			height		
	M	**F**	**<30**	**<40**	**<50**	**short**	**medium**	**tall**
Adam	X		X	X	X			X
Eva		X	X	X	X		X	
Dora		X			X	X		
Zidane	X				X			X
Ali	X			X	X		X	

Table 2.7: The binary context of *many-valued* context presented in Table 2.6 after conceptual scaling.

two binary attributes "M" and "F", which are the possible values of the "sex" attribute. Similarly, the "height" attribute is transferred into three binary attributes "short","medium" and "tall". For the "age" attribute, we use ordinal scaling to transfer it into 3 binary attributes: "< 30" for values lower than 30; "< 40" for values lower than 40; and "< 50" for values lower than 50. Table 2.7 shows the resulting binary context. Figure 2.5 presents the concept lattice generated from this binary context.

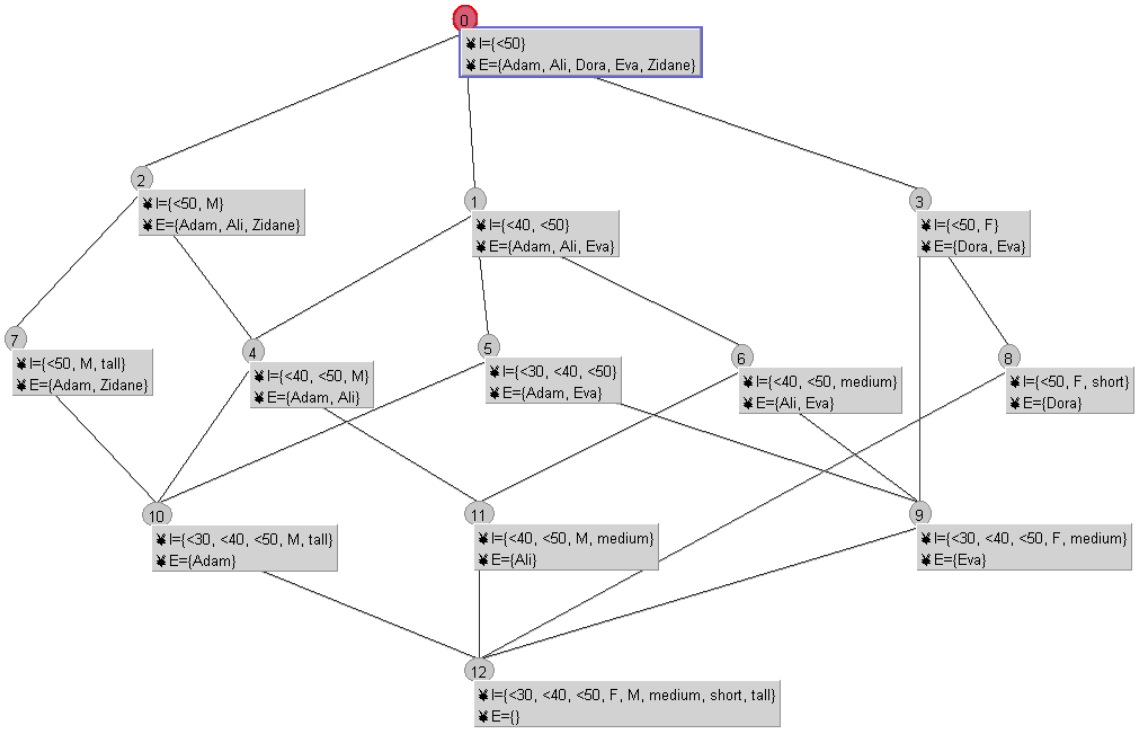


Figure 2.5: Concept lattice constructed from the many-valued context presented in Table 2.6

A conceptual scaling helps to work with complex data (*e.g.,* numerical data, graph data or ontology annotations) by transforming a many-valued context into a binary one. But, the decisions needed for the conceptual scaling could be performed by using an expert from the domain where the data is drawn or defined in supervised settings. Next subsection introduces pattern structures that enable dealing with complex data without using conceptual scaling.

2.3.3 Pattern Structures

FCA is defined using only binary contexts and consequently requires data preparation, such as data scaling, when one deals with more complex data (*i.e.,* that may not be binary). A pattern structure is a triple $(G, (D, \sqcap), \delta)$, where G is the set of objects, D is the domain of descriptions called patterns, $(D, \sqcap)$ is a meet-semilattice of descriptions, and $\delta\colon G \rightarrow D$ maps an object to its description. The derivation operators denoted $(.)^{\square}$ (Galois connection) are defined as following:

$$A^{\square} := \sqcap_{g \in A} \delta(g) \quad for\ A \subseteq G$$

$$d^{\square} := \{g \in G \mid d \sqsubseteq \delta(g)\} \quad for\ d \in D$$

where $A^{\square}$ is the description, which is common to all objects in A and $d^{\square}$ is the set of all objects whose description subsumes d.

Definition 17 *(Pattern Concept)*
A pattern concept of a pattern structure $(G, (D, \sqcap), \delta)$ is defined as a pair (A, d) where $A \subseteq G$ and $d \in D$ such that $A^{\square} = d$ and $d^{\square} = A$, where A is called the pattern extent and d is called the pattern intent. It corresponds to the maximal set of objects A whose description subsumes the description d, where d is the maximal common description of objects in A.

The concept lattice constructed from pattern structures keeps partially ordered relations between the pattern concepts. This lattice is commonly called a pattern concept lattice. Elements of D are partially ordered by a subsumption relation defined as following:

$$c \sqsubseteq d :\Longleftrightarrow c \sqcap d = c$$

$\sqcap$ is called the meet operation, it gives a more general description which stands for the maximal common description of objects of c and d. The meet operator is sometime called similarity since it generates a description that stands for 2 sets of objects and may be seen as the description they all have in common. Next paragraphs provide examples of pattern structures defined over various types of data such as integers, graphs and ontology annotations.

Interval Pattern Structures

Interval pattern structures is a pattern structure that processes the uncertainty in numerical information. A numerical attribute uncertainty is defined in terms of an interval of possible values. So, descriptions of this kind of pattern structure are defined as intervals. To define a semi-lattice operation $\sqcap$ for intervals, let's consider an example. For two intervals $[a_1, b_1]$ and $[a_2, b_2]$, with $a_1, b_1, a_2, b_2 \in R$, their meet is defined as $[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$. This means that the meet of two intervals is the smallest interval containing them. For example, the meet of the following two intervals $[2, 5]$ and $[3, 7]$ is $[min(2, 3), max(5, 7)] = [2, 7]$. Kaytoue *et al.* in [KDKN09, KKND11] successfully applied interval pattern structures to gene expression data analysis for extracting biological situations with similar gene expressions.

Table 2.8 presents an example of context containing 3 objects (g_1, g_2 and g_3) and 3 attributes (m_1, m_2 and m_3) with numerical values. Figures 2.6 and 2.7 show the meet-semilattice and the concept lattice generated from the context example presented in Table 2.8.

	m_1	m_2	m_3
g_1	1	1	2
g_2	2	4	3
g_3	3	4	4

Table 2.8: Example of interval context adapted from [KKND11].

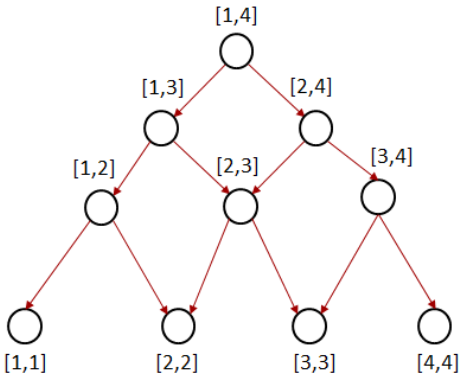


Figure 2.6: The semilattice generated from the attributes of the interval context presented in Table 2.8.

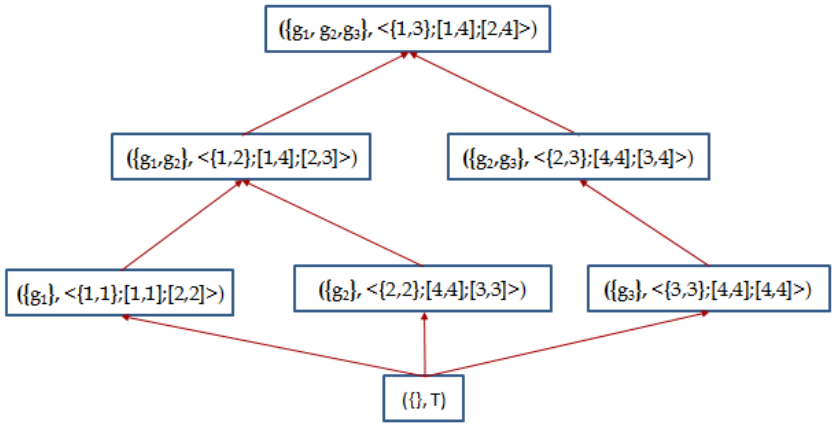


Figure 2.7: The concept lattice generated from Table 2.8.

Graph Pattern Structures

Graph pattern structure has been first introduced by Ganter and Kuznetsov [GK01]. This pattern structure deals with graph data structures such as molecular substructures [GGKS04, KS05]. A pattern structure for graphs is classically defined as $(G, (D, \sqcap), \delta)$, where the semilattice $(D, \sqcap)$ consists of the set of all possible subgraphs from the graph set G, and the meet operation $\sqcap$ that produces from two subgraphs, the smallest subgraph of D that includes the two subgraphs.

Leeuwenberg *et al.* [LBTN15] defined an original pattern structure for syntactic trees[2], named STPS. STPS is used for extracting drug-drug interactions (DDIs) from medical texts where sentences are represented as syntactic trees. A projection and a set of tree-simplification operations have been used to reduce the processing time. STPS uses a similarity operator that is based on rooted tree intersection to compute the similarity between sets of trees. Then, a Lazy Pattern Structure Classification (LPSC) [Kuz13], which is a symbolic classification method, is used to classify the trees.

Ontology Annotations Pattern Structures

Coulet *et al.* [CDKN13] proposed an approach relying on pattern structures to analyze ontology-based annotations of biomedical data. They defined a formal context of pattern structure, where G is a set of drugs and D is the description domain that is composed by a set of ontology classes in the UMLS. Ontology classes in UMLS are ordered according the subsumption hierarchy. Therefore, this order may be used for defining the meet operation. They used the "convex hull" operation to define the meet between two classes in the ontology. For this operation, they first compute the least common ancestor (LCA) between two classes and then consider all classes (including LCA) between the LCA and these two classes. Also, they generalized the "convex hull" operation in a recursive way to find the similarity between any set of ontology classes.

2.4 Conclusion

Data mining is a step in KDD process that aims at extracting hidden knowledge from very large databases by discovering hidden patterns. Different kinds of data can be processed by data mining algorithms such as sequential databases and graph databases. Several data mining approaches have been developed and adapted to these various kind of data. Resulting rules and patterns could be used for information extraction and classification tasks. Data Mining algorithms tend to produce very large number of patterns, which make their analysis is a difficult task and requires a lot of time. Therefore, focusing on a reduced set of interesting patterns (*e.g.,* frequent patterns, rare patterns) or a reduced set of compact patterns (*e.g.,* closed patterns, maximal patterns) is necessary. In this chapter, we introduced different classification approaches that are used for predicting the class of an instance. We categorize these approaches into two main categories: symbolic and numerical approaches. In the literature numerical approaches achieve reasonable results but they are not easy to interpreter in comparison with symbolic approaches. Then, we introduced basics of FCA, which is a mathematical tool for data analysis and knowledge discovery, which may be used as unsupervised classification approach. Also, we explored the case of many-valued context, that FCA cannot handle directly, with *(i)* first, conceptual scaling, used to transform a many-value context into a binary one and; *(ii)* pattern structures, an extension to FCA, used to deal directly with complex data structure such as numerical data, graphs, or ontology annotations.

[2] A tree is an undirected graph, where any two vertices in the tree are connected by exactly one path.

3

Knowledge Discovery from Text and its Biomedical Applications

Contents

3.1 Text Representation: NLP for Text Mining

Texts are written in natural languages (*e.g.,* English, French, Arabic) in the form of sequence of strings, which are in turn sequences of symbols from a given alphabet. These sequences are ordered according to some structure, known as language grammar rules. There are multiple ways to represent a single text, including strings, words, syntactic structures, entity-relation graphs, knowledge predicates, etc. Each representation explores specific features from the text and allows different applications over this representation. This section presents several forms in which a text can be represented. Next subsections explore further applications of text mining.

3.1.1 Lexical Representation

Lexical analysis is the process of converting a sequence of characters (*i.e.,* texts or sentences) into a sequence of tokens (*i.e.,* words). Tools that perform lexical analysis are named lexers or tokenizers. A tokenizer is generally chained with a syntactic and a semantic parser, which respectively analyze the syntax and semantic of a text. They are generally quite simple, where most of the complexity are moved to the syntactic or semantic analysis phases. They serve the basis for any NLP application.

Tokenization

Ex. 3.1.1 *DMD is a genetic disorder characterized by muscle degeneration and weakness.*

Given input text or set of sentences, tokenization splits it into a set of tokens. Tokenization, also known as word segmentation, breaks up the sequence of characters in a text by locating the token boundaries, the points where one token ends and another begins. Tokens could be words, numbers or punctuation marks. Tokens are the basic units for downstream processing. For instance, Figure 3.1 shows the tokenization of the sentence in Example 3.1.1.

Stemming

Stem is the base or the root form of a word. Accordingly, stemming is the task of reducing words to their word stem. Many words may have the same stem. For instance, “plays”, “playing” and “played” are sharing the same stem “play”. Stemming is useful for applications such as information retrieval, where it may help in automatically expanding the search query, by searching all documents that mention one word with the same stem as the query word [APA08].

Lemmatization

Both stemming and lemmatization aim at reducing the inflectional or derivationally forms of words to a common base form. The difference is that the stemming operates on a single word without the knowledge of the context. Therefore, it can not discriminate between words that have different meanings depending on their part of speech. While lemmatization usually does the same task properly with the use of a vocabulary and a morphological analysis of words. For example, the lemma of word “better” is “good” which is missed by stemming, as it requires a dictionary lookup.

3.1.2 Syntactic Representation

In linguistics, the syntax is the set of rules of a language that define how its words may be combined together to build sentences grammatically correct. The components of a sentence are called constituents. Each constituent has a grammatical category such as noun phrase, verb phrase and adjectival phrase, and a grammatical function such as subject, object and predicate.

Syntactic analysis, also known as Parsing, is the process of analyzing a text (string of symbols) conforming to the grammar rules of its written language. A parser is a software tool that takes a text as an input data and generates a data structure representing the text. Different data structures are used to represent the different levels of syntactic information such as sequences (*e.g.,* POS, Phrase Chunking), trees (*e.g.,* parse trees) and graphs (*e.g.,* dependency graphs). Syntactic analysis is the main step in many Natural Language Processing (NLP) applications, including Information Extraction (IE), Opinion Mining, Machine Translation (MT) and Question Answering (QA) [Bru11, Li03].

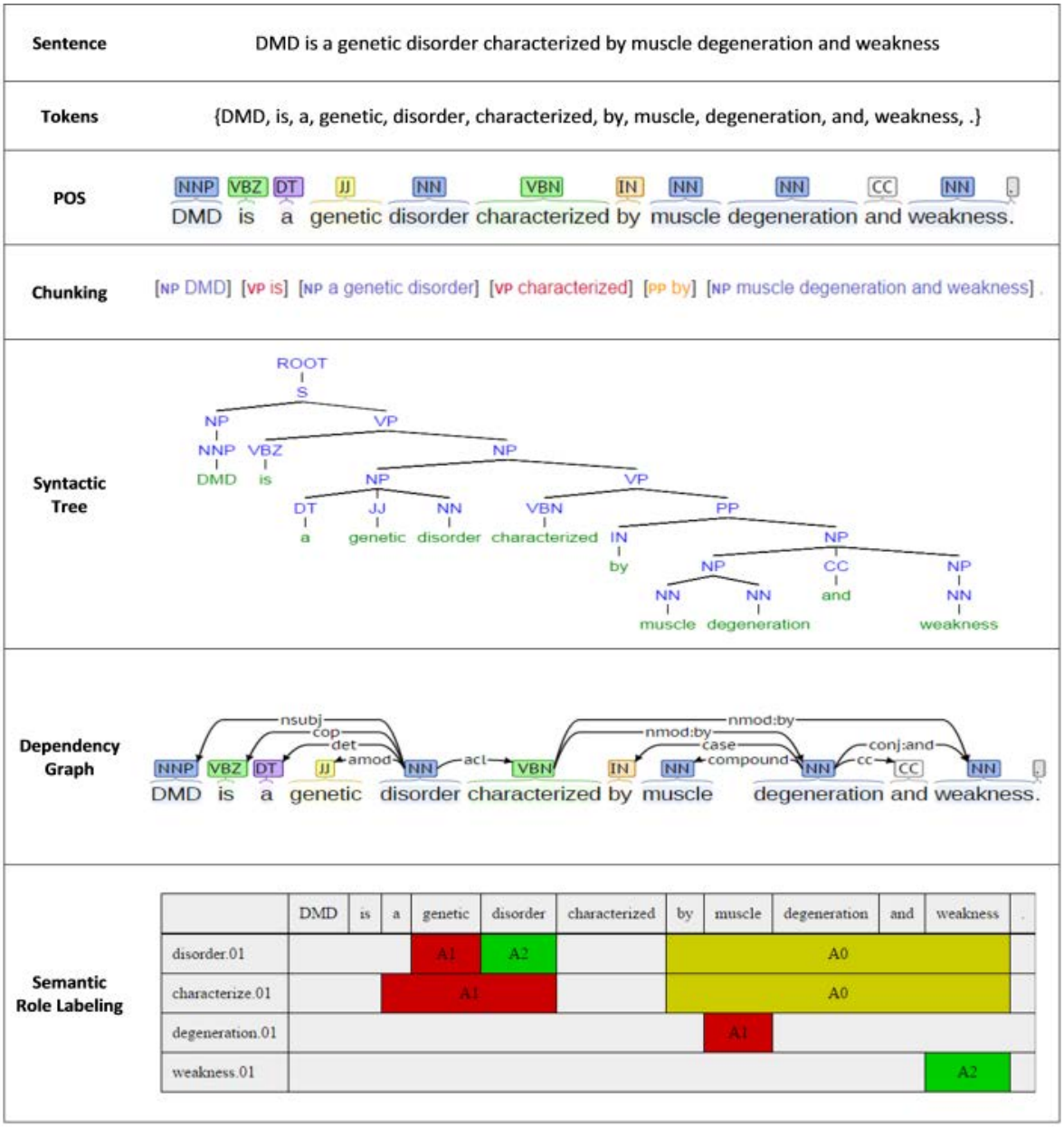


Figure 3.1: This figure shows the different levels of analysis for the sentence in Example 3.1.1.

Sentence Segmentation Sentence segmentation, or sentence boundary identification, is the process of dividing a text into set of sentences. This is usually performed before any syntactic or semantic analysis of the text. By working at the sentence level, downstream extraction tasks (*e.g.,* Tokenization, Named Entity Recognition, part-of-speech tagging, Relationship Extraction) will be easier. It involves identifying sentence boundaries between words in different sentences by identifying the start and the end word of each sentence. Sentences of a natural language are delimited by punctuation marks which help in this sentence segmentation. For example, the period used in English can be detected in text to split sentences. However, punctuation marks may be used for various

purposes, for example a period may also mark an abbreviation (for example in Mr.) or a decimal number. This makes sentence segmentation a non-trivial task.

Part Of Speech Tagging

Part Of Speech (POS) tagging assigns a grammatical category to a word. It can be seen as the classification of a word, in a word or a lexical class, based on its syntactic role within a sentence. In English, the common POS are noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection. A POS tagger is a software that reads text as an input and assigns a part of speech to each word. For instance, Figure 3.1 shows the POS tagging of the sentence in Example 3.1.1. This tagging is achieved by the Stanford POS tagger [MSB+14].

Phrase Chunking

Chunking parsing, also known as shallow parsing, finds all non-recursive noun phrases (NPs) and verb phrases (VPs) in a sentence. It analyzes a sentence by first identifying its constituent parts (*e.g.,* nouns, verbs, adjectives) and then links them to higher order units (*e.g.,* noun groups or phrases, verb groups). An example of chucking parsing of the sentence in Example 3.1.1 is shown in Figure 3.1. In this Figure, NP is a Noun Phrase, VP is a Verb Phrase and PP is a Prepositional Phrase. This chucking parsing is achieved by the Illinois shallow parser [MPRZ99].

Parse Trees

A parse tree, or syntax parse tree, is an ordered, unlabeled rooted tree that represents the syntactic structure of a sentence. Its construction relies on the language grammar rules. It specifies the syntactic structure of a sentence that in turn helps to determine its meaning. Syntax parse tree breaks a sentence into sub-phrases. Figure 3.1 gives the parse tree resulting from parsing the sentence in Example 3.1.1, by using the Stanford parser [MSB+14]. Non-terminals in the tree are types of phrases (*e.g.,* noun phrase, verb phrase, prepositional pharse) while the terminals are the words of the sentence.

Dependency Graphs

Dependency Graph (DG) is a labeled directed binary graph representing dependencies between words of a sentence. Its vertices and edges are representing the words and dependency relations between these words respectively. This graph representation is an interesting alternative to tree representation because it provides an additional level of abstraction over the syntax that is sometime easier to compute. Figure 3.1 presents the DG generated by using the Stanford parser [MSB+14] from the sentence in Example 3.1.1. This DG shows the grammatical relations between pair of words. For instance, the grammatical relation between “DMD” and “disorder” is “nsubj” (*i.e.,* nominal subject).

3.1.3 Semantic Representation

This subsection presents several semantic representations of texts such as conceptual graphs, frame semantics and semantic role labeling. Compositional semantics is another method that represents the meaning of phrases or sentences based on their distributional properties in a large corpus. In this section we detail also the semantic similarities based on compositional semantics as they are useful to find the similarities between complex terms (*e.g.,* phenotypes) based on their semantics.

Conceptual Graphs

Conceptual Graphs (CGs) represent the meaning of sentences in natural language by capturing the semantic relations between words, which was first introduced by Sowa [Sow84]. CGs is rich in semantic and can be used in knowledge representation. However, it is difficult to transform text to conceptual graphs [OSG10]. CGs represents the semantic in unlabeled graph structure. The graph contains two types of nodes which are Concepts and Relations. There is a relation node among each 2 concept nodes to indicate the semantic role of the incident concepts. For instance, the sentence "DMD is characterized by muscle weakness" can be represented by a conceptual graph as shown in Figure 3.2. The rectangles and circles in the graph are Concepts and Relations, respectively. CGs have been applied for a variety of applications, including information retrieval, database design, expert systems and NLP [CHD+07, BS16].

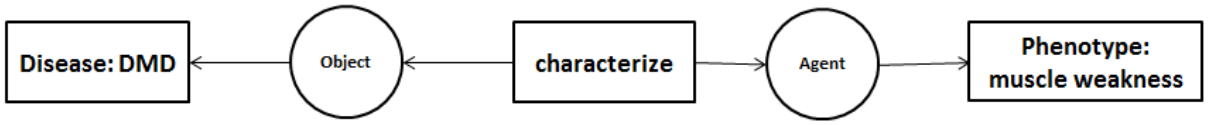


Figure 3.2: An example of Conceptual Graphs

Frame Semantics

Frame Semantics is semantic nets with properties, first introduced by Fillmore [Fil76]. It provides a visual way to see the meanings of words. Its basic idea relies on the following: the meanings of most words can best be understood on the basis of a semantic frame (*i.e.,* the description of an event, relation, or entity and the participants in it). It put together, in frames, the meanings of the elements of a text to form the full meaning of the text. A frame represents an entity as a set of slots (attributes) and associated values. It can represent a specific entry, or a general concept. Frames are implicitly associated with one another when the value of a slot is referring to another frame. Let us consider a typical example to explain frame semantics. Let's take the commercial transaction frame "Buying" as an example (see Figure 3.3). The concept frame is applied to verb mention like "buy". The concept of buying typically involves a person doing the buying (Buyer), the things that are to be bought (Goods), and other optional attributes such as (Seller) and (Price).

FrameNet [FBS02] (or formally the Berkeley FrameNet project) is based on the Frame Semantics theory. FrameNet project builds an online lexical resource for English. It is based on grouping words into semantic classes, called frames. And these frames are similar to those of Fillmore's theory. FrameNet contains four main components: Lexical Units (LU), Frames, a Frame Ontology and the Corpus of sentence examples. The previous "Buying" frame example is represented in FrameNet as a frame called "Buy", and the Buyer, Goods, Seller and Price are called frame elements (FEs). Words that evoke this frame, such as buy are called lexical units (LUs) of the "Buying" frame. The FrameNet defines the frames and annotates sentences to show how the FEs fit syntactically around the word(s) that evokes the frame. FrameNet project contains more than 170,000 manually annotated sentences that provide a training dataset for semantic role labeling. In order to make FrameNet to work in biomedicine, as we needed, it is required to be trained in a medical annotated corpus, which is a tedious task. The purpose of this training step is to build semantic frames for medical events and their entities. FrameNet could be used in applications such as IE, MT, event recognition, sentiment analysis, etc.

Frame Element	Core Type
Buyer	Core
Explanation	Extra-Thematic
Goods	Core
Imposed_purpose	Extra-Thematic
Manner	Peripheral
Means	Peripheral
Money	Peripheral
Period_of_iterations	Extra-Thematic
Place	Peripheral
Purpose	Peripheral
Rate	Peripheral
Recipient	Extra-Thematic
Seller	Peripheral
Time	Peripheral
Unit	Peripheral

Figure 3.3: The "Buying" frame in FrameNet project.

Semantic Role Labeling

Semantic Role Labeling (SRL) is a predicate-argument structure (PAS) parser that represents the sentence structure in a more formal representation schema (*e.g.,* Event, Templates, Frames) [GJ02]. It processes the sentences with the same semantic (but have different syntactic variations) as the same. SRL can be used as a base component for many NLP applications such as: IE, QA, MT, Multiple Documents Summarization, etc. Figure 3.1 shows the SRL parsing that is generated from the sentence in Example 3.1.1 by using mate-tools [BBHN10]. It annotates 4 frames "disorder.01", "characterize.01", "degeneration.0" and "weakness.01". For instance, the frame "characterize.01" is evoked by the verb "characterize" and it has two arguments: *(A1)* "a genetic disorder", which is the object (thing described) and *(A0)* "by progressive muscle degeneration and weakness", which is the subject (the describer).

Semantic Similarities and Compositional Semantics

Semantic similarities are metrics that estimate the semantic closeness between units of language (*e.g.*, words, sentences, documents), concepts or instances. They can be broadly classified into three main categories: ontology-based [BH01, BH06], distributional [Lin98, Tur01] and hybrid similarities [MCS06, XL08]. Ontology-based similarities use ontologies that provide for instance a distance between two concepts, to evaluate the similarity between them [ES13]. On the other hand, distributional similarities use corpora of texts to characterize units of language with various features (*e.g.*, neighboring words and their frequencies) subsequently used to evaluate a similarity between these units. Distributional similarities are usually unsupervised and can be used to compare the relatedness of words expressed in corpora without prior knowledge regarding their meaning. They rely on the *distributional hypothesis* that considers that words occurring in similar contexts tend to be semantically close [Har54,MR01]. Hybrid similarities take advantage of both, text corpora analysis and ontologies to evaluate the similarity.

Semantic space models, which are vector-based models, are based on the distributional hypothesis and aim at representing the semantic of a word by a vector that encodes information about its context (*i.e.*, its distributional semantics) [SWY75, TP10]. Contextual information is typically collected in a frequency matrix, where each row corresponds to a unique word, commonly referred to as "target word" and each column represents a given linguistic context, commonly referred to as "basis element". The semantic similarity between any two target words can be computed by comparing their vectors (*i.e.*, basis elements) using a similarity measure (*e.g.,* cosine, Euclidean distance).

While a semantic space model is used to represent the *semantic* of a word, compositional semantics is used to represent the semantic of a phrase or a sentence that is composed of many words. In compositional semantics, each word participates and contributes in the semantic of the phrase or sentence containing it. Word vectors are composed to create a vector representation of a phrase or a sentence. Different composition methods have been proposed in the literature. Additive and multiplicative composition methods are commonly used [ML08, ML09]. They are rather simple since they propose to represent the semantics of a phrase or a sentence by adding or multiplying the vectors of all words it contains. More sophisticated techniques have been proposed, for instance by Sochet *et al.* [SPH+11, SHMN12]. They propose using machine learning frameworks such as recursive neural network (RAE or Recursive AutoEncoder) to learn the compositional semantics representation of phrases or sentences.

3.2 Biomedical Applications of Text Mining

Text mining aims at extracting patterns and structured information from semi-structured or unstructured textual documents. This enables applications such as information extraction, ontology construction, information retrieval, document classification, clustering, content management or sentiment analysis. This section discusses especially information extraction and ontology construction from texts, which are the main focus of this thesis.

3.2.1 Information Extraction

Information Extraction (IE) aims at automatically extracting structured pieces of information from unstructured or semi-structured textual data [PY13]. IE employs NLP and linguistic-based approaches for processing human language texts (*e.g.,* scientific literature written in English). IE main subtasks include: Named Entity Recognition (NER), Relationship Extraction (RE) and Event Extraction (EE). The following subsections introduce NER and EE, while section 3.3 focuses on RE and its biomedical applications.

Named Entity Recognition

Named Entity Recognition (NER), also known as entity identification or entity extraction, is a subtask of information extraction that aims at locating the named entities in text and classifying them into pre-defined classes (*e.g.,* persons, organizations, locations). Correctly identifying named entities in text is key for a lot of applications such as question answering [TNLM05, DMS06], text summarization [AMT+09, Has03], information retrieval [GXCL09]).

NER task was firstly introduced and defined in the Message Understanding Conferences (1995) as a separated task. Then, it has been integrated as a main NLP component in different applicative contexts. For instance, NER has been used in general domain for identifying general named entities [AM02] and in domain-specific such as bioinformatics for identifying biomedical entities [Set04]. For instance, given the sentence in Example 3.2.1, a general NER could identify "François Hollande" as a Person, "12 August 1954" as Date and "Rouen, France" as a Location. Another example for the recognition of biomedical entities is shown in Figure 3.4. In this example, we

used ABNER [Set05], a biomedical named entity recognizer, for annotating the sentence of Example 3.2.2 with biomedical entities. ABNER annotates "CFTR gene" and "protein kinase A-activated anion channel" as proteins while annotates "immune cell" as cell type, which are related to a rare disease "Cystic fibrosis".

Ex. 3.2.1 *"François Hollande was born 12 August 1954 in Rouen, France."*

Ex. 3.2.2 *"Cystic fibrosis is caused by mutations of CFTR gene, a protein kinase A-activated anion channel, and is associated to a persistent and excessive chronic lung inflammation, suggesting functional alterations of immune cells."*

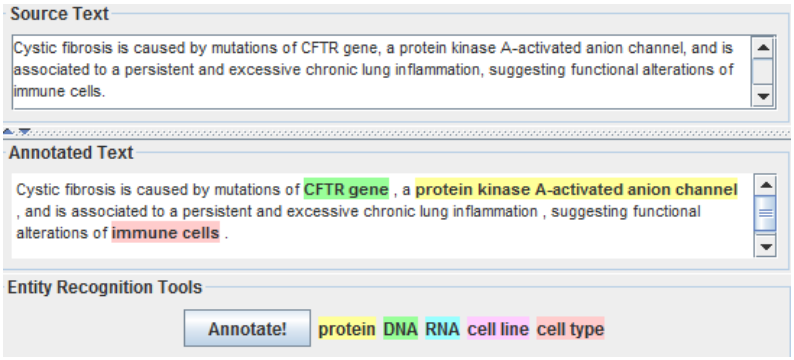


Figure 3.4: This GUI is a part of the ABNER tool and shows its results for recognizing biomedical entities from the sentence of Example 3.2.2.

Various NER approaches are available in the literature. They are based on linguistic techniques as well as rule-based, dictionary-based [QMR+16, TT04] or are using machine learning models such Conditional Random Fields (CRF) [FGM05] or Hidden Markov Model (HMM) [ZS02]). Although, most of the NER approaches based on hand-crafted grammar rules achieve better precision (depending on the corpus and the type of entity), they suffer from a lower recall as well as a cost of time and efforts of involving computational linguistic experts. Supervised ML approaches require a manually annotated large corpus for training an efficient NER. This annotated corpus should be related to the domain application, which requires linguistics or domain experts. Therefore, semi-supervised approaches [LV09, STG15] have been suggested in order to automate and avoid all or part of this manual annotation work.

Event Extraction

Event Extraction (EE) is an IE subtask that identifies complex relationships (*i.e.,* not only binary) between entities. In other words, event extraction aims at extracting several ($\geqslant$ 2) entities and the relationships between them [KOP+09]. Most of EE systems divide the event extraction process into 3 steps [BHG+09, MPHT10]: trigger (or entity) detection, argument detection and event detection and construction. Trigger detection identifies a set of candidate event trigger words. Argument detection identifies the argument mentions that are attached to these triggers. Finally, event detection decides how arguments are shared between events and then the events are constructed.

The automatic extraction of events from scientific documents was the main focus of several BioNLP challenges [KOP+09]. In particular, they focused on extracting biological events that involve genes and proteins, such as gene expression, binding, and regulation events. These events represent the relationships between a trigger and one or more arguments, which can be biomedical entities or other events. Figure 3.5 shows an example of events constructed from the example sentence 3.2.3 with EventMine, an event extraction tool [MTM+12].

First, trigger words "inhibits" and "gene expression" are detected as candidates for events "Negative_regulation" and "Gene_expression". Then, their arguments are detected: (1) "Cause" between "inhibits" and "IL-2 gene"; (2)"Theme" between "gene expression" and "IL-2 gene"; and (3) "Theme" between "inhibits" and "gene expression". Finally, two events are constructed. The first event is a "Gene_expression" event, with the trigger word "gene expression", which has one Theme argument "IL-2 gene". The second event is a "Negative_regulation" event, with the trigger word "inhibits", which has two arguments: First, a Cause argument "IL-2 gene"; Second, a Theme argument that is a reference to the first event "Negative_regulation". As a result, the "Negative_regulation' event is particularly complex because it refers to another event.

Ex. 3.2.3 *"Identification of a zinc finger protein that inhibits IL-2 gene expression."*

```
T0      Protein 54 63   IL-2 gene
T1062   Negative_regulation 45 53
T1063   Gene_expression 64 74
E104    Negative_regulation:T1062 Cause:T0 Theme:E105
E105    Gene_expression:T1063 Theme:T0
```

Figure 3.5: The events extracted from the sentence in Example 3.2.3 with EventMine.

3.2.2 Ontology Construction

An ontology is defined by Gruber 1993 [Gru93] as an "explicit specification of a conceptualization". In general, an ontology describes and represents formally a domain. An ontology consists of a set of classes and the relationships between them. But, ontologies can be associated with different levels of formalism (from controlled vocabularies to formal representations of a domain in description logics). Ontologies are used for different applications such as data sharing [SKSS08], data integration [GGNAM16, MKT+15], reasoning support [WDSS06, HV06].

Ontology, as represented by description logics, consists of two main parts: Terminological Box (TBox) and Assertion Box (ABox). TBox contains classes of a domain and the relationships between them, while ABox contains individuals (data instances) of these domain classes. There are different approaches for ontology building such as top-down and bottom-up [UG96]. The top-down approach is first modeling the classes in the TBox and then modeling the individuals in the ABox. While the bottom-up approach is first modeling the individuals in the ABox and then modeling the classes in the TBox.

The manual construction of an ontology is time-consuming and often error-prone process. Using limited resources in the ontology construction process could result in missing concepts and relationships, as well as difficulty in updating the ontology when domain knowledge changes. Due to the massive growth in the textual and web documents, there is a huge amount of information inside them. Ontology construction from texts aims at utilizing text mining techniques in order to benefit from this information in constructing ontologies and populating them with the instance data. Various ontology construction approaches have been proposed in the literature such as dictionary-based [KL02a, THE00], IE-based [BV02], text clustering-based [AHM01], association rule-based [MS00] and knowledge base driven [AKM+03] approaches. For instance, Blaschke and Valencia [BV02] proposed an automatic method that employs an IE technique for generating classifications of gene-product functions using bibliographic information. These classifications have the same structure of the ones constructed by human experts. Also, texts could be used for enriching existing ontologies [BML+13,FS02]. Booshehri *et al.* [BML+13] proposed a RE-based approach for enriching an existing ontology by extracting some hidden assertional knowledge from

text. Also, they provided an algorithm that uses large knowledge repositories to enrich the non-taxonomic relations in the ontologies extracted from texts.

3.2.3 Other Applications

Text mining has many other applications than IE and ontology construction. These applications usually introduce the notation of documents instead of texts.

Information Retrieval

Information Retrieval (IR) is the task of finding information resources from a collection or a corpus of documents [MRS08]. IR enables users to query a set of documents in order to retrieve those that match the query. A well-known example of IR system is the Google search engine, which enables retrieving web documents according to a simple user query. The query can be a textual research and can include Boolean operators. General applications of IR system include Digital Library [Sch97], Search Engines [CMS09] and Multimedia Search [ML02]. Efficient IR systems transform documents into a suitable representation (*e.g.,* binary model, vector space model). Also, various indexing methods has been proposed for reducing the search space and speeding the retrieval time (*e.g.,* signature file, inversion indices [CC05], lattice structure [CLN14]). In [CLN14], Codocedo *et al.* used a concept lattice, computed by FCA, both as a semantic index to organize documents and as a search space for terms. They provide a classification-based reasoning algorithm for navigating the concept lattice and retrieve relevant information.

Document Classification

Document classification aims at assigning a document into one or more predefined classes according to its content. Classification algorithms use features (*e.g.,* bag-of-words, latent semantic) of documents in order to determine the "class" they are belonging to. Machine learning and pattern recognition have been widely used for automatic document classification [Seb02]. This process can be divided in two main categories: supervised [FCT04, Joa98] and semi-supervised [ZHY+16, SSM11]. Supervised document classification requires labeled data for learning, while semi-supervised document classification can use both labeled and unlabeled data. For instance, Zhao *et al.* [ZHY+16] proposed a semi-supervised approach that uses Multinomial Naive Bayes with Expectation Maximization (MNB-EM) for text classification. MNB assumes that word occurrences are conditionally independent of each other given the class of the document. Their approach increases MNB-EM by leveraging the word level statistical constraint to maintain the class distribution on words. MNB-EM leverages both labeled data and unlabeled data. It maximizes the joint likelihood of labeled data and the marginal likelihood of unlabeled data. Their approach outperforms state-of-art baselines.

Clustering and Topic Extraction

Document clustering is the process of grouping a set of documents in a way that documents in the same group, called a cluster, are more similar to each other and less similar to those in other clusters [BSJ15, KLR+04]. Clustering algorithms are unsupervised learning techniques. Hu *et al.* [HZGH08] proposed a constrained K-means based approach for document clustering. They use prior knowledge (*e.g.,* two given documents belonging to the same cluster) as constraints for the clustering process. They integrate the constraints into the formulation of the sum of square of the Euclidean distance function of K-means. Euclidean distance is used to measure the distances between documents. K-means tends to assign two similar documents (*i.e.,* with a small distance between them) into the same cluster.

Text Summarization

Text summarization is the task of reducing a text into a shorter one or into a structure such as a graph that synthesizes the content of the text. While a summary should be short, it should also be coherent and retains the most important information of the original text. A text can be summarized based on different views such as extraction, abstraction, fusion and compression [RHM02]. Automatic summarization is mainly based on NLP techniques and machine learning methods (*e.g.,* Naive-Bayes [KPC95], Decision Trees [Lin99], HMM [CO01], SVM [FAR07]). For instance, Fuentes *et al.* [FAR07] used SVM for text summarization. SVM ranks sentences in order of relevance to a user query.

Sentiment Analysis

Sentiment analysis, also known as opinion mining, is the process of identifying and categorizing the opinions expressed in a piece of text [PL08]. It employs NLP and computational linguistics to identify and extract subjective information from the source texts [Liu10]. It is widely used in marketing, customer service and social media in order to determine the user/customer attitude (*e.g.,* positive, negative) toward a specific service, product or topic. Sentiment analysis approaches can be categorized into three main categories: knowledge-based [TBT$^+$11, SK11, CSL$^+$13], statistical [MVGaN13, BRO13] and hybrid approaches [BFANP14]. Balage *et al.* [BFANP14] proposed a hybrid classification approach that uses three classification methods: rule-based, lexicon-based and machine learning. This approach achieved an improvement in the F-measure (+ 9.08% of improvement) in the Twitter message-level subtask for 2013 dataset in *Semeval-2014 Task 9: Sentiment Analysis in Twitter* [RRNS14]. Given a Twitter message, this task aims at classifying whether the message is of positive, negative, or neutral sentiment.

3.3 Focus on Relationship Extraction and Its Biomedical Applications

This section gives a special focus on Relationship Extraction (RE). First, it describes the task of RE, then it details several RE methods and their applications in the biomedical domain.

RE is a text mining task that aims at extracting automatically the occurrences of relationships mentioned in text between several entities. For example, RE may be used to extract binary relationships between interacting proteins [BMRM06, MSMT09] or interacting drugs [YLL10, KLYW15]. The RE process is usually divided in two main steps: NER and Relationship Identification (RI). Figure 3.6 illustrates the process of RE with an example of relations between a disease and a phenotype. First, Named Entity Recognition (NER) identifies the interesting entities in the text and annotate them with the corrected category. The second step checks if the named entities are involved in a relationship or not, and may qualify the type of the relationship.

Ex. 3.3.1 *[From PMID:20972738] "Wolfram syndrome is a rare hereditary disease characterized by diabetes mellitus and optic atrophy. The outcome of this disease is always poor. WFS1 gene mutation is the main cause of this disease."*

Intra-sentential vs. Inter-sentential RE: In natural language texts, entities for which a relationship is holding may be found within a single sentence or over many sentences. So far, most of the RE works have focused on sentence-level relationship extraction, which is known as intra-sentential relationships, *i.e.* relationships holding between named entities of the same sentence. For instance, the first sentence of Example 3.3.1 contains two intra-sentential relationships between a disease and two phenotypes: ("Wolfram syndrome", "diabetes mellitus"), and ("Wolfram syndrome","optic atrophy"). Alternatively, other works proposed inter-sentential relationship extraction, also known as cross-sentential RE, to extract cross-sentential relationships (*i.e.* relationships between named entities beyond sentence boundaries and can be asserted over many sentences) [SS11, SS10, Ste06].

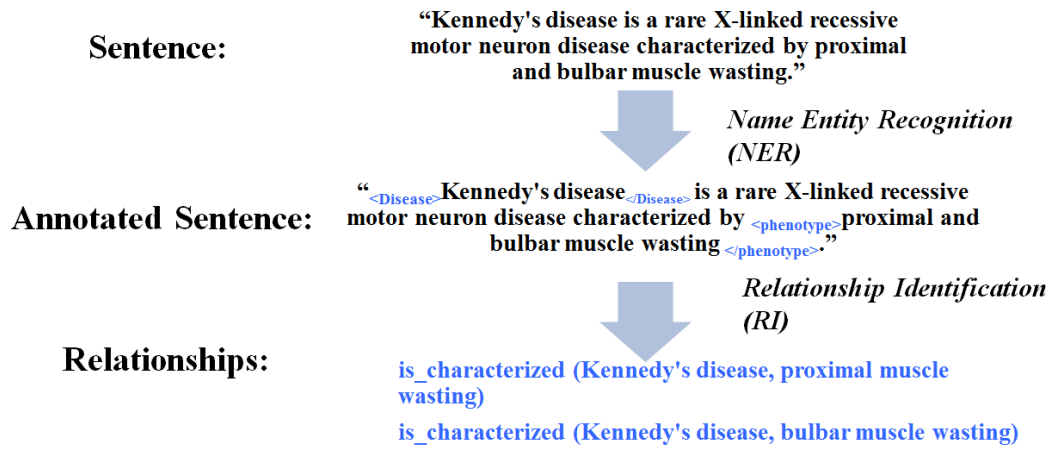


Figure 3.6: The process of Relation Extraction (RE)

Coreference resolution is a supporting task for inter-sentential RE. Example 3.3.1 shows an Inter-sentential relationship between "Wolfram syndrome" (in the first sentence) and "WFS1 gene mutation" (in the third sentence), which is difficult to extract but coreference resolution could help by finding that the expression "this disease" refers to 'Wolfram syndrome'". A biomedical coreference resolution for proteins coreference task was organized in the BioNLP Shared Task 2011 [NKT11]. Its results show that best method finds 22.18% of protein coreferences with the precision of 73.26%. The adaptation of a coreference resolution developed for one domain to another domain is not a straightforward task (such as from one kind of entity to another kind of entity). Because of the difficulty of inter-sentential RE and the complexity of biomedical texts, most of RE methods proposed in biomedicine are designed for extracting relationships at the sentence level. These are not the only reasons; also the motivation behind this choice is the vast majority of the relationships involves entities appearing both in the same sentence. For example, Swampillai and Stevenson (2010) in [SS10] reported that 90.6% of the total number of relationship in the ACE031 corpus (a benchmark news domain RE corpus) are intra-sentential. According to these reasons and similarly to most of the previous RE works, we focus in this thesis on extracting intra-sentential relationships from biomedical literature and leave the adaptation of our work to cross-sentential RE for future work.

Binary vs. n-ary RE: A binary RE system, is a simple RE, which extracts the relationships between pairs of entities, while a n-ary RE system focuses on relationships between more than two entities (*i.e.,* $n > 2$) [MPK+05]. A binary relationship can be represented simply as a pair of entities by the following tuple schema (e_1, e_2), where e_1 and e_2 are two named entities in an untyped relationship. For example, the following relation tuple ("Kennedy disease","dysarthria") represents a relationship between two named entities a disease "Kennedy disease" and a phenotype "dysarthria". In this kind of representation, the type and the direction of relationships are not represented. Extensions of this representation are possible. The order of the pair can indeed represent the direction. And, tuples can be associated through a specific relationship type. A relationship can also be represented in a form that is similar to first-order-logic predicates, of the form $predicate_x(e_1, e_2)$. For instance, HasPhenotype("Kennedy disease", "dysarthria"), which means that the relationship linking "Kennedy disease" and "dysarthria" is of the type "HasPhenotype", *i.e.,* "Kennedy disease" has phenotype "dysarthria". This representation may be even enriched by stating that e_1 should be of type 'Disease' and e_2 should be of type 'Phenotype'. Changing the order in the tuple into HasPhenotype("dysarthria", "Kennedy disease") produces a different interpretation ("dysarthria" has "Kennedy disease"), which is wrong here. Another change, PhenotypeOf("dysarthria", "Kennedy disease") means that "dysarthria" is a phenotype of "Kennedy disease",

which is a correct interpretation if the relationship type "PhenotypeOf" is the inverse of "HasPhenotype".

N-ary relationships may be represented as a larger tuple $(e1, ..., e_n)$ where $e_i \in E$ are named entities. For instance, assume that the types of named entities are "person", "place", "date" and we are interested in the ternary relation with schema (person, place, date) which means that a person was born in a place on a specific date. For example, instance tuple ("François Hollande", "Rouen, France", "August 12, 1954") means that "François Hollande" was born in "Rouen, France" on "August 12, 1954". Similarly to binary relationships, this representation can be typed, with a term such as in birth ("François Hollande", "Rouen, France", "August 12, 1954").

Several works have been proposed for extracting n-ary relations from text such as [BBDR17, ZZH14, BHG+09]. For example, Berrahou et al. [BBDR17] developed a system, named Xart, for extracting n-ary relations from text. Xart is based on a hybrid method that uses a pattern mining method and syntactic analysis. It relies on a domain ontology for discovering sequential patterns to identify arguments involved in the n-ary relations. Sequential patterns use an ontological resource and specific syntactic relations to build ontological linguistic sequential patterns (OLSPs). OLSPs are then used for extracting the arguments of n-ary relations and for populating an ontological resource with them.

For the sake of simplicity, most of the works in RE are focusing on binary RE. This is also the case of this thesis. The following details the main methods for binary RE and illustrates them with biomedical applications. We categorize these methods into four categories: *(1)* co-occurrence, *(2)* rules-based, *(3)* machine learning and *(4)* multiple classifier systems.

3.3.1 Co-occurrence Methods

Co-occurrence methods are based on the hypothesis that if two entities are mentioned frequently together in text portions (*e.g.*, sentence, paragraph), they are likely to be related [BMRM06]. Because two entities might co-occur together by chance, statistic tools may be used to reduce this bias. some of these tools enable to hypothesize about the existence of a relationship between the entities, such as pointwise mutual information (PMI), Chi-square or log-likelihood ratio [MS99].

Ramani *et al.* used a random co-citation model based on the hypergeometric distribution to identify protein-protein interactions [RBMM05]. Hypergeometric distribution is a discrete probability distribution. It is dissimilar to the binomial distribution since it describes the probability of k successes in n draws, without replacement, from a finite population of size N that contains exactly K successes. In [RBMM05], the hypergeometric distribution is used to calculate the significance of co-citation of two protein names across a set of MEDLINE abstracts (see equations 3.1 and 3.2). They extracted 31,609 interactions among 3,737 human proteins from 6,580 MEDLINE abstracts.

$$p(\text{\#of co-citing abstracts}) \geq l \mid n, m, N) = 1 - \sum \sum_{k-0}^{l-1} p(k \mid n, m, N) \tag{3.1}$$

$$p(k \mid n, m, N) = \frac{\binom{n}{k}\binom{N-n}{m-k}}{\binom{N}{m}} \tag{3.2}$$

where N is the total number of abstracts, n is the number of abstracts that cite the first protein, m is the number of abstracts that cite the second protein, and l is the number of abstracts citing both.

A drawback of co-occurrence RE is that they are not capturing the type and the direction of relationships. For this reason, these methods are not used when fine-grained relationships are required. However, they have been successfully applied to automate the construction of biomolecular networks such as protein–protein, gene–protein and gene regulatory networks, where the type and direction of relationships can be ignored [vJO+06, FKY+01].

Co-occurrence approaches require a large corpus to perform well, but they are simple and relatively fast to compute (they do not require complex linguistic analysis). They are frequently used as a baseline method to compare to, particularly when no proper benchmark is available for comparison.

Co-occurrence tends to achieve high recall as they consider all possible mentioned pairs in text. But, it usually suffers from a lack of precision because of the complexity of sentences in the biomedical domain. For instance, a sentence may contain numerous entities, while only a few of them are indeed mentioned as related. Example 3.3.2 shows such a complex sentence where 3 diseases "glycogenosis type II", "Duchenne's muscular dystrophy" and "mitochondrial myopathy" and 7 phenotypes "tired", "headaches", "hypercapnia", "dyspnoea", "hypercapnia", "dyspnoea", "headache" have been annotated by MetaMap [AL10]. A naïve co-occurrence approach will consider 21 disease-phenotype relationships, while only 7 are true and 14 are false.

Ex. 3.3.2

[from PMID:10908953] "Three patients had <disease>chronic respiratory disorders</disease>: a 42-year-old man with <disease>glycogenosis type II</disease> was <phenotype>tired</phenotype>, had <phenotype>headaches</phenotype>, poor pulmonary function values and, according to the arterial blood gas values, <phenotype>hypercapnia</phenotype>; a man aged 24 with <disease>Duchenne's muscular dystrophy</disease> had variable moderate <phenotype>dyspnoea</phenotype> with hypoxia and <phenotype>hypercapnia</phenotype>, and a man aged 64 years with an <disease>mitochondrial myopathy</disease> complained of <phenotype>dyspnoea</phenotype> and <phenotype>headache</phenotype> but had good blood gas values."

3.3.2 Rule-based Methods

Rule- and pattern methods are RE methods consisting in defining symbolic rules, or more generally, patterns over the linguistic and syntactic content of text [AG00]. Rule-based methods are similar to pattern-based methods and belong to the same category. Rule-based methods also use patterns to express the rules. But, they extend the patterns by adding constraints to express more complex situations of extraction.

Rules (and patterns) are compared to portions of text (or representation of text), and when a match is found, a relationship is extracted. Rule- (and pattern) based methods are referred as symbolic methods because they use a symbolic representation to represent the rules (and patterns) and their results. One advantage of rules is that their definition (*i.e.,* the rule itself) is easy to interpret, in regards to ML approaches for RE [DA05].

Rules (and patterns) may either be *manually defined* by an expert or *automatically defined* by a preliminary learning phase. The manual approach usually requires lots of time and efforts from experts and tends to achieve a high precision because of restrictive constraints defined by experts, but a low recall because it fails at covering uncommon forms of relationships [DA05]. The automatic approach usually enables to increase slightly the recall compared to the manual approach because it includes a larger number of patterns by covering more systematically forms of relationships in a large corpus [HPL$^+$05]. A classical automatic approach for learning patterns from text is to search for regularities in sentence syntax within an annotated corpus.

The syntax for defining rules (or patterns) varies, consequently enabling to represent different levels of constraints. Both lexical and syntactic elements may be used, likewise different levels of structures like sequences of words, syntactic trees and DGs. For example, Béchet *et al.* and Cellier *et al.* proposed methods based on sequential pattern mining to extract disease–gene and gene–gene relationships [BCC$^+$12, CCP10]. As the number of patterns they learn is very large, they introduced constraints for filtering them. For instance, Béchet *et al.* [BCC$^+$12] used constraints such as $min\text{-}sup$, the minimal length of pattern, the membership (containing one gene, one RD, and one noun or one verb) and possible gaps (the allowed number of words between two items of pattern) to reduce the number of extracted patterns. Example 3.3.3 presents a pattern instance taken from [BCC$^+$12]. This pattern combines the lemma and category levels. This pattern extracts disease–gene relationships from sentences matching with its sequence of items (an item consists of a POS and/or a lemma). The results show that the best $F\text{-}measure$ (0.65) is obtained when considering only the $min\text{-}sup$ constraint with value 0.05.

Similarly to [BCC+12, CCP10], Martin *et al.* also used sequential patterns, but for recognizing unidentified symptoms [MBC14]. Example 3.3.4 presents an example pattern they consider for extracting symptoms from texts. They do this in an iterative way, where they learn patterns and then they extract symptoms and this process is repeated. This method is applied to 25 abstracts were randomly selected and achieved $F\text{-}measure$ of 36.8 (23.7 in recall and 82.2 in precision). Fundel *et al.* developed an approach for RE from text called RelEx, in which they manually defined a small number of simple rules over DG [FKZ07]. They successfully extracted $150,000$ gene–protein relationships from one million MEDLINE abstracts with a precision of 0.80 and a precision of 0.80. Similarly, Coulet *et al.* defined a more complex set of rules over DG to extract gene–drug and drug–disease relationships from text with a high precision (0.87) [CSG+10]. Liu *et al.* proposed a method to learn rules for RE from DG [LVC+13]. Their rules are defined using the shortest path between two entities in a DG. One advantage of using the shortest paths rather than the whole DG is that they are easier to implement and to compute, but also contain the most important information to qualify the relationship. Adolphs *et al.* developed an algorithm to learn general graph rules from a set of DG [AXLU11]. First, subgraphs are extracted, then a subgraph generalization is performed by underspecifying the nodes, to finally generate rules. The generalization enables to produce a reduced set of compact rules.

Ex. 3.3.3 *(mutation NNS) (IN) (isocitrate NN) (GENE) (occur V BP) (DISEASE)*

Ex. 3.3.4 *(patient)(have)(severe)(SYMPTOM)*

3.3.3 Machine Learning Methods

RE can be considered as a classification problem and then be treated with Machine Learning (ML) methods. In simpler cases, ML algorithms classify a relationship between entities either as true or false. ML methods compute their classification on the basis of a set of features. Given a set of relationship instances and their feature vectors, ML methods train a model (or classifier) used subsequently for the classification task. Support Vector Machines (SVM) and Conditional Random Fields (CRF) are popular examples of ML methods that have been employed for RE task [KLRPV08, BDS+08].

Building a feature matrix

For training, ML methods require a feature matrix where each relationship instance is represented by a feature vector. The efficiency of a ML method depends consequently on the features considered and encoded in these vectors. Features can be bags-of-words (BOW) features, part-of-speech (POS) features, syntax tree or DG features, *e.g.,* the shortest path between two entities in a DG, the complete DG, or walk features such as e-walks and v-walks. In DGs where nodes and edges are respectively words and grammatical dependencies between words, a v-walk feature consists of a path between two vertices, whereas an e-walk feature is a path between two edges [KYY08, CL12].

Selecting the best features

Most ML methods use a combination of different types of features for improving the efficiency of the classifier. However, using a large number of features usually necessitates additional computation time. Hence, *feature selection* methods have been proposed for selecting the most informative set of features to use in a classification process [GE03, SIL07]. Hall proposed a correlation-based feature selection method, which evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them [Hal99]. It enables to select subsets of features that are highly correlated with the class while having low inter-correlation.

Computing a classifier model from a feature matrix

Various ML approaches have been proposed to compute/construct a ML classifier using a feature matrix (set of feature vectors). The methods vary from Bayes methods (*e.g.,* Naïve Bayes [JL95]), K-nearest neighbours methods (*e.g.,* Lazy IBK [AKA91]), Decision Trees methods (*e.g.,* Trees J48 [Qui93], RandomTree, RandomForest [Bre01]), Support Vector Machine (SVM) and kernel methods (*e.g.,* LibSVM [CL11]). Kernel methods are a family of ML algorithms that enable to work directly with data that have complex structural representations such as trees or graphs [ZAR03, ZZA08, APB+08]. Classical methods usually require formatting data in the form of feature vectors. Differently, kernel methods use a similarity function, named kernel function, to compare directly these complex data structures. The state of the art RE methods is based on kernels [CL12]. The following gives more details about several kernel methods for RE.

Subsequence kernels Bunescu and Mooney [BM06] proposed a subsequence kernel method using sequences containing words and word classes. The subsequence kernels are applied for extracting protein interactions from biomedical corpora (AImed and LLL) and top-level relations from newspapers corpora.

Shallow linguistic kernels Shallow linguistic kernels capture the syntactic structures or semantic meanings that could be helpful for the discovery of relations in unstructured texts. They were first proposed by Giuliano *et al.* [GLR06] for gene and protein interactions. They use a combination of kernel functions to represent two distinct information sources: the whole sentence (the global context) where the entities appear and windows of limited size around the entities (local contexts). Segura-Bedmar *et al.* [SBMdPS11] also used a shallow linguistic kernel for the automatic extraction of drug–drug interactions (DDIs) from biomedical texts.

Tree and graph kernels Some works have been proposed to use the tree and graph structures for extracting relations between entities. Zhang *et al.* proposed a kernel approach that uses the syntactic tree representation of sentences for RE. They studied how to capture the syntactic structure by using a convolution tree kernel and support vector machines [ZZA08]. Zelenko *et al.* also proposed a tree kernel method, but using shallow parse tree representations [ZAR03]. The same tree kernel approach has been used by Culotta and Sorensen, but allowed feature weighting and used additional features such as Wordnet, POS, entity types [CS04]. In both approaches, a relation instance is defined by the smallest subtree in the parse or dependency tree that includes interesting entities. Graph kernels [VSKB10] use graph kernel function to measure the similarity between graphs. The idea of constructing kernels on graphs (*i.e.,* between the nodes of a single graph) was first proposed by Kondor and Lafferty [KL02b], and then extended by Smola and Kondor [SK03]. Both tree and graph kernels approaches show reasonable results but they are still hard to implement and computationally complex.

All-paths graph kernels Gärtner [GFW03] proposed graph kernels based on the label sequences of all possible walks in the kernel. Extension to Gärtner, Airola *et al.* [APB+08] used the all-paths graph kernel for the extraction of PPIs. This kernel has the capability to make use of full, general DGs representing the sentence structure. It considers all possible paths connecting any two vertices (containing two interesting entities) in the resulting graph.

Shortest path kernels Computing a complete graph kernel (the same with tree kernels, where a tree is considered as a specific type of graph) is hard as deciding whether two graphs are isomorphic, and that the problem of computing a graph kernel based on common (isomorphic) subgraphs is NP-hard problem. Also, considering all paths is NP-hard. Bunescu and Mooney [BM05] proposed a shortest path kernel method that uses the shortest path between two entities in an undirected DG for relationship extraction. This work is based on the hypothesis that the relationship between two entities in the same sentence is typically captured by the shortest path between them in

the DG. Borgwardt *et al.* [BK05] used a shortest-path kernel for proteins classification. This shortest path kernel is simply a walk kernel run on a Floyd-transformed graph. Floyd-transformation transforms the original graphs into shortest-paths graphs.

3.3.4 Multiple Classifier Systems

Multiple classifier systems (MCS) are a broad category of approaches that combine several classifiers to improve the final results of a classification task.

Ensemble methods are a subset of MCS in which distinct classifiers are based on the same model, such as Random Forest that combines results of multiple decision trees [BK99,Die00]. Various kinds of Ensemble methods have been proposed such as *Bootstrap aggregating* [Bre96a] in which various classifiers are built on only a subset of features that are randomly selected; *Boosting* [Sch01] that relies on the idea that a classifier benefits of being used iteratively, if its next iteration focus more on previously misclassified examples than on correctly classified ones; and *Stacking* that uses a last classifier to consider and combine classification output of firstly ran classifiers [Bre96b, STML09].

Multiple kernel learning aims at combining various kernel functions to enable considering different representations of the same example. For example, Chowdhury *et al.* [CL12] proposed a multiple kernel that uses different types of information (*e.g.,* syntactic, contextual, semantic) and their different representations (*i.e.,* flat features, tree structures and graphs). Their method combines two vector-based kernels and a tree kernel. They applied it for extracting protein-protein interaction (PPI) and outperformed state-of-art approaches.

More generally, various ML methods have been successfully combined within hybrid systems to extract relationships. Huang *et al.* [HZL06] presented a hybrid approach that integrates a shallow parser and pattern matching algorithm to extract PPI from biomedical texts. Their method showed a 7% improvement of both precision and F-measure (see subsection 3.4 for definitions of precision and F-measure). Song *et al.* [SHK+14] proposed a hybrid approach for extracting PPI that combines a rule-based approach and a classification algorithm (alternatively SVM, Naïve Bayes and Decision Tree) for RE. They obtained better or equivalent performances compared to the state-of-the-art methods.

We propose in this thesis an original hybrid approach that combines a pattern-based method, named SPARE, and a classical ML algorithm: SVM. Results of both methods are combined with the hope of benefiting from the relatively high precision of SPARE and of the good recall of SVM. In the whole approach, we considered linguistic and syntactic features of text to facilitate RE. Next chapters describe this approach and its application to the extraction of RD-Phenotype relationships.

Deep learning refers to a family of machine learning methods that are based on Neural Networks that aims at learning features instead of using handcrafted features. Promising and new works used deep learning for RE from texts [LTCW16, LGYW16, LJ16, NG15]. For example, Liu *et al.* [LTCW16] used convolutional neural networks (CNN), which is a deep learning technique, for extracting DDIs. Their method achieved $F\text{-}measure$ of 69.75% on the 2013 DDI Extraction challenge corpus, which is higher than the best state-of-the-art method by 2.75%. In this thesis, we do not use deep learning but we believe it would be a good extension to our RE method that may improve performances.

3.4 Metrics for Evaluating a Text Mining Approach

It is useful to evaluate text mining approaches in order to show that an approach is efficient and also to compare its performance to other approaches. Existing metrics evaluate different characteristics of an approach induced by the algorithm. In this section we explore the most common measures used for binary RE and binary classification tasks, which are our main focus in this thesis. These two tasks can be evaluated in a similar way, as RE task can be

seen as the classification of candidate relationship as either true or false relationships. Examples of these metrics include $Precision$, $Recall$, $F\text{-}Measure$, $ROC\ curves$, $PR\ curves$ and $AUC\ ROC$.

3.4.1 Precision, Recall and F-Measure

Confusion Matrix

Confusion Matrices or contingency tables are the basis of the evaluation of binary classification tasks. It enables to distinguish four categories of instances, depending on their actual class (Positive or Negative) and the predicted label assigned to them by the classifier. These four categories, shown in Table 3.4.1, are True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). For example, a True Positive is a positive instance that is classified as positive; a False Positive is a negative instance classified as positive.

		Classification	
		True	False
Actual	Positive	TP	FN
	Negative	FP	TN

Table 3.1: The confusion matrix of a binary classifier, where rows present the actual class of instances and columns present the label assigned to instances by the classifier. Cells of the matrix represent 4 categories of instances: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN).

Precision

Precision, also known as positive predictive value, presents the fraction of instances labeled as positive and that are actually positive (*i.e.,* TP) on the set of all instances classified as positive (*i.e.,* $TP \cap FP$). It can be seen as the probability of having truly a positive instance, knowing that one classifier classified this instance as positive: $P(Y = 1|\hat{Y} = 1)$ where Y is the actual value and $\hat{Y}$ is the predicted value. Equation 3.3 gives the formula for computing the precision of a binary classifier.

$$Precision = \frac{|TP|}{|TP \cup FP|} \tag{3.3}$$

Recall

Recall, also known as sensitivity or true positive rate (TPR), presents the fraction of positive instances that are actually classified as positive. It is the probability of classifying a positive instance as a positive: $P(\hat{Y} = 1|Y = 1)$. Equation 3.4 gives the formula for computing the recall of a binary classifier.

$$Recall = \frac{|TP|}{|TP \cup FN|} \tag{3.4}$$

The precision provides a measure of how correct is the set of the result provided by the classifier, whereas the recall measures how much it is complete regarding the considered set of instances.

F-Measure

The F-Measure is the harmonic mean of the precision and recall. F-measure balances the precision and recall values, where the best F-Measure value achieves the best combination of both precision and recall. Equation 3.5

gives the formula of the $F\text{-}Measure$.

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.5}$$

3.4.2 ROC and AUC ROC

A receiver operating characteristic (ROC) curve, or ROC curve, is a graphical representation that illustrates the performance of a classifier system as its discrimination threshold is changed [Faw06]. The curve is created by plotting the false positive rate (FPR) at the x-axis against the true positive rate (TPR) at the y-axis using various threshold settings. The threshold could rely on the probability that classifier gives to an instance to be classified as positive or negative. For example, if threshold $= 0.5$ and the probability computed by the classifier for an instance is 0.6. Therefore, this instance is labeled as positive because the probability of being positive is greater than the threshold. When the threshold changes, the result of the classifier changes. For example, if the threshold is change to 0.7, then the instance is labeled as negative instance.

The ROC space, defined by FPR and TPR as x and y axes respectively, describes the relative trade-offs between the true positive (benefits) and the false positive (costs). Each point in the ROC space represents a prediction result of a confusion matrix at a specific threshold. A binary classifier usually assigns a real value as a classification result for an instance. Using specific threshold, the classifier labels this instance as a positive or a negative instance. Selecting multiple thresholds generates multiple confusion matrices (one confusion matrix for each threshold) and consequently generates multiple points in the ROC curve.

If $F\text{-}Measure$ shows the performance of one classifier, it may be insufficient to compare different classifiers. For example, if one classifier has higher precision but lower recall than other, how can you decide which classifier is better. At one threshold the first classifier will give the best $F\text{-}Measure$, while at another threshold the other classifier will give the best $F\text{-}Measure$. ROC can be used to compare the performance of different classifiers. It shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples when the threshold varies. The best possible classifier would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). Figure 3.7 shows ROC curves of 3 different classifiers. ROC curve of classifier 2 shows a better performance than the others. Classifier 0 shows the worst ROC curve that can be achieved: the results of a baseline or a random classifier.

AUC-ROC (Area Under the ROC Curve) is the probability that a classifier will rank a randomly selected positive instance higher than a randomly selected negative one. *AUC-ROC* is a useful measure to compare the performance of different classifier models. An area of 1 represents a perfect model; an area of 0.5 represents a random model. The *AUC-ROC* values of the 3 classifiers ROC curves in Figure 3.7 are 0.5, 67.3 and 84.5 respectively for classifiers 0, 1, and 2. This quantifies the fact that the classifier 2 outperforms the others.

3.4.3 PR Curve

Differently, Precision-Recall (PR) curves show the trade-off between precision and recall when the discrimination threshold changes. Figure 3.8 shows an example of PR curve. It can be plotted with the same steps of plotting the ROC curve [DG06]. We can compute the precision and the recall at various threshold settings instead of computing TPR and FPR. PR curve is an alternative to ROC curve and there is one-to-one mapping between points in ROC space and PR space. The translation between the two curves can be achieved by using the confusion table. If a curve dominates in ROC space then it dominates in PR space and vice-versa. The goal of ROC curve is to be as close as possible to the upper-left-hand corner, while PR curve goal is to be as close as possible to the upper-right-hand corner. PR curve is more appropriate than ROC curve if true negatives are not much valuable to the task (as true negatives are not a component of either Precision or Recall).

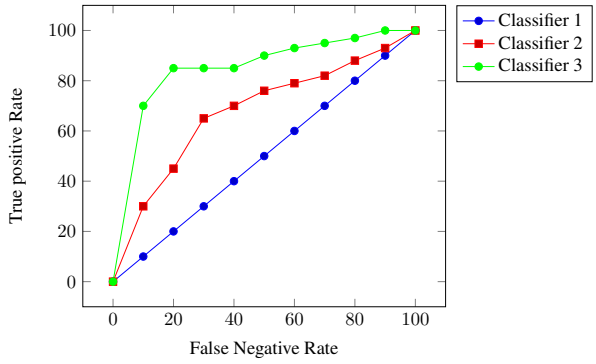


Figure 3.7: Example of ROC Curves

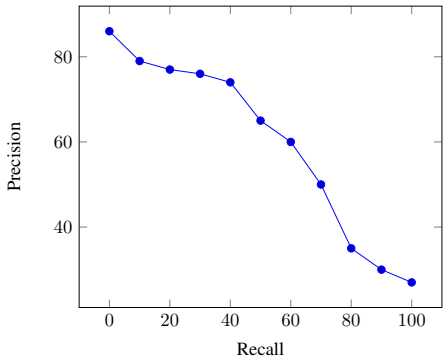


Figure 3.8: Example of PR Curve

3.5 Summary

This chapter reviewed different ways of representing text, including lexical, syntactic and semantic representations. The lexical representation shows the morphology of texts with tokens, which are the basic units for downstream processing. The syntactic representation relies on syntactic parsing that analyzes a text grammatical structure. Different syntactic representations are possible including sequences of POS or chunks, parse trees or dependency graphs. Also, we discussed the semantic representation of a sentence using CGs, Frame Semantics, SRL and compositional semantics. CGs represents the semantics of a sentence in unlabeled graph by representing the words in nodes and the semantics relations between them in other nodes connecting the word nodes. While Frames Semantics represents the semantics of a sentence in a template, called frame. This frame defines the main words of the sentence and their semantics (frame elements). SRL identifies the semantic relations among a predicate and its participants in a text. Compositional semantics represents the semantic of a sentence or a phrase by a vector. This vector represents the meaning of all words of the sentence or the phrase by encoding their contextual information in a large corpus.

Secondly, we presented some applications of text mining such as IE, ontology Construction, IR, Document Classification, Clustering, Text Summarization and Sentiment Analysis. Also, we presented the subtasks of an IE system such NER and RE. NER is used to annotate a text with interesting named entities. It is the base subtask for a lot of text mining applications. RE is a subtask of IE that extracts relationships between named entities, defined by a NER tool, from text. RE could extract relationship not only between two named entities (known as binary RE) but also between more than two named entities (known as complex RE). EE, similar to complex RE, is used to extract an event which consists of many entities and define the relationships between these entities.

In this thesis, we rely mainly on RE as a text mining approach for relationship extraction from medical texts. Furthermore, in section 3.3 we provide a detailed description of RE task and the state of art methods for RE in biomedicine. We categorize these approaches into 4 main categories: co-occurrence, patterns or rules, machine learning and multiple classifier systems.

Finally, we discussed the evaluation metrics for a text mining approach. The results of this approach are qualified based on how many corrected relationships are extracted and their coverage. So, we use metrics that are similar to what are used for evaluating binary classifiers, which depend on a confusion matrix. These measures are $Precision$, $Recall$, $F\text{-}Measure$, $ROC\ curves$, $PR\ curves$ and $AUC\ ROC$. $Precision$ is used for computing the probability of correctly extracting a relationship as true, while $Recall$ is used for computing the probability of extracting a true relationship as a true. $F\text{-}Measure$ is used to select the best settings that balance $Precision$, and $Recall$. Also, we used $ROC\ curves$, $PR\ curves$ and $AUC\ ROC$ for comparing the performances of different approaches or classifiers, which are not captured by $Precision$, $Recall$ and $F\text{-}Measure$.

4

Extracting Disease-Phenotype Relationships from Text

Contents

4.1 Introduction

Disease-Phenotype (D-P) relationships are of major importance for biomedical informatics because they provide a fine-grained description of disease and then could guide the medical diagnosis of disease in clinical care. However, biomedical databases that catalog D-P relationships such as Orphadata or OMIM are incomplete in comparison with the state of the art described by unstructured text in the scientific literature. In addition, extracting this information manually from the literature by experts requires a lot of time and effort, which motivates the need for developing automatic methods.

Orphadata and OMIM are two examples of databases that catalog D-P relationships. Orphadata is a database accessible from Orphanet, the portal for Rare Diseases (RDs) and orphan drugs. It includes descriptions of phenotypes (namely clinical signs) of RDs. OMIM (Online Mendelian Inheritance in Man) is a database for genetic diseases, which contains disease descriptions that include a list of phenotypes named "clinical synopsis".

Due to the fact that their content is manually curated by experts, Orphadata and OMIM are usually considered as high-quality resources. However, they do not offer a complete list of relationships between diseases and their

phenotypes, and consequently may be enriched by a systemic review of the biomedical literature. Table 4.1 shows that, among the 8,644 diseases listed by Orphadata, only 2,689 diseases (31.11%) are associated with clinical signs and phenotypes. Indeed, one can use cross references between Orphadata and OMIM[3] to associate Orphadata diseases to phenotypes described in OMIM. Nevertheless, even when considering these additional phenotypes, only 4,856 (56.18%) Orphadata diseases have phenotypes. The rest, 3,788 Orphadata diseases, is not related to any phenotype. In addition, knowing that a disease is associated with at least a phenotype does not mean that this list is complete. It may be an incomplete description of the disease. This motivates us to complete this lack by extracting D-P relationships from the literature.

	#Diseases	#Diseases associated with phenotypes	#Phenotype	#D-P Relations
Orphadata	8,644	2,689	1,273	52,503
OMIM	23,929	23,910	46,369	432,760
Orphadata ∪ OMIM	29,097	23,910	47,549	485,263

Table 4.1: Information about Orphadata and OMIM databases

In many domains such as biomedical research, text is a major source of information; unfortunately text corpora are often too large to be fully considered manually [LvI10]. Therefore, automatic methods for Information Extraction (IE) from text are necessary. We describe in this chapter a novel method for Relation Extraction (RE), which is a subtask of IE and consists in identifying and qualifying automatically valid relationships between entities previously recognized (see Section 3.3 in Chapter 3). In this chapter we study how text, represented in the form of graphs, can be processed with simple graph mining (*e.g.,* pattern-based, Machine Learning (ML)) methods, to perform RE.

In this context, we proposed an automatic method, called SPARE* for extracting D-P relationships. SPARE* combines indeed two methods: an original pattern-based method named SPARE (Syntactic PAttern for Relationship Extraction), and a classical ML algorithm based on SVM. SPARE is based on the identification of patterns in the shortest paths that relate entities in a sentence Dependency Graph (DG).

Objectives of this chapter are twofold, by presenting our methods for: (1) learning patterns for D-P relationships extraction; (2) combining a pattern-based method with a ML method to improve the extraction result.

This chapter is organized as follow: section 4.2 describes the manually annotated corpus we designed for D-P relationship extraction. Then, section 4.3 details our method for extracting D-P relationships. Section 4.4 presents its experimentation and results. Section 4.5 discusses the results, and finally, section 4.6 concludes the chapter.

4.2 A Manually Annotated Corpus for D-P Relationships

We built a corpus that is annotated by RDs and phenotypes to be used for learning and testing our approach for D-P RE. This corpus is made of 121,796 abstracts about 457 distinct RDs, obtained from PubMed. Abstracts were selected if they contain the name or a synonym of a RD, as defined by the Orphanet Rare Disease Ontology (ORDO) [ORD15]. The 457 RDs have been selected because they satisfy the following criteria: (1) they are associated with phenotypes (named "clinical signs") in Orphadata; (2) they can be mapped to an OMIM disease through UMLS CUI; (3) they are associated with phenotypes in OMIM. This enables having a corpus of a reasonable size and guarantees that the selected RDs are associated with phenotypes in both Orphadata and OMIM.

Abstracts are obtained by querying PubMed, using its web user interface. The query submitted to PubMed has the following form: "$(disease_{1,pref\_name}$ `or` $disease_{1,syn_1}$ `or...or` $disease_{1,syn_n})$ `or...` `or` $(disease_{k,pref\_name}$ `or` $disease_{k,syn_1}$ `or...or` $disease_{k,syn_m})$" where $disease_{i,pref\_name}$ and $disease_{i,syn_j}$ are respectively referring to the preferred name and the j^{th} synonym of disease i according to ORDO. The list of

[3] 4,162 Orphadata diseases have cross references to OMIM diseases.

457 RDs selected to build this corpus and the list of the PMIDs (*i.e.,* identifiers of articles in PubMed) of the 121,798 abstracts are available at `https://sourceforge.net/projects/spare2015/files/457-diseases` and `https://sourceforge.net/projects/spare2015/files/PMID-List` respectively.

The 121,796 abstracts were split into 907,088 sentences using LingPipe [Lin15]. Each sentence has been annotated by MetaMap with concepts associated with one of the following semantic types: Disease or Syndrome (T047), Sign or Symptom (T184). Then, sentences that are not annotated with at least one concept of semantic type T047 and another of semantic type T184 are filtered out, to obtain 2,341 sentences.

Finally, a corpus of all 2,341 sentences has been manually annotated by myself (Mohsen Hassan, noted MH afterwards) to identify true and false relationships: the annotation task mainly requires linguistics and NLP skills. A true relationship is counted when a pair D-P is found and a relationship between them is actually mentioned in the text; whereas a false relationship is listed when the pair is found but no relationship is mentioned. The total number of relationships annotated in the corpus is 5,630. 3,010 relationships were annotated as true, while 2,620 relationships as false. Our corpus annotations are kept in XML file and is publicly available at `https://sourceforge.net/projects/spare2015/files/D-PCorpus.xml`. Listing 4.1 presents an example of our annotation standard. The annotations of a sentence are provided between the tags <Sentence> and <Sentence/>. The sentence itself is provided between the tags <SentenceString> and <SentenceString/>. A list of disease mentions are between the tags <DiseaseList> and <DiseaseList/>, while a list of phenotype mentions are between the tags <PhenotypeList> and <PhenotypeList/>. A list of true D-P relationships is provided between the tags <RelationshipList> and <RelationshipList/>. The other D-P relationships that are in the sentence but not mentioned between these tags are false D-P relationships. The sentence in Listing 4.1 shows examples of both a true and a false relationship. “Neuroacanthocytosis”-“involuntary choreiform movements’ is a true relationship, while “rare hereditary disorder”-“involuntary choreiform movements” is false.

```
<Sentence PMID="18945802" id="148755" sentenceOrderInAbstract="1">
    <SentenceString>
      Neuroacanthocytosis is a rare hereditary disorder characterized by
      involuntary choreiform movements.
   </SentenceString>
    <AnnotatedSentenceString>
      DISEASE1 is a DISEASE2 characterized by PHENOTYPE1.
   </AnnotatedSentenceString>

   <DiseaseList>
        <Disease id="69373" CUI="C0393576" positionInfo="[(0, 18)]" semanticType="[dsyn]">
         Neuroacanthocytosis
      </Disease>
        <Disease id="69374" CUI="C0678236" positionInfo="[(24, 48)]" semanticType="[dsyn]">
         rare hereditary disorder
      </Disease>
    </DiseaseList>

   <PhenotypeList>
        <Phenotype id="372" CUI="C0427086" positionInfo="[(66, 98)]" semanticType="[sosy]">
         involuntary choreiform movements
      </Phenotype>
    </PhenotypeList>

    <RelationshipList>
      <Relation Disease="69373" Phenotype="372"/>
   </RelationshipList>
</Sentence>
```

Listing 4.1: Example of an annotated sentence from the corpus.

4.3 A Hybrid Approach for Extracting D–P Relationships

Similarly to [HZL06, SHK+14], we propose a hybrid approach, named SPARE⋆, that combines a pattern-based method, named SPARE, and a classical ML algorithm (SVM in our case). Results of both methods are combined to benefit from the relatively high precision of SPARE and from the good recall of SVM. In the whole approach, we consider linguistic and syntactic features of text to guide the RE.

Figure 4.1 shows the main steps of SPARE⋆, positioning particularly SPARE and the SVM. First, texts are split into sentences with LingPipe. Each sentence is tokenized, then the tokens are lemmatized and part-of-speech tags are computed with the Stanford CoreNLP suite. Next, disease and phenotype entities are recognized and annotated in text using a NER tool (MetaMap in our case). Next, a syntactic analysis, which includes the construction of DGs is performed. DGs serve as the input to the SPARE method, while a feature matrix of syntactic features (including DGs) serves as the input to the SVM algorithm. Finally, we designed several strategies to combine the results of both SPARE and SVM. Next subsections describe these two methods and their combination.

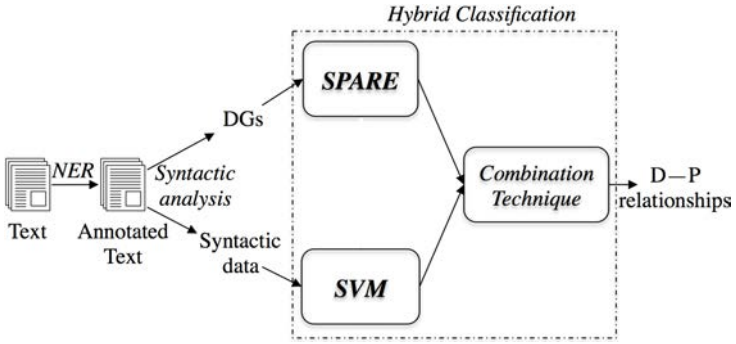


Figure 4.1: Overview of our hybrid method named SPARE⋆. First, SPARE and SVM classify D-P relationships. Then, we combine the classification results of SPARE and SVM by using a combination strategy.

4.3.1 Using Syntactic Patterns with SPARE

This section presents our first method, named SPARE (standing for Syntactic PAtterns for Relationship Extraction), for RE. SPARE is based on the learning of syntactic patterns used next for RE. SPARE has been inspired from previous works such as using the shortest path between entities of a DG as Bunescu et Mooney [BM05], Chowdhury *et al.* [CL12] and Liu *et al.* [LVC+13]. Their results show that using only the shortest path enables capturing the most important features required to describe the relationship between two entities. Considering a complete graph or all-possible paths is a NP-hard problem that costs in processing. While the shortest path is simple and easy to compute. Similarly to [LVC+13], we extract shortest paths represented by the whole subgraph (*i.e.,* all nodes and edges in the shortest path). Unlike them, we keep edge directions to make the patterns more precise.

SPARE relies on three main steps presented in Figure 4.2. *(1)* First, *syntactic patterns* are learned from a set of DGs, which all include a disease and a phenotype. *(2)* Second, patterns are selected in regard to their *support* and *positive-predictive value (ppv)* (*i.e.,* their capacity to identify true relationships). *(3)* Third, selected patterns are applied on considered texts to extract D-P relationships. Following subsections details these three steps.

Definition 18 *(Pattern support)*
The *support* of a pattern p is the number of D-P pairs that matches the pattern in our learning corpus. It can be defined as:

$$support_p = |TR_p \cup FR_p| \tag{4.1}$$

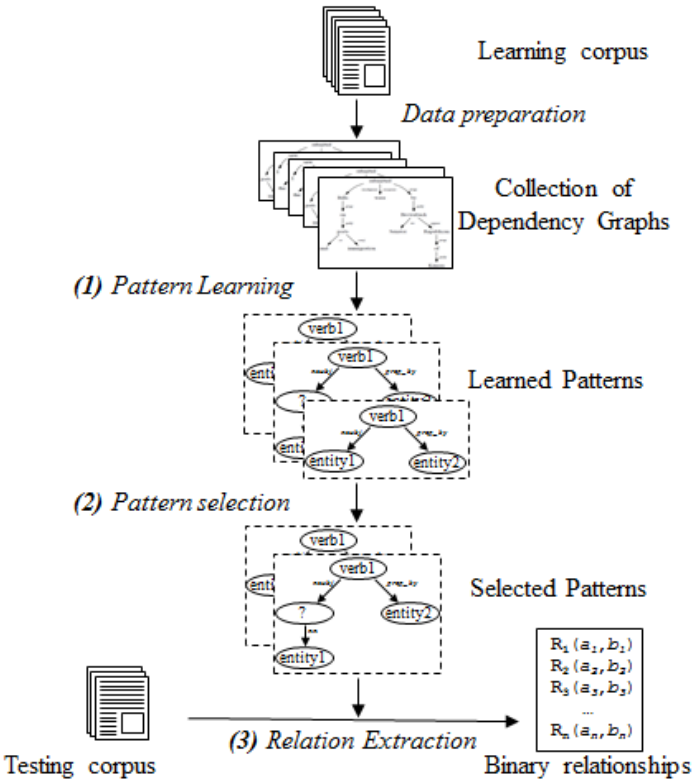


Figure 4.2: The three main steps of the SPARE method.

where TR_p and FR_p are respectively the number of true and false relationships that match with p in our learning corpus. A pattern is considered as $frequent$ if its *support* is greater than or equal to $min\_support$.

Definition 19 *(Pattern ppv)*
The *ppv* of a pattern measures the ability of a pattern to extract true relationships. It may be seen as the probability, for a relationship that matches the pattern, to be true rather than false. For a pattern p, its ppv is defined by:

$$ppv_p = P(TR_p) = \frac{|TR_p|}{|TR_p \cup FR_p|} \tag{4.2}$$

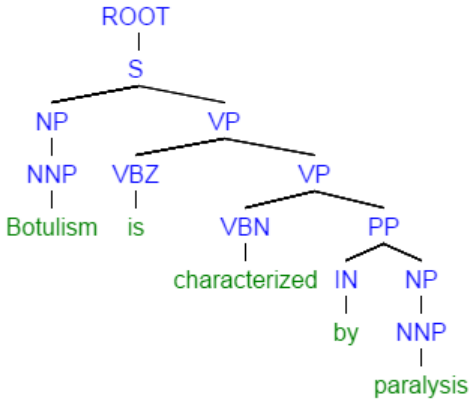
Learning Syntactic Patterns from Dependency Graphs

We selected DGs of sentences, rather than their syntax trees, to learn syntactic patterns because their structure stays consistent over various phrasing of similar events. For instance, sentences of Examples 4.3.1 and 4.3.2 propose two different phrasings of the same relationship. Their respective syntax trees (Figure 4.3) propose two different paths between the disease "Botulism" and the phenotype "paralysis", while their dependency graphs (Figure 4.4) propose one same path between them[4]. Also, the dependency path between two entities (*i.e.,* nodes) in a DG is usually shorter than their syntactic path in the corresponding syntactic tree. Consequently, using DGs instead of syntactic trees will produce a smaller set of short dependency paths, which consequently is processed faster.

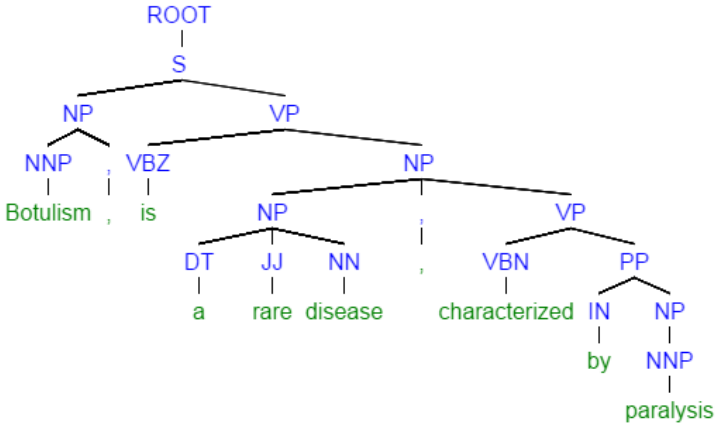
[4]In this paper, Syntax trees and DGs are computed with the Stanford Parser, and drawn respectively with Syntree, available at `http://mshang.ca/syntree/` and Brat, available at `http://nlp.stanford.edu:8080/corenlp/`

Ex. 4.3.1 "Botulism is characterized by paralysis"

Ex. 4.3.2 "Botulism, a rare disease, is characterized by paralysis"

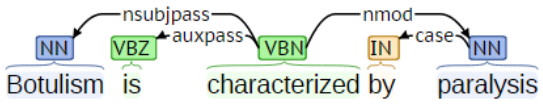


(a)

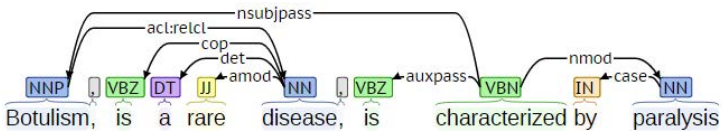


(b)

Figure 4.3: (a) and (b) present the syntax trees generated respectively from sentences of Examples 4.3.1 and 4.3.2



(a)



(b)

Figure 4.4: (a) and (b) present the Dependency Graphs (DGs) generated from sentences of Examples 4.3.1 and 4.3.2

We kept in the studied corpus only sentences that contain at least one disease and one phenotype. Then, DGs of these sentences are generated. In each DG, the annotated entities (*i.e.,* diseases and phenotypes) are replaced by generic words, *i.e.,* DISEASEx and PHENOTYPEy where x and y are indexes to distinguish between diseases or phenotypes when more than one are mentioned in a sentence; and other words are replaced by their lemmas. Figures 4.5(a) and 4.5(c) show the DGs generated from sentences of Examples 4.3.3 and 4.3.4.

Ex. 4.3.3 "*<disease>* Epidermolytic hyperkeratosis *</disease>* is a disorder of cornification characterized by *<phenotype>*hyperkeratosis*</phenotype>*"

Ex. 4.3.4 "*<disease>* Myotonic muscular dystrophy *</disease>* is a disease of autosomal dominant inheritance characterized by *<phenotype>*myotonia*</phenotype>*"

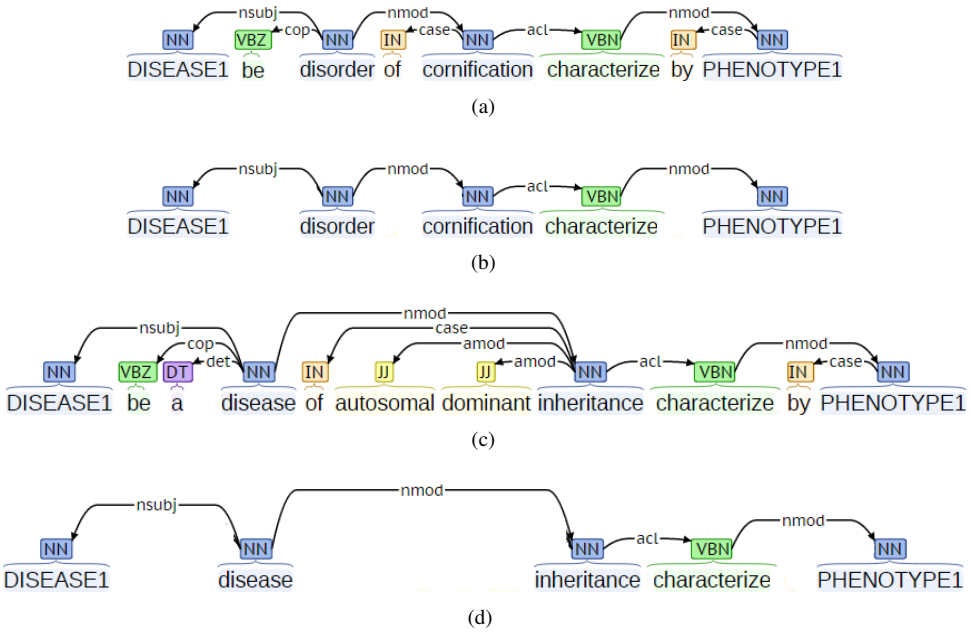


Figure 4.5: DGs (a,c) and shortest paths (b,d) between a disease and a phenotype respectively extracted from Ex. 4.3.3 and 4.3.4

Each DG is explored to find the *shortest path* between its disease and phenotype. Indeed, one sentence may mention several diseases or phenotypes thus may generate several shortest paths. Shortest paths are simply DG subgraphs (*i.e.,* including nodes, edges and directions) that connect a disease to a phenotype. Figures 4.5(b) and 4.5(d) show the shortest paths found in DGs represented in figures 4.5(a) and 4.5(c). The whole shortest path is kept, including the nodes, the edges and their orientations.

Then, shortest paths that contain the following are excluded: *(1)* conj_and or conj_or dependency relation between any two nodes; *(2)* the path from the root of the graph to the disease and the path from the root of the graph to the phenotype are identical, this means that disease and phenotype have the same semantic role in the sentence and this might be an error from NER. Figure 4.6 shows an example of such excluded paths. They can be discovered from a sentence like "This disease is characterized by DISEASE and SYMPTOM"[5].

Next, *syntactic patterns* are built on the basis of shortest paths, using a generalization process similar to the one described by Adolphs *et al.* [AXLU11]. Indeed, a pattern mining algorithm generates a very large set of patterns and the purpose of this generalization process is to keep a reduced and compact set of patterns. In this process,

[5]The uppercase words are the generic words for NEs

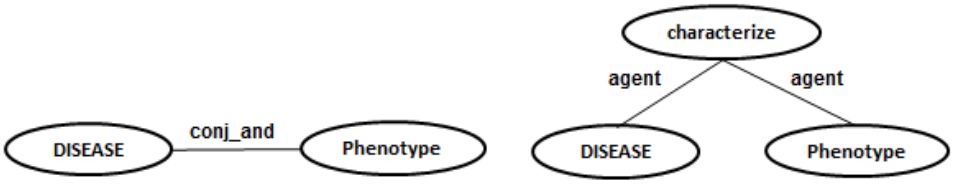


Figure 4.6: Two examples of excluded shorted paths. These paths can be obtained from a sentence of the form "This disease is characterized by DISEASE and SYMPTOM"

individual shortest paths are first generalized and merged as a single pattern when equivalent. Two generalized paths (or more) are merged into one pattern if they share the same edges and directions. Figure 4.7 illustrates this generalization process considering the shortest paths of Figures 4.5(b) and 4.5(d) obtained from Examples 4.3.3 and 4.3.4. If the values of the nodes in the pattern are different, then they are replaced by "*" (*i.e.,* a "joker" matching any token). A list of values observed for each node is kept but for documentation purpose only. Resulting patterns are named syntactic patterns and are characterized by their *support*, *i.e.,* how many relationships in our learning corpus match this pattern.

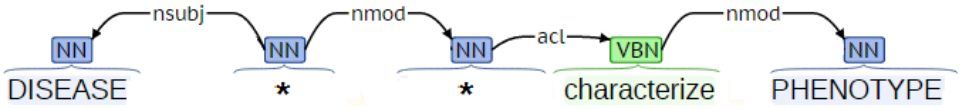


Figure 4.7: Example of syntactic pattern obtained by generalizing and merging the two shortest paths presented in Figures 4.5(b) and 4.5(d)

Pattern Selection

Once syntactic patterns are generated, they are classified in two classes: *positives* and *negatives*. This classification relies on two metrics: the $support$ and ppv of patterns. A pattern is considered positive if its $support$ and ppv are greater than or equal to a minimum support denoted $min\_support$ and a minimum ppv denoted min_ppv respectively. A pattern that does not satisfy these conditions is considered as negative.

These two metrics are computed on the basis of a manual annotation of the sentences of our learning corpus (we described this corpus in more details in Section 4.2). Each D–P pair found in a sentence is annotated manually as true if the sentence mentions a relationship between the two entities, or as false otherwise.

For example, the pattern in Figure 4.7 has a support of 23 ($support_p = 23$). This means that the number of relationships in the learning corpus that matches with this pattern is 23. As all D-P relationships matching this pattern are true, then the positive-predictive value of this pattern is 1 ($ppv_p = 23/23$).

Relationship Extraction

Positive patterns, *i.e.,* patterns with sufficient $support$ and ppv, are used for extracting relationships from our testing corpus. Each sentence of the corpus that is annotated with one (or more) disease(s) and one (or more) phenotype(s) is transformed in its DG, and then compared for *pattern matching* to our set of positive patterns. When a match is found, a D–P relationship is extracted between the matching disease and phenotype.

Let's illustrate this by a simple example. Given the sentence of Example 4.3.5, MetaMap annotates "Familial Mediterranean fever" as a disease, and "recurrent fever" as a phenotype. Then, the Stanford Parser generates the

DG of this annotated sentence as shown in Figure 4.8. When comparing this sentence with the positive patterns, the DG of this sentence matches with the pattern shown in Figure 4.7. This enables extracting the following D-P relationship ("Familial Mediterranean fever", "recurrent fever").

Ex. 4.3.5 "*<disease>* Familial Mediterranean fever *</disease>* is a disorder of autosomal dominant characterized by *<phenotype>* recurrent fever*</phenotype>*"

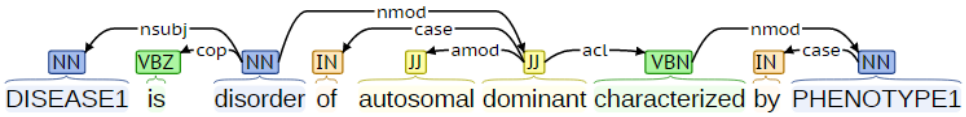


Figure 4.8: DG of the sentence in Example 4.3.5.

4.3.2 Using SVM for classifying D–P Relationships

The second method for identifying D–P relationships is SVM, a supervised ML approach. SVM is used to classify some extracted D–P relationships in 2 classes: true or false. Figure 4.9 shows the main steps for building the SVM classifier. These steps are: defining our learning and testing datasets; extracting the features that qualify instances of the learning dataset; selecting best features to keep in feature vectors; and finally learning the model to be used, in turn, to classify novel instances.

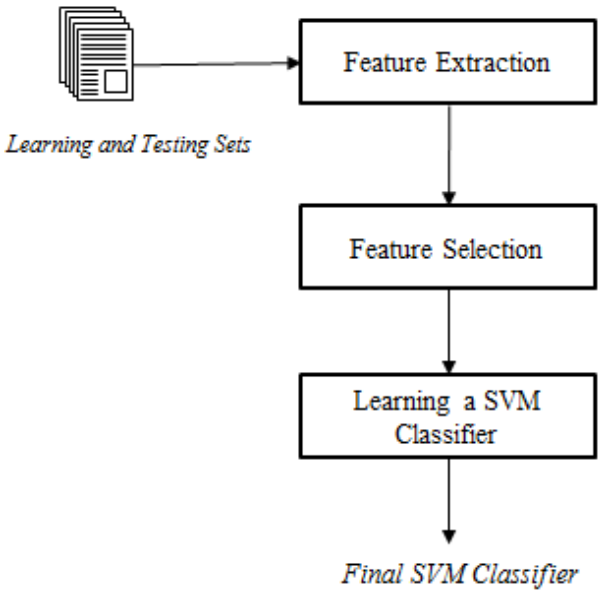


Figure 4.9: Steps of building a SVM classifier.

Data Sets

Our learning and testing sets are adapted from our effort of manually annotating D–P pairs in sentences of our corpus. D–P pairs annotated as true or false constitute respectively positive or negative instances of our data set. The class, *i.e.*, positive or negative, is considered as the target class for our classifier.

Feature Name	Information Gain	Description
DGCompletePathLemma	0.82	DG path between D and P, where vertices are lemma of words
DGCompletePathPOS	0.79	DG path between D and P, where vertices are POS of words
e_walkCompletePath	0.73	The sequence of dependency types (*i.e.*, edge labels) found in the DG between the D and the P.
DGPathToDiseaseLemma	0.63	DG path between the root and D, where vertices are lemma of words
DLPOS1	0.13	POS of the first word on the left of D
Exact1VB	0.05	DG path between D and P, containing exactly one verb

Table 4.2: Information Gain values and description of the 6 selected features for learning our SVM classifier. DG means Dependency Graph, POS means Part Of Speech, D and P stand respectively for the Disease and the Phenotype entities.

Feature Extraction

For each D–P pair annotated, 39 syntactic features, including contextual information and DG, are extracted from the syntactic analysis of the sentence where the pair appears. The exhaustive list and description of considered features are presented in Table B.1 in Appendix B. Particularly, it includes lemma, POS tag of neighbor words and DGs of various portions of the sentence. We defined this extensive list of features on the basis of previous works by Zhou and Chowdhury *et al.* [ZSZZ05, CLM11, CL12].

Feature Selection

To simplify our classification model and make it faster to train, we use a feature selection method for reducing the dimensionality of feature vectors. We used $CfsSubsetEval$, which is a method for feature selection introduced by Hall [Hal99] and available in the Weka toolbox [SF16]. It evaluates the worth of using a subset of features for classification. It considers the individual predictive ability of each feature along with the degree of redundancy between them. For our 39 features, it identifies the subset of 6 features listed in Table 4.2 as the best subset for our classification task. The worth of a feature f is evaluated by calculating its information gain with respect to the class c of instances, as shown in Eq. (4.3).

$$InfoGain(c, f) = H(c) - H(c|f) \tag{4.3}$$

where H is the information entropy (see [Gra90] for more details about information entropy).

Learning a SVM Classifier

Finally, the 6 selected features by $CfsSubsetEval$ constitute the feature vector of each instance, *i.e.,* of each D–P pair found in sentences of our learning corpus. These 6 features are then used to learn the selected ML classifier (SVM in our case) as the final classifier.

4.3.3 Combining Syntactic Patterns and SVM: the SPARE* Approach

This subsection details the combination of SPARE and SVM methods and the elements that explain its design. The combination considers the classification results of the two methods to propose a final classification. We expect this combination could improve the final results thanks to the good precision of SPARE and the good recall of SVM.

SPARE extracts D–P relationships from text. For these extracted relationships we define 3 classes: 1) positive D–P relationships, which matched with a positive pattern; 2) negative D–P relationships, which matched with a negative pattern; 3) unknown D–P relationships, when a D–P pair does not match any pattern. In subsection 4.3.1, we considered indifferently 2) and 3) as negative. For the combination, we distinguish unknown from negative relationships to propose to classify differently these two groups.

To adopt the best combination strategy, we experimented with several strategies that we compared for their ability in extracting positive D–P relationships. Table 4.3 details the 6 combination strategies we considered. Distinction between various strategies is: first, the consideration of unknown relationships as negative similarly to what is done by SPARE, or as an independent kind of relationships; second, the logical operator chose for the combination, *i.e.*, AND, OR and the priority of one classifier.

For example, SPARE_AND_SVM and SPARE_AND_SVM_Unknown combine SPARE and SVM classifications with the logical "AND", meaning that a relationship is positive if the SPARE and the SVM are classifying at as positive. SPARE_AND_SVM considers unknown relationships (matching with neither positive or negative syntactic pattern) as negative, whereas SPARE_AND_SVM_Unknown considers them separately and then let the SVM considering on its own these relationships for classification. SPARE_pr_Unknown and SVM_pr_Unknown use SVM classification for classifying unknown relationships, but propose different priority between SPARE and SVM classifications in case they disagree. SPARE_pr_Unknown strategy uses SPARE classification for the final result if it is not unknown, while SVM_pr_Unknown strategy uses SVM classification for the final result.

SPARE	SVM	SPARE_AND_SVM	SPARE_OR_SVM	SPARE_AND_SVM_Unknown	SPARE_OR_SVM_Unknown	SPARE_pr_Unknown	SVM_pr_Unknown
Unknown	+	-	+	+	+	+	+
	-	-	-	-	-	-	-
+	+	+	+	+	+	+	+
	-	-	+	-	+	+	-
-	+	-	+	-	+	-	+
	-	-	-	-	-	-	-

Table 4.3: Presentation of the 6 combination strategies we considered for classifying D-P relationships either as positive (+) or negative (-). Strategies combine the output of SPARE and SVM. They consider either unknown relationships as negative or as unknown. In the latter case, strategy names are suffixed with 'Unknown'. 'AND' and 'OR' refers to the logical operator considered for the combination. In the two last strategies, priority is given to the results of one classifier in regards to the other. This is denoted with the 'pr' suffix. We can notice that $SPARE\_OR\_SVM$ and $SPARE\_OR\_SVM\_Unknown$ produce the same classification. This is explained by the fact that in this case of the OR, considering unknown relationships as negative does not impact the final result.

4.4 Experiments and Results

4.4.1 SPARE

Fixing SPARE parameters

The 2,341 sentences of our manually annotated corpus (described in Section 4.2) with at least one disease and one phenotype are used to define the min_ppv and $min\_support$ thresholds. These sentences are split into a learning corpus made of 90% of sentences (randomly selected) and a testing corpus made of 10% of sentences. Table 4.4 shows the characteristics of the learning and testing corpora in term of a number of sentences and of true and false relationships. The number of generated patterns from the learning corpus is 1,049. We fixed $min\_support = 2$ and $min\_ppv = 0.5$, because this reduces our selected patterns to 235 and enables to achieve the best $F\text{-}measure = 0.57$ ($precision = 0.88$, $recall = 0.42$) on the testing corpus. Figure 4.10 shows the changes of $F\text{-}measure$ by using different min_ppv thresholds. It shows that the best $F\text{-}measure$ is 0.57 at $min\_ppv = 0.5$.

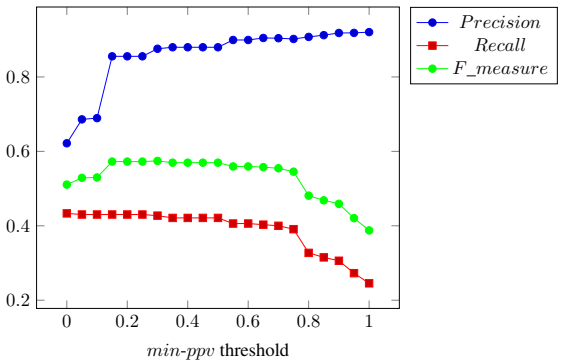


Figure 4.10: The effect of $min\text{-}ppv$ threshold on $precision$, $recall$ and $F\text{-}Measure$ values.

Corpus	#Sentences	#True Relationships	#False Relationships
learning	2,107	2,680	2,294
testing	234	330	326
total	2,341	3,010	2,620

Table 4.4: Size and content of the learning and testing corpora used for learning SPARE parameters.

Evaluating SPARE for RE

The SPARE method is evaluated by using a 5-folds cross-validation. This approach is used to learn, then test SPARE patterns using the corpus of 2,341 sentences with at least one disease and one phenotype. As shown in Table 4.4, this corpus contains 5,630 D–P relationships where only 3,010 are true and 2,620 are false. The 5-folds cross-validation results a precision of 0.87, a recall of 0.51 and a F-measure of 0.65.

4.4.2 SVM

Choosing SVM

We experimented different ML methods using a 5-fold cross-validation to choose the one that gives the highest recall. The set of considered methods includes Naïve Bayes, SVM, instance-based learning, two rule-based

ML Method	Precision	Recall	F-measure	AUC-ROC
Naïve Bayes	0.83	0.80	0.82	0.89
SVM	0.55	**0.99**	0.71	0.57
Lazy IBK	0.82	0.85	0.83	0.85
Rules One R	0.71	0.90	0.79	0.76
Rules Zero R	0.52	1	0.68	0.5
Trees J48	0.71	0.93	0.81	0.5
RandomForest	0.80	0.79	0.79	0.89
RandomTree	0.78	0.69	0.73	0.83

Table 4.5: Results of 5-fold cross-validation for the classification of D–P relationships using various ML methods. Classified D–P pairs were manually annotated.

Method	Attributes	Precision	Recall	F-Measure	AUC-ROC
SVM	All	0.55	0.99	0.71	0.57
	One Attribute	0.70	0.76	0.73	0.70
	6 Attributes	0.68	0.92	**0.78**	**0.73**

Table 4.6: The results of applying SVM by using different selections of features.

methods and three decision tree methods. We used these methods from the Weka toolbox for extracting D-P relationships on the same corpus of 2,341 sentences. Table 4.5 shows the results of the 5 cross-validation for these ML methods. As shown, SVM that is implemented by LibSVM in Weka achieves the best *recall* (0.99). We choose SVM to combine with SPARE as we are looking for a ML classifier that achieves the highest recall.

Features selection

For the 39 features we considered initially with SVM, $CfsSubsetEval$ identifies a subset of 6 features listed in Table 4.2 as the best subset for our classification task. Also, the computation of the InfoGain of the 39 features shows that the feature named "DGCompletePathLemma" that corresponds to the DG path between the disease and the phenotype (where vertices are lemma of words) has the highest value ($InfoGain$=0.82). It is interesting to note that elements constituting this feature are closed to those constituting the syntactic patterns used in our SPARE method.

Table 4.6 shows that using vectors of the 6 selected features achieves a better F-measure (0.78) than using vectors of all 39 features or vector of only one feature ("DGCompletePathLemma", which has the highest information gain value). Also, it achieves the best AUC-ROC (Area Under ROC Curve) value, 0.73. Figure 4.11 shows their ROC curves and illustrates that SVM with the 6 selected features outperforms the others.

4.4.3 Combinations

The choice of designing a hybrid approach was to enrich SPARE with a method associated with a good recall, such as SVM. Hence, we evaluated and compared the distinct combination strategies of both methods for D–P relationship extraction. Table 4.7 presents the results of an evaluation of the different combinations of SPARE and SVM, when considering the set of the 6 features selected by $CfsSubsetEval$ (listed in Table 4.2). Figure 4.12 shows the associated ROC curves. It shows that SPARE_AND_SVM_Unknown and SPARE_pr_Unknown provide the best results in term of F-measure, which are 0.81 and 0.80 respectively. They also give the best AUC-ROC values, which are 0.79 and 0.78 respectively. These explain that SPARE_AND_SVM_Unknown and SPARE_pr_Unknown outperform the other combinations. Between these two strategies, we arbitrary chose

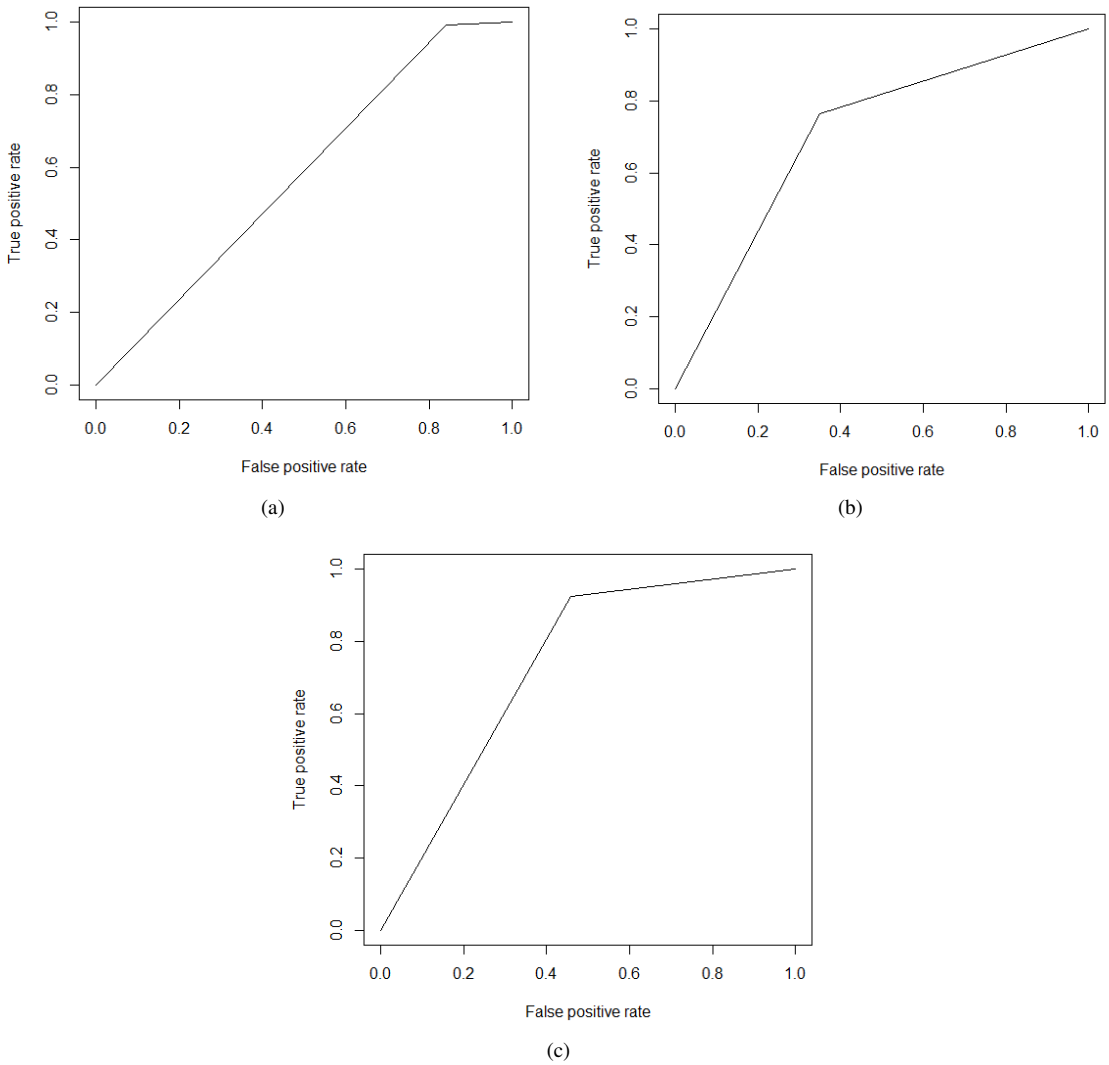


Figure 4.11: The ROC curves of SVM considering different feature vectors of all 39 features, one feature and the selected 6 features respectively.

SPARE_AND_SVM_Unknown for SPARE*. This hybrid approach shows an improvement in $F\text{-}measure$ with 16% and 3% over SPARE and SVM respectively.

Method	Precision	Recall	F-Measure	AUC-ROC
SPARE	0.87	0.51	0.65	0.72
SVM	0.68	0.92	0.78	0.73
SPARE_AND_SVM	0.89	0.51	0.65	0.72
SPARE_OR_SVM	0.68	0.93	0.79	0.73
SPARE_AND_SVM_Unknown	0.76	0.86	**0.81**	**0.79**
SPARE_OR_SVM_Unknown	0.68	0.93	0.78	0.73
SPARE_pr_Unknown	0.75	0.86	**0.80**	**0.78**
SVM_pr_Unknown	0.68	0.92	0.79	0.73

Table 4.7: Evaluation of various combination strategies for RE. Table 4.3 details how SPARE and SVM are combined. The evaluation is achieved while selecting the 6 features suggested by $CfsSubsetEval$.

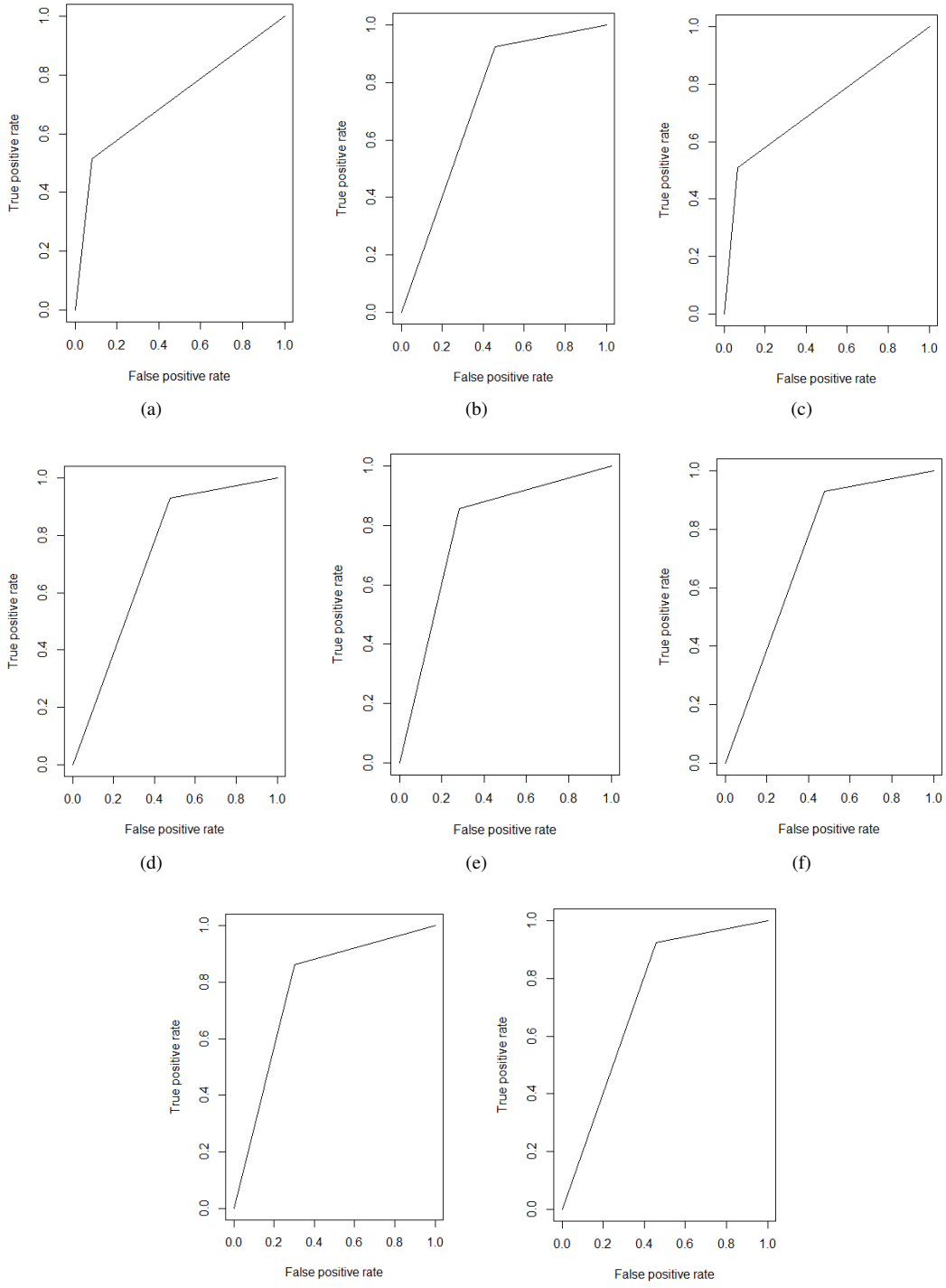


Figure 4.12: ROC curves of SPARE, SVM, SPARE_AND_SVM, SPARE_OR_SVM, SPARE_AND_SVM_Unknown, SPARE_OR_SVM_Unknown, SPARE_pr_Unknown and SVM_pr_Unknown respectively.

4.5 Discussion

SPARE* was learned and tested on a manually annotated corpus. This corpus was annotated only by MH to identify true and false relationships. MH has linguistics and NLP skills, but has no experience in the medical domain. Involving many experts, especially from medical and linguistics domain, in the annotation task will improve its quality and enables extending the size of the corpus.

With SPARE, we obtained a relatively good precision but a low recall. We consider that a larger corpus for learning patterns could enable us to increase the recall. Our learning corpus is annotated manually with true and false relationships and increasing its size would require a consequential effort.

The generalization process of building SPARE patterns affects the precision and the recall of the patterns. Replacing the node value in the shortest path by using "*" (*i.e.,* any token) makes the pattern more generic, and has the consequence of increasing the recall of the patterns. On the other side, we assume that edges (*i.e.,* dependency types of DGs) and their directions in the pattern guarantee its precision.

In SPARE, the choice of min_ppv has important consequences on the results of the relationship extraction. Figure 4.10 shows how the quality of the extraction changes when the min_ppv value is changed. We observed relatively few evolution of the *F-measure*. In Figure 4.10, min_ppv between 0.35 and 0.5 achieved the best *F-measure* of 0.57. They give the same result because the number of extracted patterns with min_ppv between 0.35 and 0.5 stays the same (235 patterns). Consequently, we chose arbitrarily, in this interval, $min\_ppv = 0.5$.

Usually, a D-P relationship is associated with a percentage in databases that says how often has been observed the presence of a phenotype with a disease. For instance, Orphadata relates each D-P relationship with a frequency, which may be assigned to the value "obligate" (the phenotype is always present and the diagnosis could not be achieved in its absence), "very frequent" (80 to 99%), "frequent" (30 to 79%), "occasional" (5 to 29%) or "very rare" (1 to 4%). Our approach extracts D-P relationships, but does not extract their frequencies. Extensions to our work may gain at considering this frequency, for instance by not considering only shortest paths, but also other additional nodes related to the shortest path. These additional nodes could be matched with a list of word clues that express the frequency such as {frequent, usually, rare, not, ...}. For example, Figure 4.13 shows the DG of a sentence "DISEASE is usually characterized by PHENOTYPE", where the disease and the phenotype are replaced by their generic word. A positive pattern could extract the following relationship <DISEASE, PHENOTYPE>. By exploring the shortest path in the original DG, we find that the node containing the adverb "usually" is linked with the node containing the verb "characterized". By considering this additional information, we could qualify in this particular example the relationships with its frequency.

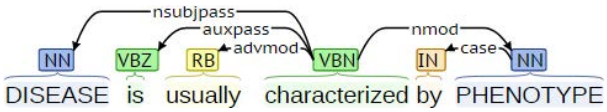


Figure 4.13: Example of a DG of a sentence that contains a frequency modifier such as "usually".

A SPARE pattern can discover D-P relationships from sentences with different and complex phrasing. Figure 4.14 shows an example of such patterns. Table 4.8 presents examples of sentences matched by this pattern and the extracted relationships from them. The sentences show these variations. For example, the phenotype in the first sentence comes close to the main verb "characterize", while in the second and third sentences it comes late. The fourth sentence shows that the main verb is different, which is "caused". The fifth sentence shows that 2 diseases joined together with "and" are successfully identified to be associated with the same phenotype "weakness" using the same pattern. The sixth example shows a complex sentence, where the disease is far from the phenotype, that matched with the same phenotype. These examples explain the capability of DG over other representation such as syntactic trees (see also subsection 4.3.1). In addition, its capability to identify the complete description. As

ID	PMID	Sentence	D-P Relationship
1	18945802	Neuroacanthocytosis is a rare hereditary disorder characterized by involuntary choreiform movements.	<Neuroacanthocytosis, involuntary choreiform movements>
2	22468686	Canavan disease is a severe autosomal recessive leukodystrophy characterized by macrocephaly, ataxia, severe motor and mental retardation, dysmyelination, and progressive spongial atrophy of the brain.	<Canavan disease, ataxia>
3	21931045	X-linked dominant chondrodysplasia punctata, also known as Conradi-Hunermann-Happle syndrome, is a rare skeletal dysplasia characterized by,short stature, craniofacial defects, cataracts, ichthyosis, coarse hair, and alopecia.	<X-linked dominant chondrodysplasia punctata, coarse hair>
4	24008937	Late-onset Pompe disease is a progressive metabolic myopathy caused by decreased activity of the enzyme acid alpha-glucosidase (GAA), which gives rise to reduced degradation and, later accumulation of glycogen in the lysosomes and cell cytoplasm.	<Pompe disease, decreased activity>
5	20443038	Both the myotonic dystrophy type 1 (DM1) and the X-linked dominant Charcot-Marie-Tooth disease (CMTX1) are well-established inherited neuromuscular,disorders characterized by progressive weakness and atrophy of the distal limb muscles.	<myotonic dystrophy type 1 (DM1), weakness>, <X-linked dominant Charcot-Marie-Tooth disease (CMTX1), weakness>
6	9054082	Krabbe's disease, globoid cell leukodystrophy, is a rare autosomal recessive demyelinating neurodegenerative disease caused by reduced activity of the lysosomal enzyme galactosylceramide beta-galactosidase which is involved in myelin metabolism.	<Krabbe's disease, reduced activity>

Table 4.8: Example of sentences matched with the pattern of Figure 4.14. This table presents the sentence and the D-P relationship extracted by the pattern.

shown in examples 5 and 6, MetaMap annotates only part of the phenotypes, which are "weakness" and "reduced activity". DGs helps to identify the complete description of phenotypes, which are "progressive weakness" and "reduced activity of the lysosomal enzyme galactosylceramide beta-galactosidase", by considering the modifiers linked with the phenotype in the DG.

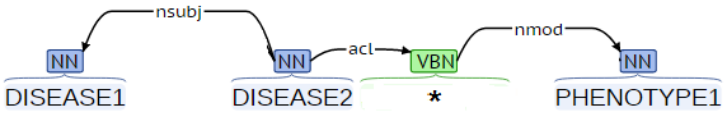


Figure 4.14: Example of a SPARE pattern.

Table 4.5 shows the results of the cross-validation using the Weka toolbox. Naïve Bayes achieves the best *precision* (0.83), whereas SVM achieves the best *recall* (0.99) and Lazy IBK achieves the best $F\text{-}measure$ (0.83). For the design of our hybrid approach, built-up on SPARE, we were looking for a method with a high recall that may balance the low recall of SPARE. For this reason, we chose SVM (and the LibSVM) that achieves the highest recall. Also, we chose a vector of 6 selected features for SVM classifier as proposed by the feature selection method "CfsSubsetEval". Table 4.6 shows that this selection leads to a better *F-measure* than using the vector of all the features or the vector of only one feature ("DGCompletePathLemma", which is associated with the highest information gain value). As shown in Table 4.7, its combination with SPARE increases the $F\text{-}measure$ 16% and 3% over SPARE and SVM respectively.

The computation of the InfoGain of the 39 features shows that the feature named "DGCompletePathLemma" that corresponds to the DG path between the disease and the phenotype (where vertices are lemma of words) has the highest value ($InfoGain$=0.82). It is interesting to note that elements constituting this feature are closed to those constituting the syntactic patterns used in our SPARE method.

4.6 Conclusion

In this chapter, we introduced a novel hybrid method named SPARE* to extract D-P relationships. It combines a pattern-based method, named SPARE (Syntactic PAtterns for Relationship Extraction), and a ML method (SVM).

SPARE is a pattern-based method that is composed of three successive steps used for learning high-quality patterns and for extracting D-P relationships. First, syntactic patterns are learned from shortest paths observed between the entities of interest (diseases and phenotypes) within DGs. Using only the shortest path is interesting because it is simple and it captures the most important features required to describe the relationship between two entities. Second, the patterns are selected based on two metrics: their *support* and their *positive-predictive value*.

Finally, the selected patterns could then applied on new texts for extracting D-P relationships.

For choosing a ML classifier, we experimented different ML approaches. Finally, we chose SVM because it achieves the best recall value (0.99), and was consequently adapted to combine with SPARE.

Various combination techniques are proposed for the composition of SPARE and SVM. The experiment shows that the hybrid method benefits from the relatively good $precision$ of the pattern-based method and of the good $recall$ of the ML method. Our experiment shows that this combination increases $F\text{-}measure$ and $AUC\text{-}ROC$. The best combinations are SPARE_AND_SVM_Unknown and SPARE_pr_Unknown that provide $F\text{-}measures$ of 0.81 and 0.80 respectively; and also the best AUC-ROC values, which are 0.79 and 0.78 respectively. These conclude that SPARE_AND_SVM_Unknown and SPARE_pr_Unknown outperform the other combinations. They also show a better $F\text{-}measure$ than using only SPARE or using only SVM.

5

Recognizing Complex Phenotypes Using Syntactic Patterns

Contents

5.1 Introduction

This chapter presents an approach for identifying phenotypes in relation with a Rare Disease (RD) from text. The particularity of this approach is that it enables the identification of complex phenotypes not listed in dictionaries,

by re-using syntactic patterns built by SPARE (introduced in Chapter 4). For identifying phenotypes, we proposed *(i)* to select good quality patterns that are more specifically associated with D-P relationships, and *(ii)* to relax these patterns on the phenotype constraint to enable identifying phenotypes. Applying this identification to the case of RD is of particular importance since it provides a fine-grained description of diseases, which could be used to guide medical diagnosis and clinical care.

Even though databases of D-P relationships, such as Orphadata, are populated carefully by qualified human curators, the scientific literature in this domain evolves fast causing their content becoming rapidly out of date in comparison with what is available in the literature. It is the goal of this chapter to propose an approach to facilitate the extraction of information from biomedical articles related to RD *i.e.,* the extraction of RD phenotypes.

This goal is challenging because the phrasing of RD phenotype is ***complex*** and with a ***high variability***.

One phenotype may be described with several words and then have a complex structure (*e.g., self-mutilation, subclinical defect in pancreatic exocrine function*). This structure may even be similar to a sentence (*e.g., climbing stairs becomes difficult, bone maturation is delayed*). It is consequently challenging to define correctly the boundary of a phenotype in text. Example 5.1.1 is an illustration of a phenotype with a complex phrasing, where the phenotype is located between the two following tags: *<phenotype>* and *</phenotype>*.

Ex. 5.1.1 [*from PMID:21467825*] "*<disease>*Cluster headache*</disease>* is a neurovascular disorder characterized by *<phenotype>*attacks of severe and strictly unilateral pain presenting in and around the orbit and temporal area*</phenotype>*".

The high variability of phenotype phrasing may be at the lexical (*e.g.,* Hypsarrhythmia vs. Hypsarhythmia), syntactic (*e.g.,* Growth delay vs. Delayed Growth) or semantic levels (*e.g.,* Growth delay vs. Growth failure). This causes their recognition even more complex. Examples 5.1.2 and 5.1.3 show two variants phrasing of the same phenotype ("abnormal keratinization" and "keratinization abnormalities").

Ex. 5.1.2 [*from PMID:20857128*] "*<disease>*Darier disease*</disease>* (DD; OMIM 124200) is a rare, autosomal dominant hereditary skin disorder characterized by*<phenotype>* abnormal keratinization *</phenotype>* and *<phenotype>* acantholysis *</phenotype>* ".

Ex. 5.1.3 *[from PMID:10599941]* "Real-time confocal images are illustrative and can be well correlated with known light microscopic phenomena, particularly in the case of *<phenotype>* keratinization abnormalities *</phenotype>* in *<disease>* Darier-White's disease *</disease>*".

Lastly, a phenotype mention can be ambiguous *i.e.,* it may refer to several known phenotypes. For example, the term "cold" may be associated with six distinct UMLS concepts such as "Cold Temperature (C0009264)" and "Cold Sensation (C0234192)" [AL10].

In this chapter, we present a novel method for recognizing phenotypes in relation with a RD in the literature, even when those are complex and rare, and potentially not referenced in phenotype databases or ontologies. This method relies on an initial step of D-P relationship extraction named SPARE* and presented in Chapter 4.

Here we are: *(1)* proposing a method for extracting RD phenotypes from the literature; *(2)* evaluating the validity and novelty of extracted phenotypes in regards to phenotypes listed in specialized ontologies and databases; and *(3)* developing an application for guiding improvements of the content of both the Orphanet encyclopedia and Orphadata.

The rest of the chapter is structured as follows: Section 5.2 presents the state of the art in phenotype recognition from text. Section 5.3 introduces our relation-based approach for identifying complex phenotype candidates, while section 5.4 introduces a method to evaluate the validity and novelty of the identified phenotype candidates by

mapping them to a reference phenotype ontology. Section 5.5 presents the results of this evaluation and Section 5.6 presents an application of our approach for enriching the Orphanet summaries and Orphadata. Section 5.7 discusses our method and their results. Finally, section 5.8 ends the chapter with the conclusion.

5.2 Background on Phenotype Recognition from Text

Named Entity Recognition (NER) aims at automatically associating words and phrases from texts with a pre-defined semantic category from such as gene, disease, phenotype. NER may be followed by a normalization task, which aims at matching recognized entities to a concept in a terminological resource (*e.g.,* UMLS, HPO). Concept recognition (CR) merges NER and normalization in one unique task. It uses terminological resources for looking directly at text for mentions that match the corresponding concepts of these resources.

The problem of phenotype recognition has been studied in a few works in comparison to the recognition of other biomedical entities such as genes or drugs. This may be related to the complexity and the variability of phenotype phrasing.

Figure 5.1 proposes a categorization of phenotype recognition works where some of them distinguish between the NER and normalization tasks and others use CR, integrating both of NER and normalization tasks into one module.

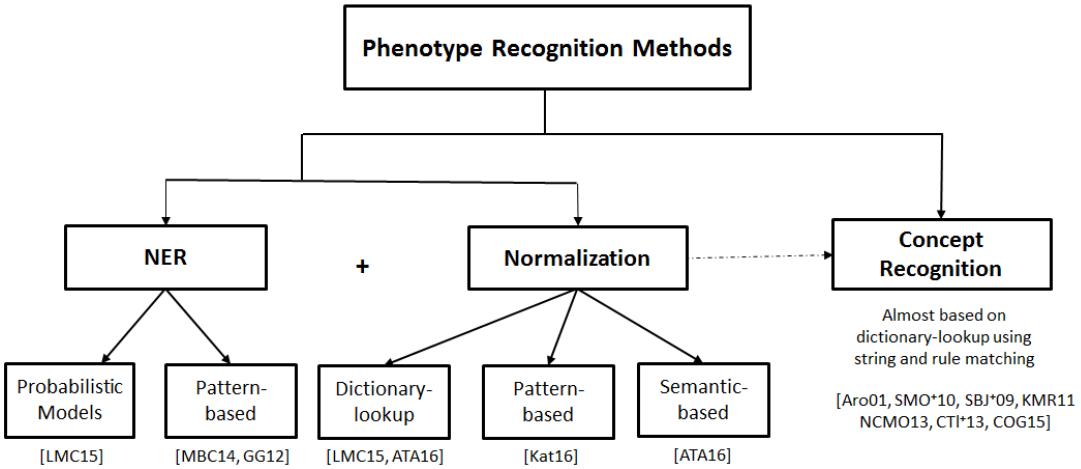


Figure 5.1: The categorization of the related works for phenotype recognition. References in the figure refer to the references cited in Section 5.2.

Both Probabilistic models and Pattern-based methods have been used for phenotype NER task. Leal *et al.* [LMC15] developed disorder recognition module based on Conditional Random Fields (CRF). CRF is used for learning models on biomedical annotated corpus for identifying disorder mentions. Martin *et al.* [MBC14] learned sequential patterns for recognizing symptoms from biomedical texts. Glass et Gliozzo[GG12] proposed a pattern-based approach to extend the coverage of UMLS in the recognition task. They learned patterns from clusters of words by analyzing the internal structure of multi-word terms of each cluster. These patterns are represented by the sequence of clusters representing their words. They are learned from known terms in UMLS and then applied in the identification of new terms with the same structure.

For the normalization task, we distinguish between dictionary-based, pattern-based and semantic-based approaches. Leal *et al.* in [LMC15] developed, besides the recognition module, a normalization module. It searches UMLS concepts and finds the best match to an extracted disorder. The best match is founded with a lexical similarity function using NGram and Levenshtein distances. Kate [Kat16] introduced a solution to

the variations of medical terms that violate the match with terminology concepts when using a string-matching algorithm. He learned patterns by computing edit distance between the medical terms and their known variations in UMLS using Levenshtein distance. This method captures the morphological and typographical variations; but it does not perform any semantic analysis. Alnazzawi *et al.* [ATA16] introduced a method, called PhenoNorm, that automatically maps phenotype mentions to UMLS concepts. To consider the variability of phenotype phrases, PhenoNorm integrates different similarity methods including string-based and semantic similarity (using WordNet) measures.

Most of the state of the art in CR such as MetaMap [Aro01], cTAKES [SMO+10], the NCBO annotator [SBJ+09] and BeCAS [NCMO13], are based on dictionary look-up approaches. Khordad et al [KMR11] relied on existing resources such as MetaMap and HPO for identifying phenotypes in text. Additionally, they implement five simple rules to improve the results of these resources. These additional rules show an increase of 11.87% in F-measure. Alternatively, Collier *et al.* [CTL+13] introduced a hybrid approach for identifying complex phenotype mentions by exploiting the combination of three models based on machine learning, rules and dictionary matching. Machine learning model employs maximum entropy model with beam search using linguistics features; Rule matching model is based on MetaMap; and Dictionary matching model employs the longest matching to map the entity mention to a concept of a terminological resource. Finally, they employ a learn to rank algorithm (SVM-LTR) in order to select the best match; a scoring function is used to score the result of each model. Similarly, Collier *et al.* [COG15] employed SVM-LTR in an ensemble approach for phenotypic recognition. But in this work, SVM-TRL has used to re-rank the results of 4 different CR systems MetaMap, cTAKES, NCBO annotator and BeCAS.

In this thesis, we used NER recognition module that relies on a pattern-based approach (described later in Section 5.3). We learn patterns for recognizing phenotypes that are in a relation with a RD. For the normalization, we use a dictionary-based approach using MetaMap. In case a phenotype is not recognized by MetaMap, a semantic-based approach (described later in Section 5.4) is used. Section 3.1.3 in Chapter 3 presents background information about the semantic similarities and compositional semantics. In this chapter, we are interested in the distributional measures as we deal with complex phenotype description coming from a medical corpus, but not always referenced in ontologies. Also, we use a compositional semantics model because we aim at finding the similarity between complex phenotypes that may be composed of many words.

5.3 SPARE* for Identifying Phenotypes Candidates

Syntactic patterns, built with the SPARE method, can be used for identifying phenotype candidates in text that are not identified by classical NER tools such as MetaMap. Most of these phenotype candidates correspond to variants of phenotypes described in ontologies, *i.e.,* phenotype classes. One problem is then to identify, through a normalization process, the adequate phenotype class in the ontology to assign to each phenotype candidate. Next section presents a compositional semantics model to normalize the phenotype candidates identified from text with phenotypes defined in the HPO ontology (refer to Chapter 1, section 1.1 for more details about the HPO ontology).

In addition to *support* and *ppv* metrics, previously defined in Chapter 4, we define a third quality measure for syntactic patterns: their *specificity*. For example, the *specificity* of a pattern measures how much a pattern is specific to D-P relationships rather than other relationships such as Disease–Drug or Disease–Gene. To compute the specificity, we use a set of sentences, where not only diseases and phenotypes were annotated, but also genes, treatments and living beings, as defined by the UMLS semantic types. Computing the specificity also requires *relaxing* the definition of patterns by enabling their second entity to match any type of entity, instead of matching with phenotypes only. As a result, relationships that match with one pattern may be a D–P or "D–*something else*" relationship. In our pattern definition, the general string PHENOTYPE is then replaced by "?".

Definition 20 *(Relaxed pattern)*
A pattern is relaxed on a constraint by replacing the constraint node of the pattern by "?" to enable the pattern to match any string with this node.

Definition 21 *(Pattern specificity)*
Specificity measures how a relaxed pattern is specific for a kind of relationships (*e.g.,* D-P relationships). In other words, it is the probability that the constraint node "?", in the relaxed pattern, matches with the looked-up entity (*e.g.,* phenotype) rather than any other entities (*e.g.,* gene, drug). The specificity of the pattern p is defined as:

$$specificity_p = \frac{|TR_p^{D-P}|}{|TR_p^{D-?} \cup FR_p^{D-?}|} \tag{5.1}$$

where TR_p^{D-P} is the set of true D-P relationships extracted by the pattern p (denoted simply by TR_p in (1) and (2)) and $TR_p^{D-?} \cup FR_p^{D-?}$ is the set of all (true and false) relationships that are extracted by the relaxed pattern, including for example D–P, disease–gene, disease–treatment and disease–living being relationships.

Relaxed patterns may be noisy and bring wrong relationships. Therefore, the specificity is used to select the patterns that are the most specific to a kind of relationships such as D–P relationships. We propose that, if the specificity of a pattern is greater than or equal to a minimum specificity denoted $min\_specificity$, then the pattern is specific to this kind of relationships. We rely on the specific patterns learned for D–P relationships to capture correct D–P relationships even when no phenotype entity has been recognized by NER tools such as MetaMap, consequently providing the capability to identify complex and novel phenotypes associated with RDs.

The $specificity$ of patterns is different from the ppv in that ppv qualifies D-P relationship patterns, whereas, $specificity$ qualifies relaxed patterns. For example, if a pattern p_i matches with 10 D–P relationships, from which only 7 are true. Then, ppv_{p_i} is $7/10$. If the pattern p_i is relaxed, it matches with 5 additional D–*something else* relationships. Then, $specificity_{p_i}$ is $7/(10+5)$.

Positive patterns (with a $support$ and a ppv higher than a specified threshold) are relaxed on the phenotype constraint, meaning that one entity must be annotated as a disease, but there is no requirement for the second entity to be annotated as a phenotype. The resulting set of positive patterns is reduced to specific patterns, *i.e.,* patterns with a specificity higher than a specified threshold named $min\_specificty$. Specific patterns are then used to identify D-P relationships, which involve a phenotype that was missed by a NER tool. For this reason, sentences of the corpus annotated with one (or more) disease(s) are transformed in DGs, then considered for *pattern matching*.

During pattern matching, the word that matches the node of the second entity (not constrained) is considered to be a phenotype candidate. In other terms, we rely on the specificity of the pattern for identifying novel phenotypes associated with a disease. To be more precise, this second entity is considered to be a phenotype if this word is a leaf of the DG, but is only considered as the "head" of a more complex phenotype description if it is not a leaf. In this later case, we extract the complete description of the phenotype by exploring the subtree that has, as a head, the node that matches as a phenotype. For example, the positive pattern presented in Figure 5.2 may be relaxed to generate the pattern presented in Figure 5.3. If this new pattern is associated with a high specificity, it is considered as a specific pattern and may be used on the sentence Ex. 5.3.1 for pattern matching. Here, the sentence has been annotated with two diseases, but no phenotype. The pattern matches the sentence and the word "lack" matches as the phenotype. Exploring the subtree represented in Figure 5.4(b) enables us to reconstruct the full phenotype description "a lack of specific lysosomal enzymes", which is subsequently used to identify the following relationship: <Mucopolysaccharidose, a lack of specific lysosomal enzymes>. This illustrates the ability of our method to extract D–P relationships that include complex and non-referenced phenotypes.

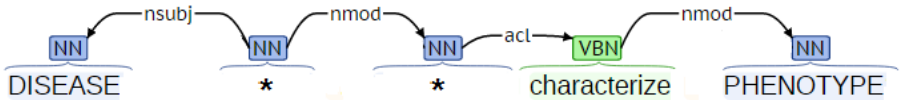


Figure 5.2: Example of DG pattern we recalled from Figure 4.7 in Chapter 4.

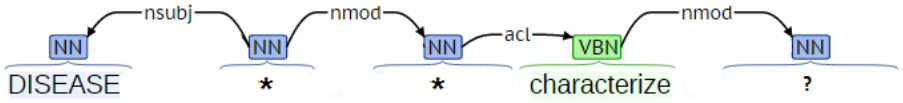
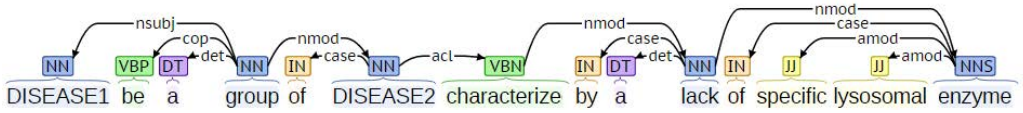
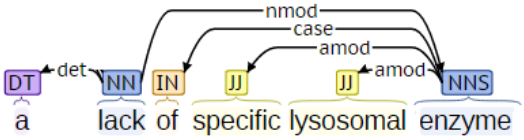


Figure 5.3: The relaxed pattern generated from the pattern of Figure 5.2.

Ex. 5.3.1 "<*disease*>Mucopolysaccharidoses</*disease*> are a group of <*disease*>inherited disorders</*disease*> characterized by a lack of specific lysosomal enzymes"



(a)



(b)

Figure 5.4: Example of phenotype identification: Sentence Ex. 5.3.1 is transformed in a DG (a) that matches the syntactic pattern of Figure 5.3. Once relaxed, this pattern points at the root of a sub-tree (b) used to extract a phenotype description.

Finally, the phenotype identified by SPARE and the RD itself are used to build a D-P relationship pair. Then, SPARE*, with the selected combination SPARE_AND_SVM_Unknown (as preferred in Section 4.3.3 of Chapter 4), is used to classify the pair and to associate (or not) the extracted phenotype to the RD. Hereafter, we refer to the phenotypes identified by SPARE* as SPARE* phenotypes.

5.4 Phenotype Normalization Using Compositional Semantics

The method presented in Section 5.3 identifies phenotype candidates that are not recognized by NER (*e.g.*, MetaMap). These candidates may be referenced in phenotype ontologies but mentioned with a different phrasing, or may not be referenced at all. We describe here an original method that aims at comparing phenotype candidates extracted from text to those listed in ontologies, such as HPO. We define a set of five mapping categories to distinguish between various levels of mappings. For instance, a SPARE* phenotype may match exactly with an HPO phenotype, then its mapping belongs to the "Exact" category; or it may only match to a term in HPO that is more general than itself, their mapping consequently belongs to the "more general" category. Table 5.1 lists and describes the five mapping categories.

We propose a mapping strategy based on a compositional semantic space model, completed with few rules that we defined manually. First, we build a dictionary containing all words, named target words, that compose

Mapping Category	Description
Exact	SPARE* phenotype is matching exactly with HPO phenotype
More General	SPARE* phenotype is a more general term than HPO phenotype
More Specific	SPARE* phenotype is a more specific term than HPO phenotype
Sibling	SPARE* phenotype and HPO phenotype are siblings
None	SPARE* phenotype is not mapped with HPO phenotype

Table 5.1: Mapping categories and their description. Every phenotype we extracted from text (named SPARE* phenotype) is compared for mapping to phenotypes defined in the HPO ontology (named HPO phenotype). The result of the comparison falls in one of these categories.

all considered phenotype descriptions including both phenotypes extracted from text (SPARE* phenotype) and phenotype defined on the HPO ontology (HPO phenotype). Second, we build semantic vectors that represent the semantics of all target words. Third, a compositional semantic model is employed for generating a semantic representation vector for each considered phenotype. This representation enables evaluating a similarity between terms. Then, we evaluate a cosine similarity between every pair made of one SPARE* phenotype and one HPO phenotype by using their semantic representation vectors. Finally, the best similarities of each SPARE* phenotype are used to propose a mapping to HPO and assign a mapping category.

5.4.1 A Dictionary of Target Words

SPARE* and HPO phenotypes are grouped to build a dictionary. This dictionary is obtained by tokenizing and lemmatizing SPARE* and HPO phenotype strings. We used an arbitrary, but classical, list of stop words[6], to clean our dictionary from them. Remaining tokens are considered as target words for building the proposed semantic space model. To each target word, we associate a set of synonyms obtained from the union of synonyms defined in WordNet[7] [Fel98] and MeSH[8].

5.4.2 Vector Representation of Target Words

For each target word, we generate a vector named "context vector" representing its compositional semantics. The generation of the model follows two steps: *(1)* generating the context vector associated with each target word; *(2)* building the semantic space model, *i.e.*, selecting a subset of basis elements from the full set of basis elements available in context vectors and associated with target words.

First, a context vector for each target word is built. This vector represents the semantic of the word because its n-dimensions encompass the contextual information of the word. Each of its dimension represents a feature or basis element such as a word, lemma, POS or a word-dependency relation pair in relation with this target word. A value is assigned to evaluate the relation between a target word and a basis element in the vector. This value represents the contribution of the basis element to the context of the target word. To build this vector, a Dependency Graph (DG) dataset of all sentences of our corpus is first computed. Then, for each target word, candidate basis elements are extracted from the DGs that mention the target word. Basis elements are the words in relation with the target words in at least one DG of our set. Here, we consider as a basis element, every word in a DG that is in a distance up to L_{max} to the target word. Arbitrary, we fix L_{max} at 2 and use the lemma form of words as basis elements.

[6]The list of stop words removed is at: `https://sourceforge.net/projects/spare2015/files/StopWords`

[7]We used the version 3.0 of WordNet.

[8]We used the MeSH vocabulary of the UMLS 2015AA, released 06/09/2015, downloaded from `https://bioportal.bioontology.org/ontologies/MESH`.

Then, a value for each basis element is computed with regard to the corpus. As recommended in [PL07], we use a function based on the length of the dependency path, $L(dp_{t,i})$, between the target word t and the basis element i to quantify this value, denoted $V_t[i]$. This function consider indeed the inverse of the length of the dp in each DG. For example, if considering the DG in Figure 4.4(a) and the relationship between words "Botulism" and "paralysis", then $L(dp_{t,i})$ is $1/2$. Usually, there are several DGs with a $dp_{t,i} < L_{max}$. Hence, the function accumulates the value of each $dp_{t,i}$ in the final value. For example, if the length of the dp between a target word and a basis element is 2 in two different DGs, then 1 is assigned to $V_t[i]$ (1/2 + 1/2). Formally,

$$V_t[i] = \sum \frac{1}{L(dp_{t,i})},\ \forall\ dp_{t,i} \in \mathcal{G},\ L(dp_{t,i}) \leq L_{max} \tag{5.2}$$

where $\mathcal{G}$ is the set of DGs of our corpus and $dp_{t,i}$ is a dependency path between the target word *t* and the basis element *i*.

Once context vectors are computed, each vector V_t may have a different size. One way for normalizing the dimension of all these vectors is to combine all basis elements of all vectors in order to consider the full context of the set of target words. With this solution arises two drawbacks: the large size and the sparsity of vectors. Another way is to reduce the size of vectors by using only the most k-frequent basis elements of all target words. We use $k = 2000$ as recommended in [PL07] for an optimal dependency-based model. Then, the vector values $V_t[i]$ are normalized by dividing them by the sum of all vector values as formalized in 5.3:

$$\overline{V_t[i]} = \frac{V_t[i]}{[\sum_{j=1}^{k}]V_t[j]} \tag{5.3}$$

Finally, we obtained for each target word a vector composed of 2000 basis elements, with normalized values.

5.4.3 Compositional Representation of Complex Phenotype Descriptions

Phenotype descriptions are complex terms in their linguistic representation, especially for RD phenotypes. Phenotype descriptions mostly consist in multiple word compositions, where each word contributes to the overall meaning of the phenotype. To build a semantic representation vector of a phenotype, we used an algebraic composition method because it is simple to implement and to compute, and achieves results close to RAE (Recursive AutoEncoder) [BL12]. We adopted and implemented an average composition method that is similar to additive composition model, but includes a normalization by getting the average value instead of the sum. It achieves better results than additive and multiplicative methods because the number of words in each phenotype description differs (*i.e.* variability of phenotype phrasing), enabling a fine-grained description.

5.4.4 Semantic Similarity between Two Phenotypes

For computing the semantic similarity between two phenotypes, we used the *cosine similarity* defined by

$$cos(x, y) = \frac{x \cdot y}{\| x \| \| y \|} \tag{5.4}$$

where x and y are the vectors of the two compared phenotypes. Because a SPARE* phenotype can be closely similar to one or more HPO phenotypes, we keep for each SPARE* phenotype, the top 10 closest HPO phenotypes. This is also because a SPARE* phenotype may be related with the first best similar HPO phenotype with a "Sibling" relation, while it is related with the second best similar HPO phenotype with a "More Specific" relation. As we prefer the "More Specific" relation than the "Sibling" relation, we choose the second best similar one.

5.4.5 Semantic Similarity Rules for Phenotype Mappings

Every mapping between a SPARE* and a HPO phenotype falls in one of the mapping categories listed in Table 5.1. The mapping rules defined in Algorithm 1 are applied to achieve this assignment.

A SPARE* phenotype has an "Exact" mapping with a HPO phenotype if the similarity value between them is 1 (calculated by Eq. (5.4)) or all tokens of SPARE* phenotype are the same as all tokens of HPO phenotype (*i.e.,* if they have the same number of words and the different words have the same steaming form[9]). Else the two phenotypes do not have an "Exact" mapping and the next rule is considered for assigning a different mapping category.

A SPARE* phenotype may be related to a HPO phenotype with an "is-a" relationship. This means that the SPARE* phenotype is more specific or more general than a HPO phenotype. If the tokens of a HPO phenotype are included in (*i.e.,* are a subset of) the tokens of a SPARE* phenotype, then the SPARE* phenotype is more specific than the HPO phenotype and the "More Specific" category is assigning to the mapping. Else, if the tokens of SPARE* phenotype are included in (*i.e.,* are a subset of) the tokens of a HPO phenotype, then the SPARE* phenotype is more general than the HPO phenotype and the "More General" category is assigning to the mapping between them.

A SPARE* phenotype may be a "sibling" of a HPO phenotype. This occurs when some tokens of a SPARE* phenotype, but not all, are equal to some tokens, but not all of a HPO phenotype [10] and these tokens must include the head of subtree defining SPARE* phenotype (see subsection 5.3).

Algorithm 1 Mapping Rules

Input:
p_1 # a SPARE* phenotype
p_2 # a HPO phenotype
Output:
$Mapping\_Category$

```
1: if sim(p1, p2)=1 OR (length(p1)=length(p2) AND diff_have_same_stem(p1, p2)) then
2:     Mapping_Category = "Exact"        # p1 is mapped exactly with p2
3: else
4:     if tokensOf(p2) ⊂ tokensOf(p1) then
5:         Mapping_Category = "More Specific"        # p1 is more specific than p2
6:     else
7:         if tokensOf(p1) ⊂ tokensOf(p2) then
8:             Mapping_Category = "More General"        # p1 is more general than p2
9:         else
10:            if headTokensAreMatched(p1, p2) then
11:                Mapping_Category = "Sibling"        # p1 and p2 are siblings
12:            else
13:                Mapping_Category = "None"        # p1 is not mapped with p2
14:            end if
15:        end if
16:    end if
17: end if
```

SPARE* phenotype can be mapped to more than one HPO phenotypes. In our case, we choose the first top closest HPO phenotypes. To resolve this ambiguity, we keep only from the top 10 the most interesting mapping. For example, we firstly consider the "Exact" mapping. In case there is no "Exact" mapping, we consider then the "More Specific" and "More general" respectively.

[9] Porter stemmer is used for the steaming of all words

[10] In the case of some tokens of one phenotype are equal to all tokens of another phenotype, then the "More General" or "More Specific" mapping category is used.

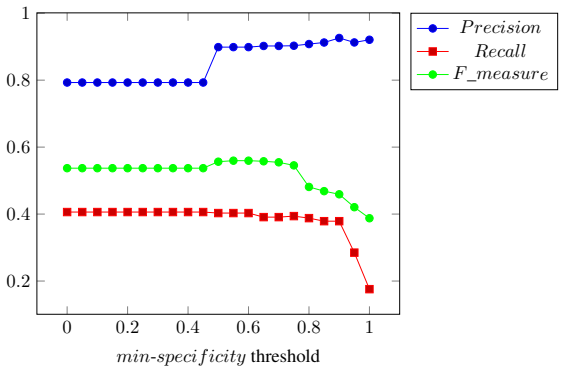


Figure 5.5: The effect of specificity threshold on $precision$, $recall$ and $F\text{-}measure$ values.

Finally, a SPARE* phenotype does not map with HPO phenotype if all of the previous rules can not apply. A SPARE* phenotype that does not map with any HPO phenotype may be new and requires to be validated by an expert of the domain.

5.5 Experiments and Results

This section presents our experiments and results for identifying phenotypes in text.

5.5.1 Corpus

We used the same RD corpus of 121,796 PubMed abstracts about 457 distinct RDs presented in Chapter 4, section 4.2. The 121,796 abstracts were split into 907,088 sentences using LingPipe. Each sentence has been annotated by MetaMap with the concept associated with the following semantic types: Disease or Syndrome (T047), Sign or Symptom (T184), Abnormalities (T019, T020, T037, T048, T049, T050, T190), Therapeutic or Preventive Procedure (T060, T061), Genes (T028, T045) and Living Beings (T005, T007, T004). Finally, sentences that are not annotated with at least one disease (T047) are filtered out, to obtain 301,599 sentences.

5.5.2 Automatic Selection of Thresholds

We propose to fix the specificity minimal threshold ($min\_specificty$) similarly to the minimal thresholds of the $support$ and ppv (min_sup and min_ppv), see Chapter 4, section 4.4.1. As a recall, for min_sup and min_ppv, we divided a set of manually annotated sentences in 2: one set of 90% of the sentences to make a learning set; and one set of 10% of the sentences to make a testing set. Patterns are built from the learning set and then applied on the testing set for RE. $F\text{-}measures$ are evaluated for various subsets of patterns when min_sup and min_ppv are progressively increased.

For computing the pattern specificity, the 301,599 sentences that contain at least one disease and another UMLS entity from our selection are used. Then, all previously selected positive patterns (235 patterns, see Section 5.5.2 in Chapter 4) are relaxed on the phenotype constraint and then applied for pattern matching to the DGs of these sentences. The specificity of each pattern is computed (see formula 5.1). Then, $F\text{-}measures$ are computed for subset of specific patterns while increasing $min\_specificty$. Finally, we selected $min\_specificty$ that achieves the best $F\text{-}measure$. Figure 5.5 shows that $min\_specificty = 0.5$ achieves the best $F\text{-}measure$ of 0.65.

5.5.3 Phenotype Candidates

For identifying RD phenotype candidates from new texts, we learn D-P specific patterns from the 2,341 sentences. We used $min\_support = 2$, $min\_ppv = 0.5$ (as explained in Chapter 4, section 4.4.1) and $min\_specificity = 0.5$ (as explained in the previous subsection 5.5.2) as they give the best $F\text{-}measure$. These thresholds enable selecting 217 specific patterns[11]. Figure 5.6 proposes 10 examples of them.

We applied the 217 patterns to the corpus of 301,599 sentences with at least one disease for pattern matching. The extracted relationships are divided into two groups: 4,886 D–P relationships where the phenotype was previously recognized by MetaMap; and 6,572 where the phenotype was not recognized by MetaMap. After manual evaluation, in these two groups our method achieved respectively 0.91 and 0.83 precision. The number of distinct phenotypes in the first group is 1,457 (recognized by MetaMap, thus mapped to UMLS concepts) and in the second group is 3,821 (unrecognized by MetaMap and potentially new).

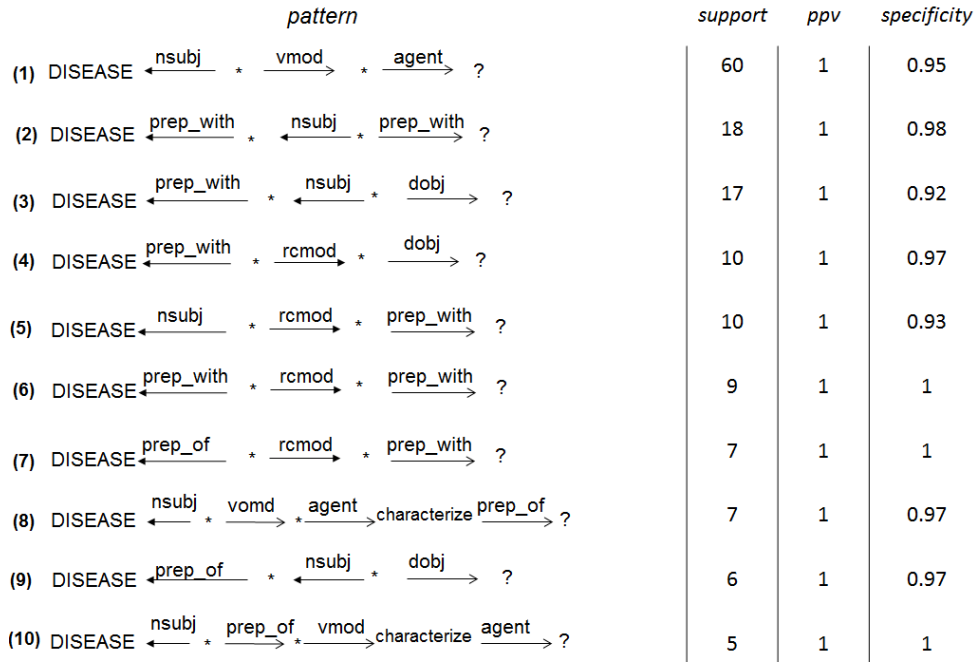


Figure 5.6: 10 examples of specific patterns, along with their *support*, *ppv* and *specificity*.

5.5.4 Normalization

The mapping approach described in section 5.4 has been applied between 3,821 phenotypes extracted by SPARE⋆ and the 11,021 phenotypes defined in HPO. We used in this work the version 1.2, releases 2015-08-17 of HPO. For each SPARE⋆ phenotype, the closest HPO phenotypes are kept, and then the mapping rules are applied to assign the most appropriate mapping category. As shown in table 5.2, 3,296 SPARE⋆ phenotypes are mapped to HPO phenotypes within one of the four proposed, while 525 SPARE⋆ phenotypes do not[12]. Table 5.3 shows examples of "Exact", "More Specific", "More General", "Sibling" and "None" mappings. For instance, "severe mental retardation" and "Severe mental retardation" have "Exact" mapping as their similarity is 1, also

[11] The list of the 217 patterns is available at `http://sourceforge.net/projects/spare2015/files/217Patterns`.

[12] The list of the 3,296 mappings is available at `https://sourceforge.net/projects/spare2015/files/SPAREstar_HPO_MappingList.rar`

Mapping Category	Mappings Count
Exact	346
More Specific	214
More General	229
Sibling	2,507
All Mappings	**3,296**
None (No Mappings)	**525**
Total	**3,821**

Table 5.2: The results of mapping 3,821 SPARE⋆ phenotypes to 11,021 HPO phenotypes

SPARE⋆ Phenotype	HPO Phenotype	Similarity	Mapping Type
severe mental retardation	Severe mental retardation	1	Exact
progressive neurological deterioration	Progressive neurologic deterioration	0.99	Exact
bilateral renal cell carcinoma	Renal cell carcinoma	0.994	More Specific
severe juvenile-onset osteoporosis	Severe osteoporosis	0.945	More Specific
reticular skin hyperpigmentation	Skin hyperpigmentation	0.974	More Specific
the basal ganglia	Basal ganglia calcification	0.98	More General
an increased risk of cancer	increased risk of pancreatic cancer	0.99	More General
early death	Death in early childhood	0.988	More General
malignancy	Multiple cutaneous malignancies	0.959	More General
vertebral abnormalities	Vertebral anomalies	0.983	Sibling
disorders of fructose metabolism invariably	Impairment of fructose metabolism	0.971	Sibling
retarded bone maturation	Delayed bone maturation	0.981	Sibling
skeletal defects	Skeletal abnormalities	0.988	Sibling
skeletal deformities	Skeletal abnormalities	0.977	Sibling
severe growth retardation	Marked growth retardation	0.987	Sibling
pancreatic exocrine deficiency	Abnormal exocrine pancreatic function	0.985	Sibling
stereotypical hand movements	Stereotypical motor behaviors	0.95	Sibling
body asymmetry	Asymmetry of the size of ears	0.959	Sibling
early onset diabetes	Maternal diabetes	0.956	Sibling
alterations in the structure	Abnormality of the paralabial region	0.977	None
microthrombocytopenia	eczema	0.641	None

Table 5.3: Examples of mappings between SPARE⋆ phenotypes and HPO phenotypes, their similarity values and their mapping type.

"progressive neurological deterioration" and "Progressive neurologic deterioration" have "Exact" mapping because "neurological" and "neurologic" have the same stem.

5.6 Application: Enriching Orphanet and Orphadata

In this section, we present an experiment that evaluates the use of SPARE⋆, associated with our normalization strategy to guide the update of RD resources Orphanet and Orphadata (using knowledge resources in the literature). We limited this study to only 16 RDs, chosen by an expert for their heterogeneity in terms of poor *vs.* rich documentation in Orphanet. The purpose of the study is to evaluate the shift between the D–P relationships reported in Orphanet summaries, the D-P relationships extracted form PubMed abstracts using SPARE⋆ and those listed in Orphadata.

Table 5.4 lists the 16 RDs included in this study, their identifiers in Orphanet, the number of associated PubMed

Orpha Number	Disease Name	#Abstracts	Last Update
ORPHA336	Fibromuscular dysplasia of arteries	54	Dec. 2014
ORPHA364	Glycogen storage disease due to glucose-6-phosphatase deficiency	1,344	Nov. 2010
ORPHA365	Glycogen storage disease type 2	2,829	Jan. 2014
ORPHA399	Huntington disease	12,078	Jan. 2011
ORPHA481	Kennedy disease	29,944	Jul. 2011
ORPHA511	Maple syrup urine disease	1,279	Apr. 2014
ORPHA803	Amyotrophic lateral sclerosis	27,706	May 2011
ORPHA910	Xeroderma pigmentosum	7,843	May 2011
ORPHA1002	Cluster headache	3,339,227	Jul. 2008
ORPHA1267	Botulism	3,623	Jan. 2011
ORPHA1667	Wolcott-Rallison Syndrome	391	Feb. 2011
ORPHA1727	22q11.2 microduplication syndrome	212	Feb. 2011
ORPHA3463	Wolfram Syndrome	397,218	Sep. 2014
ORPHA163966	X-linked dominant chondrodysplasia, Chassaing-Lacombe type	30	Feb. 2011
ORPHA168972	Kahrizi syndrome	26	–
ORPHA238446	15q11-q13 microduplication syndrome	3	Mar. 2011

Table 5.4: List of the 16 rare diseases used in our case study for comparing phenotypes extracted with SPARE* to those listed in Orphanet and Orphadata. The third column shows the number of PubMed abstracts obtained by a simple query to PubMed, similar to those defined in Chapter 4, section 4.2. The last column provides the date of the last update of the Orphanet summary on the 07/11/2015. No date is provided in Orphanet for the last update of the summary of the Kahrizi syndrome (ORPHA168972).

abstracts[13], and the date of the last update of their summary in Orphanet at the 07/11/2015.

5.6.1 Data Preparation

We obtained *(1) Orphanet D–P relationships*, by manually annotating the mentions of phenotypes in the 16 RD summaries from Orphanet web site. All summaries were annotated manually by two physicians. We developed a software to support our experiment. It particularly enables to visualize the annotations used by the experts. Considering the Kennedy disease as an example, its Orphanet summary is available in appendix A in Figure A.2. Figure C.10 shows a screenshot of our tool where annotated phenotypes are colored in red. The user can go to the online version of the summary with the "Go online !" button.

We extracted *(2) PubMed D–P relationships* by applying our SPARE* method to a set of PubMed abstracts related to considered RDs. Table 5.4 lists the number of abstracts retrieved for each of the 16 RDs. We used the positive patterns learned from our manually annotated corpus to extract D-P relationships. Because the abstracts may contain mentions of other RD, we keep only the relationships that are related to the considered 16 RDs.

We obtained *(3) Orphadata D–P relationships* by extracting the values associated with the attribute "clinical sign" of each RD in Orphadata. This data are publicly available at `http://www.orphadata.org/cgi-bin/inc/product4.inc.php`.

Table 5.5 provides the number of phenotypes we were able to identify respectively from Orphanet, Orphadata and PubMed. For example, the Orphanet summary of Kennedy disease contains 40 phenotypes (annotated by our experts), while Orphadata contains only 12 and 100 are extracted by SPARE* from the literature.

[13]These abstracts were retrieved from PubMed in 20/10/2015, by a simple query similar to those defined in Chapter 4, Section 4.2.

Disease Name	Orphanet	Orphadata	SPARE⋆
Fibromuscular dysplasia of arteries	15	0	9
Glycogen storage disease due to glucose-6-phosphatase deficiency	34	13	0
Glycogen storage disease type 2	11	25	402
Huntington disease	37	8	1163
Kennedy disease	40	12	100
Maple syrup urine disease	23	29	158
Amyotrophic lateral sclerosis	10	0	1033
Xeroderma pigmentosum	32	60	13
Cluster headache	11	21	85
Botulism	26	18	0
Wolcott-Rallison Syndrome	29	38	32
22q11.2 microduplication syndrome	8	40	0
Wolfram Syndrome	41	49	60
X-linked dominant chondrodysplasia, Chassaing-Lacombe type	29	26	0
Kahrizi syndrome	15	0	0
15q11-q13 microduplication syndrome	25	18	0

Table 5.5: Numbers of phenotypes associated with each of the 16 considered RDs according to Orphanet summaries, Orphadata and SPARE⋆. For the four diseases in bold, no UMLS CUIs exist, and we are consequently unable to extract any associated phenotype with SPARE⋆.

5.6.2 Comparing Phenotypes from Orphanet, Orphadata and PubMed

The phenotypes extracted from PubMed are mapped with the phenotypes in the Orphanet summary and Orphadata. First, a composition semantic vector is built for each phenotype by using the compositional semantics space model described in Subsection 5.4. Then, the semantic similarity is computed for every possible pair made of one SPARE⋆ phenotype and one Orphanet phenotype; made of one SPARE⋆ phenotype and one Orphadata phenotype; or made of one Orphanet phenotype and one Orphadata phenotype. Finally, the mapping rules defined in Algorithm 1 are applied to assign a mapping category from the list presented in table 5.1, but here the reference ontology is replaced by reference phenotypes as defined either by Orphanet or Orphadata.

SPARE⋆ phenotypes that do not map to any Orphanet or Orphadata phenotypes are supposed to be either new or noisy. In turn, these phenotypes are tried to be mapped to HPO phenotypes using the same strategy. Phenotypes that can be mapped to HPO phenotypes are proposed as new and recommended to enrich Orphanet summaries and Orphadata. They are issued along with the PMID and the sentence they have been extracted from.

5.6.3 Improving Orphanet Summaries

Orphanet summaries can be enriched by using both phenotypes extracted by SPARE⋆ and Orphadata phenotypes. To assess these potentials, we mapped the latter to Orphanet summary phenotypes. Table 5.6 presents the average numbers of phenotypes for the 16 RDs for each “Exact”, “More Specific”, “More General”, “Sibling” or “None” mapping. It presents also their average number of phenotypes that are potentially new.

Compared Resources	Exact	More Specific	More General	Sibling	None	Potentially New
SPARE⋆-Orphanet	5.19	19.56	1.94	60.06	179.75	128.81
Orphadate-Orphanet	3.38	1.06	2.25	6.19	12.63	16.5
SPARE⋆-Orphadata	2.31	12.63	0.63	32.44	171.93	189.75
Orphanet-Orphadate	3.5	0.94	3.06	7.13	12.31	14.63

Table 5.6: The first two rows present the average numbers of SPARE⋆ and Orphadata phenotypes which have "Exact", "More Specific", "More General", "Sibling" or "None" mappings to Orphanet summary phenotypes and potentially new phenotypes. The last two rows present the average numbers of SPARE⋆ and Orphanet phenotypes which have "Exact", "More Specific","More General", "Sibling" or "None" mappings to Orphadata summary phenotypes and potentially new phenotypes. The complete list of these mappings is available at `https://sourceforge.net/projects/spare2015/files/16RD_MappingList.rar`

ID	PMID	Sentence	Disease
1	18945802	Neuroacanthocytosis is a rare hereditary disorder characterized by involuntary choreiform movements.	{}
2	22468686	Canavan disease is a severe autosomal recessive leukodystrophy characterized by macrocephaly, ataxia, severe motor and mental retardation, dysmyelination, and progressive spongial atrophy of the brain.	{macrocephaly, severe motor retardation, mental retardation, dysmyelination, progressive spongial atrophy of the brain}
3	21931045	X-linked dominant chondrodysplasia punctata, also known as Conradi-Hunermann-Happle syndrome, is a rare skeletal dysplasia characterized by,short stature, craniofacial defects, cataracts, ichthyosis, coarse hair, and alopecia.	{short stature, craniofacial defects, cataracts, ichthyosis, alopecia}
4	24008937	Late-onset Pompe disease is a progressive metabolic myopathy caused by decreased activity of the enzyme acid alpha-glucosidase (GAA), which gives rise to reduced degradation and, later accumulation of glycogen in the lysosomes and cell cytoplasm.	{}
5	20443038	Both the myotonic dystrophy type 1 (DM1) and the X-linked dominant Charcot-Marie-Tooth disease (CMTX1) are well-established inherited neuromuscular,disorders characterized by progressive weakness and atrophy of the distal limb muscles.	{atrophy of the distal limb muscles}
6	9054082	Krabbe's disease, globoid cell leukodystrophy, is a rare autosomal recessive demyelinating neurodegenerative disease caused by reduced activity of the lysosomal enzyme galactosylceramide beta-galactosidase which is involved in myelin metabolism.	{}

Table 5.7: Example of sentences matched with the pattern of Figure 4.14. This table presents the sentence and the D-P relationship extracted by the pattern. The empty set $\{\}$ means that no relationship has been extracted.

5.6.4 Improving Orphadata

Orphadata may be enriched similarly by using both SPARE⋆ and Orphanet summary phenotypes. To assess these potential, we mapped these phenotypes to Orphadata phenotypes. Table 5.6 presents the average numbers of the phenotypes, for the 16 RD, that have "Exact", "More Specific", "More General", "Sibling", "None" or "Potentially New" mappings.

5.7 Discussion

The originality of the SPARE method relies on measuring how syntactic patterns between diseases and phenotypes are specific to D-P relationships. Using highly specific patterns allow us to consider the case where phenotypes are not recognized by NER tools, which consequently offers the opportunity to discover new phenotype descriptions that can be potentially rare and complex.

Using DGs, in our pattern-based method SPARE, shows its ability to identify the complex phenotypes by considering the complete subtree headed by the node recognized by SPARE as a phenotype (see subsection 5.3).

We recall from Table 4.8 in Chapter 4 that the SPARE pattern of Figure 4.14 extracts D-P relationships from sentences with different phrasing. By relaxing this pattern on the phenotype constraint, it identifies novel phenotypes that are not recognized by MetaMap. Table 5.7 lists these sentences again and presents the novel phenotypes identified by the relaxed SPARE pattern.

In SPARE, the choice of $min\_specificity$ has important consequences on the results of the RD phenotype recognition. Figure 5.5 shows how the results in terms of $precision$, $recall$ and $F\text{-}measure$ vary when the $min\_specificity$ threshold is changed. We observe relatively few evolution of the F-Measure. As shown in

Figure 5.5, we chose $min\_specificity = 0.5$ because it achieves the best F-Measure. The result of F-Measure is constant when $min\_specificity$ between 0 and 0.45 because the number of patterns in this interval is the same.

Our NER recognition module relies on SPARE*. We learn specific patterns for recognizing phenotypes that are in a relation to RD. The SVM is used to classify the extracted RD phenotypes. For normalization, we use a dictionary-based lookup approach using MetaMap. For unrecognized phenotypes by MetaMap, we employ a compositional semantics approach. This approach handles the variations of phenotype mentions (morphological, syntactic and semantic) by learning its semantic from a large corpus and using synonyms from WordNet and MeSH. To the best of my knowledge, this is the first work that combines both a dictionary-based approach and a compositional semantics approach for normalizing complex phenotypic data.

The ability of SPARE* to propose phenotypes that are not listed in Orphanet summaries or in Orphadata has been assessed for 16 RDs. It successfully identifies phenotypes that we propose to be validated by RD experts, in an interactive approach. However, some of these RDs do not have a UMLS CUI (*e.g.,* diseases with Orpha numbers ORPHA1727, ORPHA163966, ORPHA168972 and ORPHA238446). In this case, SPARE* fails at extracting phenotypes for them. In addition, our SPARE* method also fails at extracting phenotypes for "Glycogen storage disease due to glucose-6-phosphatase deficiency" and "Botulism" because no SPARE pattern matches to them. This may be due to the limited size of the corpus we considered for pattern learning.

We consider phenotypes that do not have any mapping and phenotypes that have only "Sibling" mappings as potentially new phenotypes. If considering the Kennedy disease (ORPHA481) as an example, our approach suggests a list of 85 SPARE* phenotypes (*e.g.,* "muscle dysfunction", " adult-onset muscle weakness", "Difficulties in climbing stairs") and a list of 10 Orphadata phenotypes (*e.g.,* "Abnormal gait", "Movement disorder", "Hypotonia"). Then, following our approach, these phenotypes are potentially new and may be added to Orphanet. Also, our approach suggests a list of 94 SPARE* phenotypes (*e.g.,* "tremor", "weakness of limb", "swallowing impairment", "motor neuron degeneration") and a list of 30 Orphanet phenotypes (*e.g.,* "fatigue", "dysarthria", "dysphonia") that are potentially new and may be added to Orphadata. The complete list of suggestions is available at `https://sourceforge.net/projects/spare2015/files/16RD_MappingList.rar`

5.8 Conclusion

In this chapter, we presented the use of our RE method, SPARE* presented in Chapter 4, for phenotype recognition. We relaxed the SPARE patterns and introduced a specificity measure of pattern in order to learn patterns that are more specific for D-P relationships. These patterns are able to recognize phenotype candidates that are not discovered by a NER and are in relation with a RD. Then, SPARE* (SPARE_AND_SVM_Unknown) can classify the RD phenotype candidates extracted by SPARE patterns.

Next, we introduced a compositional semantics model in order to validate and evaluate the novelty of the extracted candidates. Then, we implemented mapping rules to map the phenotype candidates to their closest HPO phenotypes with the most favorable mapping category. This shows the ability of our proposed method to discover existing and potentially new RD phenotypes.

Finally, we applied SPARE* to propose enriching Orphanet and Orphadata. SPARE* extracts RD phenotypes from related biomedical articles, which are compared to those listed in Orphanet and Orphadata. We proposed a semantic space model and mapping rules for identifying which phenotypes are known and which phenotypes are potentially new. New phenotypes are proposed to refine the content of Orphanet summary and Orphadata.

6

Using Text Mining and Pattern Structures for Disease Classification and Ontology Enrichment

Contents

6.1 Introduction

This chapter presents an original use of text mining and pattern structures to provide a new Rare Disease (RD) classification based on the phenotypic descriptions of RD, which we propose to use for enriching an existing RD ontology by suggesting new RD classes. In addition to the existing databases and ontologies about RDs and their phenotypes, text mining is used to complete the data about phenotypic description of RDs. Next, pattern structures

classify objects, here RDs, based on their descriptions, *i.e.*, sets of phenotypes in a structure named a concept lattice. That we redefine the meet operator of pattern structure to take into consideration both a RD ontology, *i.e.*, an existing classification of objects, and a phenotype ontology, *i.e.*, a classification of elements of object descriptions. The resulting concept lattice is a classification of objects, by grouping RDs that share common or similar phenotypes according to the ontology. Indeed, this classification groups new sets of RDs that were not associated in the original RD ontology.

Diseases are described by their phenotypes. Knowing disease phenotypes is helpful for medical diagnosis and for therapeutical decisions. Moreover, classifying diseases or finding disease similarity in terms of their phenotypic descriptions, plays an important role in their diagnostics. Therefore, the main goals of this work are to classify RDs on the basis of their common phenotypes and to provide new RDs classes that could be useful in RD diagnosis. This task is not straightforward. We list here three main challenges. As the quality of the results for any data mining algorithm relies on the quality and the adequacy of the input data. The first challenge is to provide a complete set of RD phenotypes. The second challenge is to define the classification of RDs using background knowledge introduced by a phenotype ontology. The classification of RDs should take into account two dimensions: the families of RD existing ontologies and families of phenotypes. The third challenge is to find a small set of interesting RD classes among a large set of classes (*i.e.,* concepts proposed by lattice) to be useful for experts.

A naïve classification can be done by grouping a set of RDs in a class if they are associated and this class is described by their common phenotypes. Then, these classes can be organized in taxonomy (hierarchy), where a class c_1 is subsumed by a class c_2 if the set of RDs in c_1 are included in c_2. A simple approach for classifying objects such as RDs is to use Formal Concept Analysis (FCA) [OPG13] with RDs as objects and phenotypes as attributes. In this case, FCA produces a lattice of formal concepts partially ordered where the extent of each concept is a set of RDs and the intent is a set of phenotypes shared by these RDs. FCA groups RDs using only the intersection between sets of phenotypes shared by them. FCA works with binary contexts, and taking into account attributes that belong to ontology is not straightforward.

Phenotype ontologies (*e.g.,* the phenotype ontology provided by Orphadata) propose hierarchical relations between phenotypes that can be used to compare phenotypes. For instance, the disease "Mitochondrial myopathy" has the phenotype "Transient amaurosisl" and the disease "Fukuhara syndrome" has the phenotype "Optic nerve anomaly". Using FCA, as there is no phenotype common to these two diseases, hence there is no similarity between them. The phenotype ontology of Orphadata relates these two phenotypes ("Transient amaurosisl" and "Optic nerve anomaly") by having a common parent that is "stature". Thus, the two diseases have a common phenotype "stature".

We propose the use of pattern structures, an extension of FCA, for considering the disease similarity and then offering a disease classification that considers the existing hierarchy of a disease ontology and their phenotypic descriptions with respect to a phenotype ontology. In particular, we define a meet operator that enables considering domain knowledge previously defined within a disease ontology and a phenotype ontology.

Pattern mining methods (*e.g.,* FCA, Pattern structures) generally produce a huge number of concepts. It is then difficult to explore manually all the concepts to identify those of interest. In this work, we propose an approach for selecting a small set of more interesting patterns that are easy to be interpreted by an expert.

The main two applications of this work are: *(1)* providing a new RD classification based on their phenotypic descriptions and *(2)* enriching an existing RD ontology by suggesting new and interesting (with the hypothesis that they are useful in RD diagnosis) RD classes.

The chapter is structured as follows: Section 6.2 introduces the operations on ontologies we used in the method section. Section 6.3 presents the materials. Section 6.4 introduces our method. Section 6.5 presents our experiments on sets of RDs and their results. Section 6.6 presents the related works. Finally, Section 6.7 concludes on this work.

6.2 Ontologies and Operations on Ontologies

In this chapter, we define our ontologies as being a formal knowledge representation constituted of a set of domain concepts and relations between them [Gru93]. To avoid any confusion, we use the term *ontology class* for a concept lying in an ontology and *pattern concept* for concepts in FCA and pattern structures. Ontology classes are ordered by a subsumption relation, denoted $\leqslant$. Given two classes c_1 and c_2, $c_2 \leqslant c_1$ means that c_2 is subsumed by c_1, *i.e.,* all elements in c_2 are included in c_1. We use in this work two particular operators on ontologies: the Least Common Ancestor (LCA) and the Most Specific (MS) concept of a set of concepts. In the general case, two classes may admit several LCAs. For example, LCAs of c_6 and c_7 in the ontology shown in Figure 6.1 are c_3 and c_4. In the case where class c_i is subsumed by c_j, then c_j is the LCA of c_i and c_j. For example, the LCA of c_4 and c_7 is c_4, because c_4 subsumes c_7.

Definition 22 *(LCA of two classes)*
The LCAs of two ontology classes c_i and c_j are the most specific classes of the ontology that subsume both c_i and c_j.

Definition 23 *(LCA of two sets of classes)*
The LCA for a pair of classes can be generalized to the LCA of a pair of sets of classes as defined by Alam *et al.* [ABNS15]. In this case the LCA of each pair of elements from the two sets is computed. Then, only the most specific LCAs are kept as the LCAs of the two sets.

In this chapter, we will consider two kinds of simplified ontologies: DAG (Direct Acyclic Graph) and Tree ontologies. A consequence is that LCA will be unique for Tree but may be multiple for DAG.

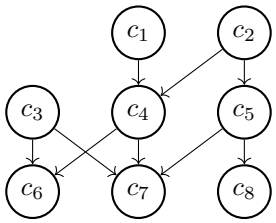


Figure 6.1: Example of ontology, with a DAG structure.

6.3 Materials

We used four resources in this chapter: the Orphanet RD classifications, the Orphadata phenotype classification, the Orphadata RD-Phenotype relationships and PubMed abstracts that are related to RDs.

6.3.1 Orphanet RD Classifications

Orphanet RD Classifications classify 8,644 RDs from Orphadata into several and distinct *classification groups*. Each classification group consists of a set of RDs ordered in an ontology, denoted O^{rd}. It orders RDs with the subsumption relation $\leqslant$, where $rd_i \leqslant rd_j$ means that rd_i is subsumed by rd_j. Each RD classification is structured as DAG, allowing one RD being subsumed by several parents. This means that a rd_i in the ontology is a class of RDs, which may contain only one RD (the leaves of the DAG) or several RDs. Currently, Orphanet contains 33 RD classifications. We focused on the group of rare cardiac diseases. It contains 207 RDs organized over 7 *classification levels* (*i.e.,* maximum depth of the classification is 7 and the average number of RDs in each level is

about 29). The Orphanet data (in general) are publicly available at `http://www.orphadata.org/cgi-bin/inc/product3.inc.php`.

6.3.2 The Orphadata Phenotype Classification

Orphadata Phenotype Classification contains phenotypes, named clinical signs. We used the version 1.1.4/4.1.3 updated 2016-03. It contains 1,273 phenotypes. These phenotypes are organized in an ontology structure $O^p$with a subsumption relation $\leqslant$, where $p_i \leqslant p_j$ means that the phenotype p_i is subsumed by p_j. O^p structure is a tree where, consequently, each phenotype is subsumed by at most one parent.

6.3.3 RD-Phenotype Relationships

Orphadata lists a large set of *RD-phenotype* relationships. Version 1.1.4/4.1.3 of the Orphadata contains 52,503 of them. Only 2,689 from all 8,644 RDs of Orphadata are associated with phenotypes. In the rest of this chapter, we refer to the association of RD with phenotypes as the *D-P-List*.

6.3.4 Corpus of PubMed Abstracts

To enrich existing RD-knowledge, we analyze publications in the medical domain. Indeed, we build a corpus of 148,596 PubMed abstracts that are related to RDs of cardiac rare disease classification. The set of their PMIDs is available at https://sourceforge.net/projects/spare2015/files/PubMedAbstractIDs. In the previous chapters, we applied new phenotype extraction methods to extract phenotypes related to RDs.

6.4 Methods

We propose an original method based on pattern structures using ontologies for classifying RDs based on their phenotypes and for enriching an existing RD ontology. Our approach is depicted in Figure 6.2. It consists of three main steps. The first step prepares the required data for pattern structures. The second step uses pattern structures to build a concept lattice that offers a new classification of RD. The third step finds interesting elements (*i.e.,* pattern concepts) of the resulting lattice. These elements are potential enrichments to the initial RD ontology. Next subsections detail these three steps.

6.4.1 Text Mining for Data Completion

In this step, we use text mining for completing the data needed to feed our pattern structure context. For all RDs of rare cardiac disease classification, we extract their sets of phenotypes from Orphadata. Some of these RDs have no phenotype listed in Orphadata. To solve this lack, we use a text mining approach (here we use our SPARE* method described in Chapters 4 and 5), to extract phenotypes from related PubMed abstracts. For instance, cardiac rare disease classification contains 207 RDs. Their phenotype sets are required to build the context of our pattern structure approach. Only 114 RDs (over 207) have phenotypes in Orphadata. Thus SPARE* method extracted phenotypes from PubMed abstracts for those that do not have any phenotype in Orphadata. In order to do that, we first queried PubMed for retrieving abstracts related to these RDs. SPARE* is then run over these abstracts to extract RD phenotypes from.

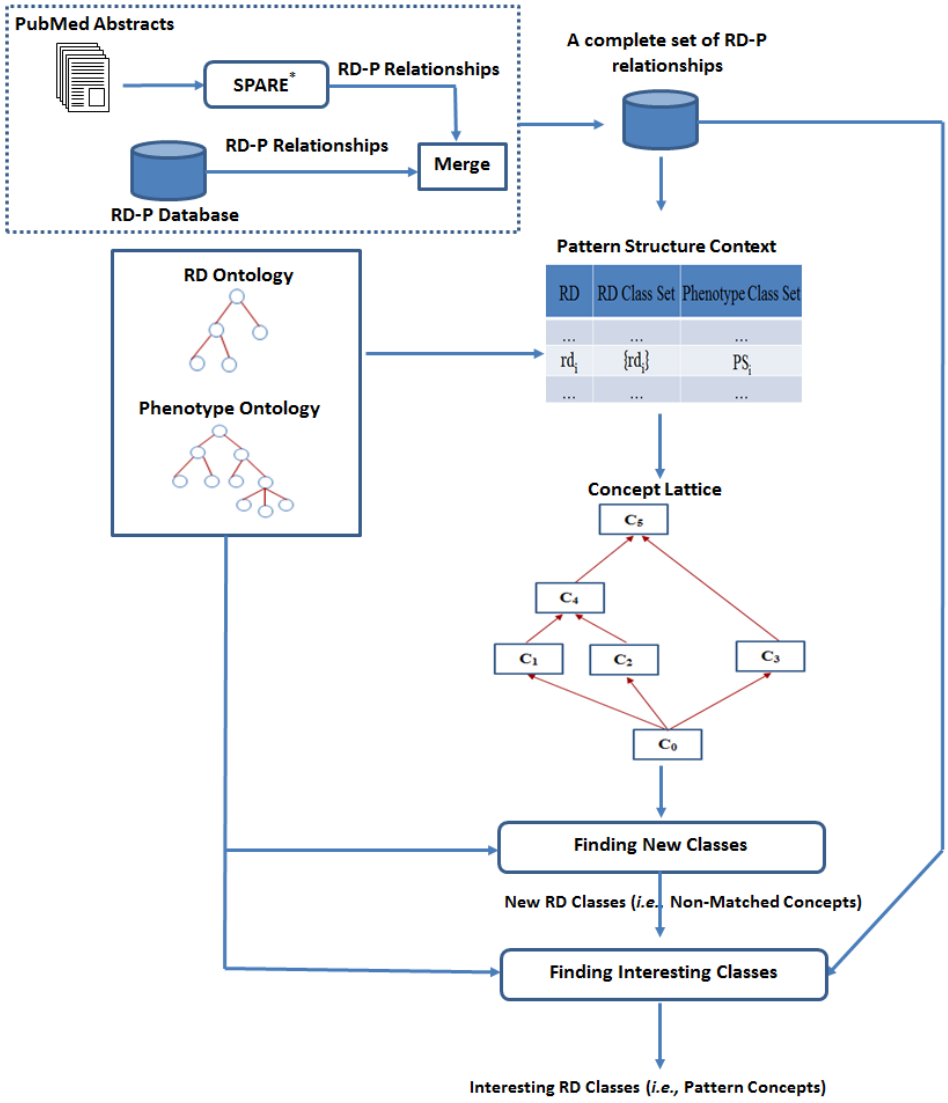


Figure 6.2: Overview of our method

6.4.2 Pattern Structure for Disease Classification

Given two sets of RDs and phenotypes that are defined in ontologies O^{rd} and O^{p} respectively, we define our pattern structures as a triple $(G, (D, \sqcap), \delta)$, where

- $G = \{rd_1, rd_2, \ldots, rd_n\}$ is a set of RDs.
- $D = \{\langle DS_1, PS_1\rangle, \langle DS_2, PS_2\rangle, \ldots, \langle DS_m, PS_m\rangle\}$ is the domain of descriptions, where descriptions are vectors of two elements denoted $\langle DS, PS\rangle$ and detailed in the following.
- $\delta : G \rightarrow D$ maps an object to a description, *i.e.* $\delta(rd_i) = \langle DS_i, PS_i\rangle$.
- $(D, \sqcap)$ is a meet-semilattice on D *w.r.t.* $\sqcap$, which is the meet operator defined between elements of D.

A rare disease $rd_i \in G$ is described by $\delta(rd_i) = \langle DS_i, PS_i\rangle$, which is a composite structure of two elements: *(1)* Disease Set (DS) and *(2)* Phenotype Set (PS).

DS is a set of RDs from O^{rd}. For one RD object, its DS contains the RD class itself. For two RD objects or more, DS contains the RD classes that are their LCAs in O^{rd}. We added DS to our context in order to enable retrieving the corresponding concepts that are matched with all RD classes of the original ontology. This is useful to keep the original ontology structure in the resulting lattice, *i.e.,* each RD class in the original ontology has an exact match with a concept in the lattice. This DS dimension is similar to some nominal scalings in FCA but it is considered as a pattern. To illustrate, if we use only DS as the description (*e.g.,* $\delta(rd_i) = \langle rd_i \rangle$), then the resulting lattice is exactly the same as what would produce on FCA-based classification.

For a given RD, the phenotype set of description is defined as follows: $PS_i = \{p \in M \mid \exists rd \in G, (rd, p) \in$ ***D-P-List***$, rd \geqslant rd_i\}$, is a set of phenotypes from M, where M is a set of all phenotypes from O^p, and these phenotypes have a relation defined in ***D-P-List*** with rd_i or any of its ancestors.

The meet operation between two RDs rd_i and rd_j, whose descriptions are $\delta(rd_i) = \langle rd_i, PS_i \rangle$ and $\delta(rd_j) = \langle rd_j, PS_j \rangle$ respectively, is defined as:

$$\delta(rd_i) \sqcap \delta(rd_j) = \langle rd_i, PS_i \rangle \sqcap \langle rd_j, PS_j \rangle \; \delta(rd_i) \sqcap \delta(rd_j) = \langle rd_i \sqcap rd_j, PS_i \sqcap PS_j \rangle$$

$rd_i \sqcap rd_j$ gives the similarity between two RDs rd_i and rd_j with respect to a RD ontology O^{rd}. While $PS_i \sqcap PS_j$ gives the similarity between two phenotype sets PS_i and PS_j with respect to a phenotype ontology O^p. The LCA is the similarity operator applied to each DS / PS description.

LCA-Based Similarity

Similarly to Alam *et al.* [ABNS15], LCA is the similarity operator between two descriptions that are formatted in terms of ontology classes. The LCA between two disease classes may be a set of classes as the disease ontology O^{rd} has a DAG structure. On the opposite, the LCA between two phenotype classes is unique as the phenotype ontology O^p has a tree structure. As the DAG of O^{rd} is rooted DAG, this ensures that there is a LCA between any two disease classes.

The Similarity between Two RD Classes in O^{rd}: We use a simple brute-force algorithm, proposed previously by Aït-Kaci et al. [AKBLN89] for computing LCA of two classes in a DAG, to find the LCA between two disease classes in O^{rd}. We first compute all their common ancestors, and then we keep the most specific ones as their LCA.

The Similarity between Two Phenotype Classes in O^p: As O^p has tree structure, we optimized LCA computation following the proposed technique by Bender *et al.* [BFCP$^+$05] that reduces the LCA problem to Range Minimum Query (RMQ) problem. RMQ allows answering a LCA query in a constant time.

Similarly to Alam *et al.* [ABNS15] we generalized LCA to be computed between two sets of ontology classes (*e.g.,* two sets of RDs, two sets of phenotypes). Given two sets $X_N = \{x_1, x_2, \ldots, x_n\}$ and $Y_M = \{y_1, y_2, \ldots, y_m\}$ of ontology classes, the similarity between X_N and Y_M is defined as the following:

$$X_N \sqcap Y_M = MS(\bigcup LCA(x_i, y_j)), \forall i \in N, \forall j \in M \tag{6.1}$$

where MS is a function that takes as an input LCAs of each pair from X_N and Y_M. Then, it discards any general classes and keeps only the most specific ones as the final output of the similarity operation.

Lattice Generation

We use FCAPS[14] software that is developed in C++ for dealing with pattern structures. FCAPS contains an adapted implementation of AddIntent algorthim [vdMOK04] for building a lattice of pattern structures. The resulting concept lattice, from our pattern structure context, consists in a set of pattern concepts, each associating a set of RDs (the extent) with the description of this set of RDs (the intent). We represent our pattern concept as follow: $c_i = \{Ext(c_i), < Int_{rd}(c_i), Int_p(c_i) >\}$, where c_i is a pattern concept, $Ext(c_i)$ is its extent (set of RDs), $< Int_{rd}(c_i), Int_p(c_i) >$ is its intent that is a composite of $Int_{rd}(c_i)$ and $Int_p(c_i)$, which are respectively the common RDs and the common phenotypes for RDs in the pattern extent. For instance, the following "{{"MELAS", "MERRF", "Glycogen_storage_disease_due_to_glycogen_debranching_enzyme_deficiency"}, <{"Rare_familial_disorder_with_hypertrophic_cardiomyopathy"}, {"Intellectual deficit", "Myopathy", "Short stature"}>}" is an example of a pattern concept. The extent of this pattern contains 3 RDs "MELAS", "MERRF" and "Glycogen_storage_disease_due_to_glycogen_debranching_enzyme_deficiency". The intent of this pattern contains their common RD, which is "Rare familial disorder with hypertrophic cardiomyopathy", and also their common phenotypes, which are "Intellectual deficit", "Myopathy" and "Short stature".

Finding New RD Classes

To find new RD classes, we propose to compare every RD class from the RD ontology to every concept in the lattice. This comparison process produces two sets of concepts: *(1)* a set of concepts that match with RD classes, *(2)* a set of concepts that do not match any RD class. By construction, thanks to the DS description, each RD class in the ontology O^{rd} has an exact match with one pattern concept from the resulting concept lattice. A RD rd_i is matched with a concept c_i if $Ext(c_i) = \{rd_i\}$, *i.e.,* the set of RDs in the extent of c_i is the same RD set of rd_i. This process produces pairs of matched RD classes and concepts. The concepts that are not included in these pairs are considered as non-matched concepts. A non-matched concept is a potential new RD class that we could suggest to add in the ontology O^{rd}.

An Illustrative Example

This subsection presents a toy example to illustrate our lattice generation step and finding new elements from the resulting lattice. Figure 6.3 presents this toy example. Initially, we started with a RD ontology, a phenotype ontology and a RD phenotype database (these phenotypes would be from Orphadata and from PubMed) to format our pattern structure context. Each row in the pattern structure context contains the definition of a RD object. For instance, the second row of the context represented in Figure 6.3 shows that rd_1 is defined by $\langle\{rd_1\}, \{p_3, p_5\}\rangle$, where $\{rd_1\}$ is its corresponding RD class in O^{rd} and $\{p_3, p_5\}$ is the phenotypes of rd_1 listed in a RD phenotype database.

It should be noted that a RD object inherits all phenotypes of their ancestors. For example, the phenotypes of both rd_2 and rd_3 are $\{p_4\}$ and $\{p_5, p_6\}$ respectively. rd_1 is the ancestor of rd_2 and rd_3. When formatting the pattern structure context, the phenotypes of rd_1 are added to the phenotype sets of both rd_2 and rd_3 to be $\{p_4, p_3, p_5\}$ and $\{p_5, p_6, p_3\}$ respectively. In Orphadata this is not always the case, thus this is the result of a preprocessing phase to ensure coherence in the data. Then, a concept lattice is generated according to this context using the meet operator described in subsection 6.4.2. This concept lattice shows a new RD classification which is different from the original one, RD ontology.

Finally, the resulting lattice or classification is compared to RD ontology to propose new RD classes. For instance, this new classification suggests a new RD class, which is $(\{rd_3, rd_4\}, \langle\{rd_0\}, \{p_6\}\rangle)$.

[14] https://github.com/AlekseyBuzmakov/FCAPS

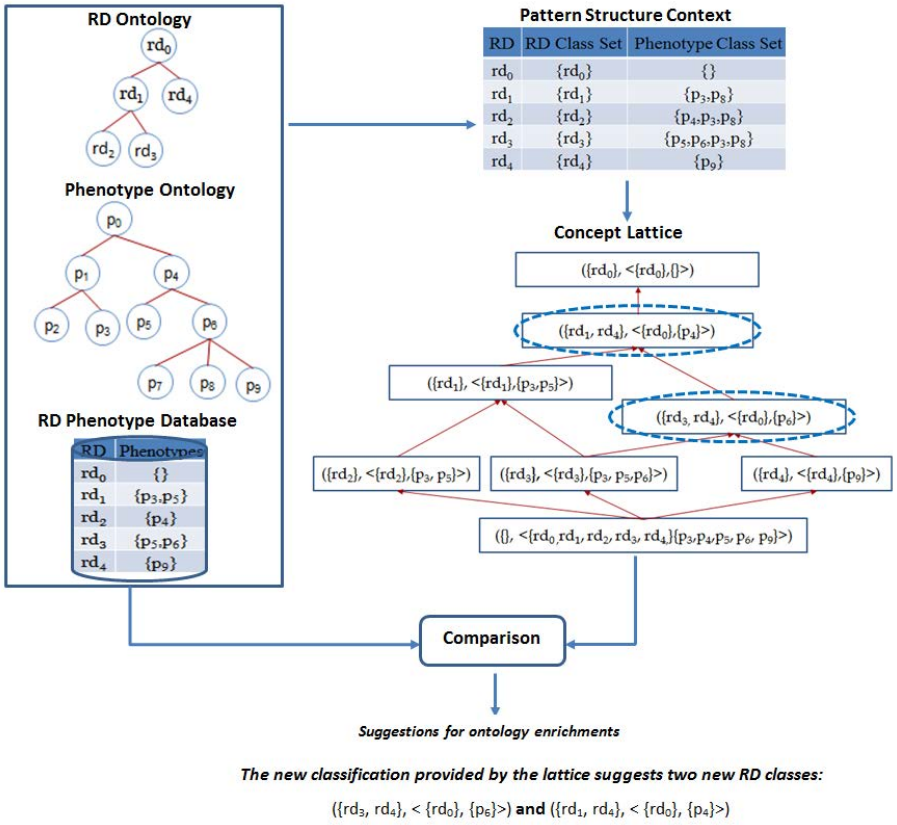


Figure 6.3: A Toy Example for the illustration of our method

6.4.3 Finding Interesting Concepts

The number of pattern concepts in a lattice may be very large, which complicates their manual analysis by an expert. Therefore, we need a way to select a smaller set of pattern concepts in agreement with our objective. We propose here several measures that to select subsets of patterns potentially more interesting than others. The definition of an interesting pattern is vague, and should be adapted to the purpose of the application and its goals. In this chapter, we propose two different measures for finding interesting pattern concepts from a large set of concepts. In our case, the objective is to suggest to experts concepts for further analysis that could help in medical diagnosis and disease treatments. First, we introduce an objective measure named the $p\text{-}value$, which measures the statistical significance of the association between two diseases. Second, we introduce another measure, which benefits, both from the dissimilarity between concepts computed on the basis of the sets of phenotypes in their intents (here we *Gap*) and from the lattice structure, for the concept selection.

Statistical Selection

In statistics, the null hypothesis states that there is no relation between two observations or two elements. To test this null hypothesis, the $p\text{-}value$ may be used. A small $p\text{-}value$ (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so we can reject the null hypothesis. A large $p\text{-}value$ (> 0.05) indicates weak evidence against the null hypothesis, so we can accept the null hypothesis. $p\text{-}value$ has been used in biomedicine for the validation and the selection of data mining results [ADR17,FG13,RBMM05]. For example, Ramani *et al.* in [RBMM05] used the hypergeometric distribution to compute $p\text{-}value$ in order to evaluate how strong the association between two

Example ID	rd_i	rd_j	n_i	n_j	k	p-$value$
1	MELAS	COXPD10	209	68	68	3.407068977679168E-38
2	MELAS	GSD type 0b	209	68	68	3.407068977679168E-38
3	MELAS	CPVT	209	7	7	3.175957484831089E-4
4	MELAS	JET	209	7	5	0.037011822591468985
5	MELAS	Multifocal atrial tachycardia	209	0	0	1

Table 6.1: Examples of p-$values$ of associating two RDs.

drugs is, which is helpful for identifying drug-drug relationships. In this work, we similarly use the hypergeometric distribution to compute the p-$value$, but here to evaluate if there is a strong the association between two RDs based on their shared phenotypes.

Definition 24 *(p-$value$)*
The p-$value$ is defined as:

$$p - value = \frac{\sum_{m=k}^{min(n_i,n_j))} \binom{n_i}{m}\binom{N-n_i}{n_j-m}}{\binom{n_i}{m}} \tag{6.2}$$

where N is the total number of phenotypes, n_i, n_j are the number of phenotypes of RDs rd_i and rd_j respectively, and k is the number of shared phenotypes by rd_i and rd_j.

Table 6.1 presents 5 examples of sets of two RDs (rd_i and rd_j). For each pair of RDs, the table presents their names, the numbers of phenotypes they have (n_i and n_j), the number of their shared phenotypes (k) and their p-$value$ calculated by Equation 6.2. For example, "MELAS (Mitochondrial myopathy, encephalopathy, lactic acidosis, and stroke)" RD has 209 phenotypes and "COXPD10" has 68 phenotypes. They have in common 68 phenotypes, so their p-$value$ is $3.407068977679168E$-38. From the table, It should be noted that the first two examples have the same p-$value$ as they have the same numbers of phenotypes and the same number of common phenotypes. The third example has lower p-$value$ than the fourth example as the RDs of the third example share more phenotypes ($k = 7$) than the RDs of the fourth example ($k = 5$). This means that RDs of the third example have a stronger association between them than RDs of the fourth example. Consequently, the medical diagnosis for RDs of the third example could be more similar than the others in the fourth example. In the last example, "Multifocal atrial tachycardia" disease does not have any phenotype. So, the p-$value$ of associating this RD with "MELAS" disease is 1, which means that there is no association between them.

For each concept in the lattice, we compute its average p-$value$, denoted avg-p. A concept c_1 is of interest if its avg-$p(c_1)$ is lower than a threshold denoted min-avg-p. Given two concepts c_1 and c_2 we consider c_1 is more interesting than c_2 if avg-$p(c_1) < avg$-$p(c_2)$.

Definition 25 *(avg-p of a concept)*
avg-p of a concept is the average value of p-$values$ quantifying the association between every RD pairs of the concept extent.

Definition 26 *(A Concept support)*
The support of a concept c_i is the number of objects in its extent, denoted $sup(c_i) = |Ext(c_i)|$.

Algorithm 2 presents our method, named $Statistical$ method, for selecting the interesting concepts based on their avg-p values. We check every concept in the resulting lattice if it is interesting or not. A concept c_i is of interest if it has the following: *(1)* $sup(c_i) \geq 2$; (2) avg-$p(c_i) \leq min$-avg-p and (3) avg-$p(c_i)$ is lower than all avg-p values of its superconcepts. Condition *(1)* ensures that the selected concept should include at least two RDs

in its extent because we are interested in finding the concepts associating two RDs or more. Condition *(2)* ensures that the concept contains RDs that have, in average, significant associations between them. Condition *(3)* ensures that the selected concept does not have a more interesting superconcept.

Algorithm 2 select concepts of interest based on a statistical significance measure, $p\text{-}value$

```
1: procedure Statistical(conceptList, min-avg-p)
2:     selectedConceptList = {}
3:     for each concept c_i in conceptList do
4:         if (sup(c_i) ≥ 2 and avg-p(c_i) ≤ min-avg-p) then
5:             isSelected = True
6:             superConceptList = getSuperConceptList(c_i)
7:             for each superConcept c_j in superConceptList do
8:                 if (avg-p(c_j) ≤ avg-p(c_i)) then
9:                     isSelected=false
10:                    break
11:                end if
12:            end for
13:            if (isSelected) then
14:                selectedlList.add(c_i)
15:            end if
16:        end if
17:    end for
18:    return selectedConceptList
19: end procedure
```

Gap-Based Selection

Similarly to Keller *et al.* [KK12b], the $Gap\text{-}based$ selection method looks for RD classes that are most-informative (*i.e.,* in terms of maximal information about phenotype descriptions) in the sense that if we add another RD, the set of shared phenotypes is relatively smaller. To illustrate this measure, suppose we have 3 RDs: rd_1, rd_2 and rd_3. rd_1 and rd_2 share a large proportion of their associated phenotypes. rd_3 shares relatively few phenotypes with each of rd_1 and rd_2. In this scenario, the gap between phenotype sets drops significantly when superconcepts involving rd_1 and rd_2 are formed by adding rd_3. To identify which RD classes to consider, we need to find the concepts that have superconcepts with a very big change in the intent. This can be done by traversing the concept lattice from bottom to top, and visiting superconcepts looking for significant drops in the intent (set of phenotypes).

The gap between two concepts is quantified by computing the asymmetric dissimilarity between them based on their phenotype sets of their intents. Equation 6.3 shows the formula for this computation. As shown in the equation, we use asymmetric dissimilarity because the dissimilarity(c_i, c_j) is different form the dissimilarity(c_j, c_i). Therefore, we compute both of them and then take their average. Algorithm 3 presents the procedure of computing the dissimilarity between two concepts c_i and c_j in sequence. To do this, for one phenotype in $Int_p(c_i)$ we calculate the dissimilarities between it and each phenotype in $Int_p(c_j)$. Then, we keep the smallest dissimilarity, *i.e.,* the dissimilarity with the closest phenotype in $Int_p(c_j)$. We repeat this process for all phenotypes in $Int_p(c_i)$ and finally we get the average of these smallest dissimilarities. As shown in the algorithm, the dissimilarity uses $path\text{-}length(p_i, p_j)$ function, which is the length of the path between the two phenotypes p_i and p_j in the ontology O^p.

$$\text{asym\_dissimilarity}(c_i, c_j) = \frac{(\text{dissimilarity}(c_i, c_j) + \text{dissimilarity}(c_j, c_i))}{2} \tag{6.3}$$

In the $Gap\text{-}based$ method, we consider a concept is an informative concept and is of interest if the gap (here the dissimilarity) with each of its superconcepts is higher than a specific threshold, denoted $min\text{-}gap$. If the gap between the concept and one of its superconcept is lower than $min\text{-}gap$, we assume that its superconcept is more informative as it has similar phenotypic description and larger extent (because it is a superconcept). Therefore, we

Algorithm 3 compute the dissimilarity between two concepts

procedure COMPUTE_DISSIMILARITY(c_i, c_j)
  dissimilarity=0.0
  for each phenotype p_i in $Int_p(c_i)$ **do**
    $dis\_p_i = min_{p_j \in Int_p(c_j)}\{1 - \frac{1}{path\_length(p_i+p_j)+1}\}$
    dissimilarity = dissimilarity + dis_p_i
  end for
  dissimilarity = $\frac{dissimilarity}{|Int_p(c_i)|}$
  $return$ dissimilarity
end procedure

decide to discard the concept and keep the superconcept with the possible largest extent. Algorithm 4 describes the $Gap\text{-}based$ method for selecting the informative concepts from the lattice based on their gaps with other concepts in the lattice. Algorithm 5 tests if a concept should be preserved in the final selected list or not. $EXTEND\_GAP$ procedure (shown in Algorithm 6) used in Algorithm 4 handles the case of having a chain of concepts (*e.g.,* a concept and their ancestors) where the gap between each two consecutive concepts is very low. Therefore, If a concept is discarded, we adapt the threshold of its superconcepts and we attach for each superconcept the maximum gap value with all of its subconcepts.

Algorithm 4 select concepts from lattice based on a gap between the concepts

procedure $Gap\text{-}based(conceptList, min\text{-}gap)$
  selectedConceptList = {}
  for each concept c_i in $conceptList$ **do**
    superConceptList = getSuperConceptList(c_i)
    if (IS_PRESERVED_CONCEPT(c_i, superConceptList, $min\text{-}gap$)) **then**
      selectedlList.add(c_i)
    else
      EXTEND_GAP(c_i, superConceptList)
    end if
  end for
  $return$ selectedConceptList
end procedure

Algorithm 5 check if a concept should be preserved or not

procedure IS_PRESERVED_CONCEPT(c_i, superConceptList, $min\text{-}gap$)
  for each superConcept c_j in superConceptList **do**
    gap = asym_dissimilarity(c_i, c_j)
    if (gap==0) **then**
      $return\ false$
    end if
    extendedGap = gap + concept.$getExtendedGap()$
    if (extendedGap < $min\text{-}gap$) **then**
      $return\ False$
    end if
  end for
  $return\ true$
end procedure

6.5 Experiments and Results

6.5.1 Data Preparation

The context for the pattern structure contains 207 RDs. Only 114 RDs (or their ancestors) have at least one phenotype from Orphadata. SPARE* extracts phenotypes of 65 RDs (from 93 RDs that have not any phenotype in

Algorithm 6 extend the gap value

```
1: procedure EXTEND_GAP(c_i, superConceptList)
2:     for each superConcept c_j in superConceptList do
3:         gap = asym_dissimilarity(c_i, c_j)
4:         if (c_j.getExtendedGap() < gap) then
5:             c_j.setExtendedGap(gap)
6:         end if
7:     end for
8: end procedure
```

Orphadata), while it fails at extracting phenotypes for the other 28 RDs because they do not have UMLS CUI. In total, 179 (114+65) RDs have phenotypes either extracted from Orphadata or from PubMed abstracts.

6.5.2 Construction of RD Lattice

To formulate our pattern structure context, we use the 207 RDs as the objects. The descriptions of these objects are their classes in the RD classification and their phenotypes (or the phenotypes of their ancestors) extracted from both Orphadata and PubMed abstracts (by using SPARE*). This context contains 207 RDs, 600 phenotypes and 8,789 RD-Phenotype relationships. Only 179 RDs have non-empty phenotype sets and the other 28 RDs have empty sets of phenotypes. Then, we used FCAPS software to build the concept lattice from this pattern structure context. The resulting concept lattice contains 4,829 concepts and 17,367 subsumption relations between these concepts. The maximum depth of this lattice is 26 levels and the average depth is 15. The average number of concepts in each level is about 185.

When comparing the "Rare cardiac diseases" classification given by the Orphanet to the concepts of the generated lattice. All 207 RDs of the original classification are matched exactly with corresponding 207 concepts. This means that the number of the other non-matched concepts is 4,622. These non-matched concepts provide new RD classes that may suggest new enrichments to the original classification.

6.5.3 Selection of Interesting Concepts

Statistical Selection

The whole lattice contains 4,829 concepts. Concepts with unique RD in their extent correspond to already existing concept in Orphanet classification. Thanks to SPARE*, 65 of these concepts are described in a more precise way, *i.e.,* with a more detailed list of phenotypes. For instance, the RD "Restrictive cardiomyopathy" has new phenotypes: "Respiratory Failure", "Vasculitis" and "Congenital heart disorder". However, the main improvement from this lattice comes from concepts where extent groups several RDs. Only 4,662 concepts (from 4,829 concepts) contain two RDs or more in their extents, *i.e.,* have $support \geq 2$. The list of these concepts is sorted by their $avg\text{-}p$ values and is accessible online at `https://sourceforge.net/projects/spare2015/files/Concepts4662supGT2`. An expert can examine them in sequence starting from the most interesting concepts with the lowest $avg\text{-}p$ values. Table 6.2 shows the first top 10 concepts from this list. It presents their support (number of RDs in their extents), the number of shared phenotypes (the list of their names are available in the online file), their $avg\text{-}p$ values and stability values. Figure 6.4 shows the number of concepts that have $avg\text{-}p$ lower than or equal to a given $min\text{-}avg\text{-}p$ threshold. This figure shows that the number of concepts increases when the $min\text{-}avg\text{-}p$ threshold is changed from 0.0 to 0.05 and increases slightly when the threshold is changed from 0.05 to 0.9, only 26 concepts are the difference. We can choose any value between 0.05 and 0.9 as a threshold. Therefore, we chose the lowest value, $min\text{-}avg\text{-}p = 0.05$, as our threshold value in order to discard the lowest interesting concepts. In addition, the uses of $p\_value = 0.05$ as a threshold is very common in the literature. The number of the concepts that have $avg\text{-}p \leq 0.05$ is 3,870.

Concept Support	#Shared Phenotypes	avg-p	Stability
8	71	3.276E-131	0.96875
3	68	9.96E-127	0.625
5	81	3.19E-113	0.8125
2	82	3.19E-112	0.25
3	58	7.57E-109	0.625
4	55	1.73E-106	0.75
4	49	4.23E-99	0.75
3	41	4.03E-92	0.75
3	38	1.52E-87	0.75
2	38	1.15E-85	0.25

Table 6.2: The first top 10 concepts sorted by their $avg\text{-}p$ values.

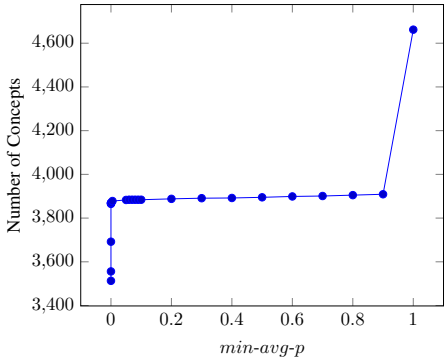


Figure 6.4: The number of selected concepts according to different $min\text{-}avg\text{-}p$ thresholds.

Figure 6.5 shows the number of selected concepts by the $Statistical$ method (described in Algorithm 2) when using different $min\text{-}avg\text{-}p$ thresholds. At the selected threshold $min\text{-}avg\text{-}p = 0.05$, it selects 934 concepts as the most interesting concepts among 3,870 concepts that have $avg\text{-}p \leq 0.05$. This selection algorithm benefits from the statistical significance of associating RDs in the concept extent and from the lattice structure (subsumption relations between concepts). The 934 selected concepts are sorted ascending by their $avg\text{-}p$ and they are accessible at `https://sourceforge.net/projects/spare2015/files/Concepts934-p_valueSortedAsc`.

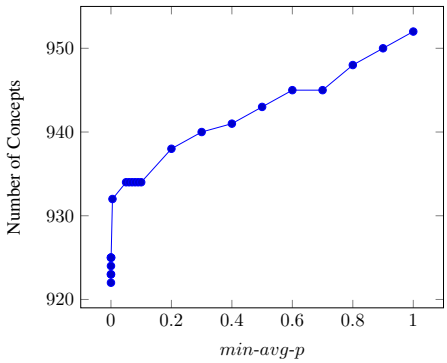


Figure 6.5: The number of selected concepts by $Statistical$ algorithm (Algorithm 2) with different $min\text{-}avg\text{-}p$ thresholds.

Figure 6.6 shows the whole lattice including the 4,829 concepts. It shows also the 934 concepts (blue and black nodes) selected by the $Statistical$ algorithm (Algorithm 2) and how they are located in the complete lattice. The

black nodes are the concepts that match exactly the RD classes of the original classification. The blue nodes are the selected concepts. The red nodes are the filtered concepts. We observed that even if there is a dense region for the blue nodes (selected concepts), there is no regularity in being closer to the top or to the bottom of the lattice. The selected concepts form a sublattice of 934 concepts from the complete lattice. The maximum depth of this sublattice is 11 levels and the average depth is 3. The average number of concepts in each level is about 85.

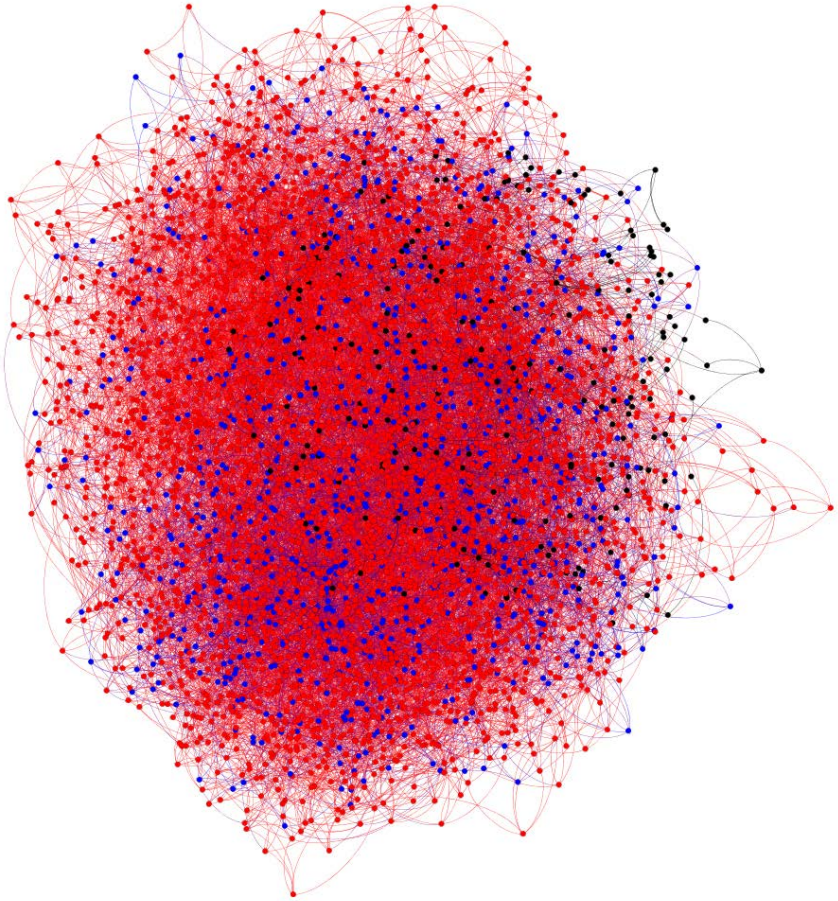
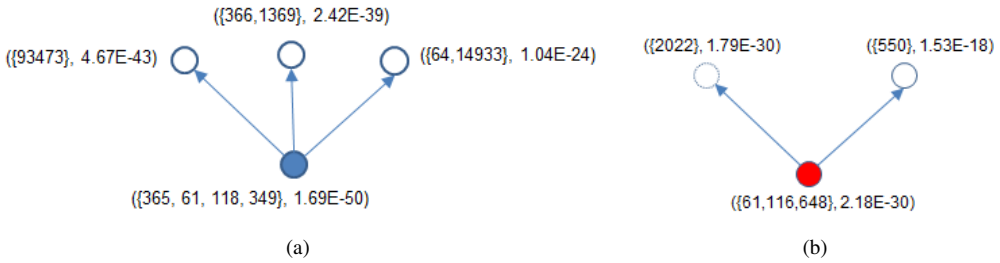


Figure 6.6: Overview of the whole lattice and the selected sublattice. Black nodes are the concepts that match exactly the RD classes of the original classification. Blue nodes are the selected concepts, while red nodes are the filtered concepts.

Figure 6.7(a) shows an example of selected concept. The blue node is the selected concept and the others are its superconcepts. Its extent contains RDs with the following orphan numbers: 365, 308552, 420429, 61, 118 and 349. The avg-p value of this concept is 1.04E-2. We select this concept because its avg-p value is smaller than the avg-p values of its superconcepts, which are 4.67E-43, 2.42E-39 and 1.04E-24 respectively.

Figure 6.7(b) shows an example of a discarded concept. The red node is the discarded concept and the others are its superconcepts. Its extent contains RDs with the following orphan numbers: 61, 116 and 648. We discard this concept because its avg-p, 2.18E-30, is higher than the avg-p of one of its super concept, which contains RD 2022 and avg-$p = 1.79E$-30. This means that the superconcept is more interesting than it, *i.e.,* RD 2022 has a strong association with RDs of the discarded concept.



(a) (b)

Figure 6.7: Two examples of *(a)* selected and *(b)* discarded concepts from the lattice.

Gap-Based Selection

The number of selected concepts by the *Gap-based* method (see Algorithm 4) relies on the selection of the gap threshold value. Figure 6.8 shows the number of the concepts that are selected when using different thresholds. This number decreases when the gap threshold increases. Also, the figure shows the number of selected concepts when using different settings with the *Gap-based* algorithm such as "Gap+Statistical" and "Gap+Statistical Rule".

Definition 27 *("Gap+Statistical" method)*
"Gap+Statistical" applies the *Gap-based* method (Algorithm 4) on the concepts that have $support \geq 2$ and $avg\text{-}p \leq 0,05$ (3,870 concepts)

Definition 28 *("Gap+Statistical Rule" method)*
"Gap+Statistical Rule" combines both of the *Statistical* (Algorithm 2) and the *Gap-based* method (Algorithm 4) together. The resulted concepts should have $support \geq 2$ and $avg\text{-}p \leq 0,05$ that should be lower than the $avg\text{-}p$ values of their superconcepts. In addition, they should keep the gap between them and their superconcepts with the value of the gap threshold.

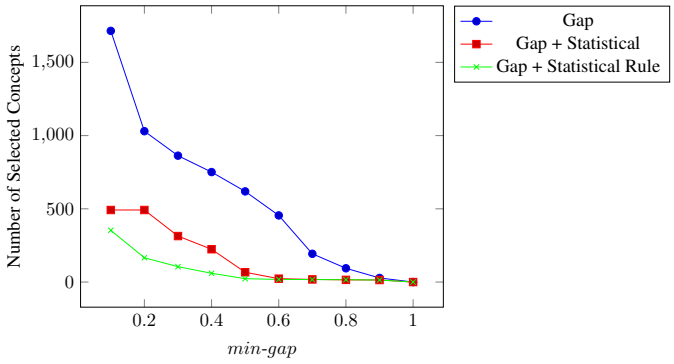


Figure 6.8: The number of selected concepts with "Gap", "Gap+Statistical" and "Gap+Statistical Rule" selection methods.

Figure 6.9 shows an example of selected concept, blue node, by *Gap-based* method. Its extent contains RDs with the following orphan numbers: 365 and 34587. They share 41 phenotypes. At $min\text{-}gap = 0.1$, we select this concept because there is a gap between it and one of its superconcepts higher than the threshold, which is 0.182. While at $min\text{-}gap = 0.2$, we discard this concept because all gaps between it and all of its superconcepts are lower than the threshold. This means that their superconcepts are very similar to this concept, while their extents are larger (*i.e.,* contain more RDs).

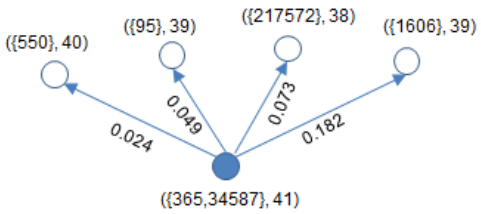


Figure 6.9: Example of selected concept.

Choosing The Gap Threshold

We conducted several experiments to study the effects of using a specific gap threshold. This could help at choosing the best gap threshold value based on the experiment results. These experiments rely on two different measures that are the stability and the similarity with the original classification.

Our first approach for defining $min\text{-}gap$ is based on the stability of the concepts. Stability is a measure for ranking concepts of a lattice that is first proposed by Kuznetsov, 1990 [Kuz90]. The idea behind stability is estimating the probability of preserving the concept intent when some objects of the context are removed. Stability of a concept is the relative number of subsets of the concept extent, denoted by $Ext(c)$, whose description is equal to the concept intent, denoted by $Int(c)$, where $\wp(X)$ is the power set of X. Given a concept c, its stability $Stab(c)$ is defined as:

$$Stab(c) = \frac{|\{s \in \wp(Ext(c))|s\prime = Int(c)\}|}{|2^{Ext(c)}|} \tag{6.4}$$

In this approach, we choose the gap threshold that selects a set of concepts with the highest $average\ stability$. Figure 6.10 shows that $min\text{-}gap = 0.9$ gives the best $average\ stability = 0.489$. For the other two concept selection methods "Gap+Statistical" and "Gap+Statistical Rule", the best $min\text{-}gap$ values are 0.7 and 0.7 with average stability values 0.413 and 0.409 respectively.

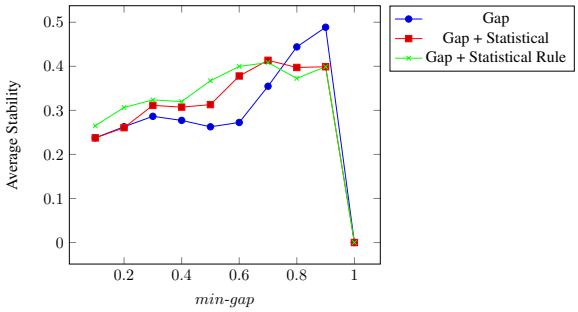


Figure 6.10: The average stability values for Gap, Gap+Statistical and Gap+Statistical Rule methods.

It is noted from the previous results that the highest average stability value does not exceed 0.5 (average stability=0.489) for the best $min\text{-}gap = 0.9$. This value is reasonable because most of the concepts have stability lower than 0.5. Figure 6.11 shows the number of concepts by using different stability thresholds. It shows that the number of concepts that have stability > 0.5 is 462 ($\simeq$ 10% of all concepts) and the number of concepts that have stability ≤ 0.5 is 4,367 ($\simeq$ 90% of all concepts).

Figure 6.12 presents the results of our second approach for the gap threshold selection based on the similarity with the original classification. Here, we choose the threshold that selects a set of concepts that are the most similar

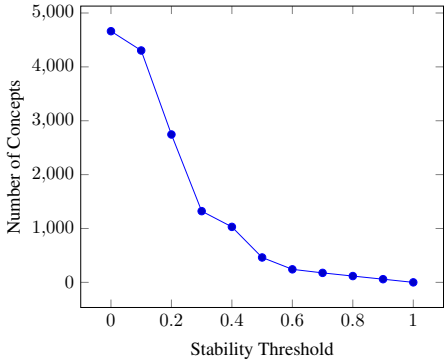


Figure 6.11: The number of concepts based on the stability threshold.

to the original classification. As shown in the figure, $min\text{-}gap = 0.8$ gives the best similarity of 0.505. For the other two concept selection methods "Gap+Statistical" and "Gap+Statistical Rule", the best $min\text{-}gap$ values are 0.8 and 0.9 with similarity values 0.603 and 0.599 respectively.

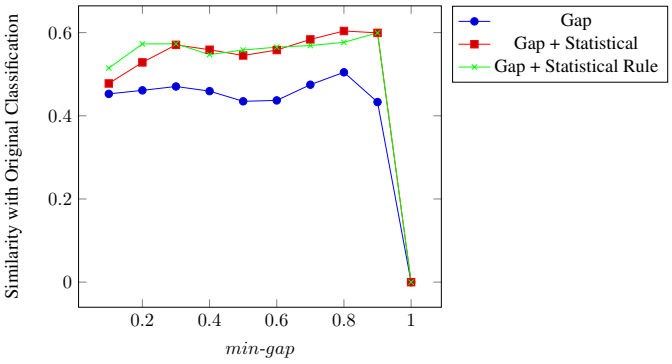


Figure 6.12: The similarities with the original classification for Gap, Gap+Statistical Selection and Gap+Statistical Rule methods.

The similarity between the selected sublattice (*i.e.,* the selected concepts fro the lattice) and the original ontology is computed by the following equation:

$$sim(LC, G) = \frac{\sum_{\forall c_i \in LC} max\_sim(c_i, G)}{|LC|} \tag{6.5}$$

$$max\_sim(c_i, G) = max(sim(c_i, rd_j), \forall rd_j \in G) \tag{6.6}$$

$$sim(c_i, rd_j) = \frac{|EXT(c_i) \cap rd_i|}{|EXT(c_i) \cup rd_i|} \tag{6.7}$$

where LC is the list of selected concepts and G is the set of RD classes of O^{rd}.

Comparison with Random Selection

In the following, as there is no benchmark to compare with our selection methods, we consider a random selection as a baseline method. The random selection keeps randomly some concepts from the set of all 4,662 concepts.

The number of randomly selected concepts is set every time to be equal to the number of selected concepts by every method of our selection methods (these numbers are shown in Figure 6.8). This ensures comparing two sets of concepts (*i.e.,* one set selected randomly and one set selected by one of our methods) with the same size. We proposed to compare these two sets on the basis of their average stability, similarity with the original classification and $avg\text{-}p$ values. In the following, all results of the random method are the average results of running it 10 times.

Figure 6.13(a), 6.13(b) and 6.13(c) show the comparison of the methods "Gap, "Gap+Statistical" and "Gap+Statistical Rule" respectively with the random method using the average stability. Figure 6.13(a) shows Random method gives a constant average stability value ($\simeq 0.29$) over all $min\text{-}gap$ values. When $min\text{-}gap \leq 0.6$, Gap method provides average stability lower than the random method. When $min\text{-}gap > 0.6$, Gap method provides average stability higher than the random method. The best average stability is achieved by Gap method when $min\text{-}gap = 0.9$. Figure 6.13(b) shows the average stability values of "Gap+Statistical" method against the random method. At $min\text{-}gap \leq 0.2$, 'Gap+Statistical" method provides average stability lower than the random method. When $min\text{-}gap > 0.2$, "Gap+Statistical" method provides average stability higher than the random method. The best average stability is achieved by "Gap+Statistical" method when $min\text{-}gap = 0.7$. Figure 6.13(c) shows the average stability values of "Gap+Statistical Rule" method against the random method. At every point, "Gap+Statistical Rule" method provides average stability higher than the random method. It achieves the best average stability at $min\text{-}gap = 0.7$.

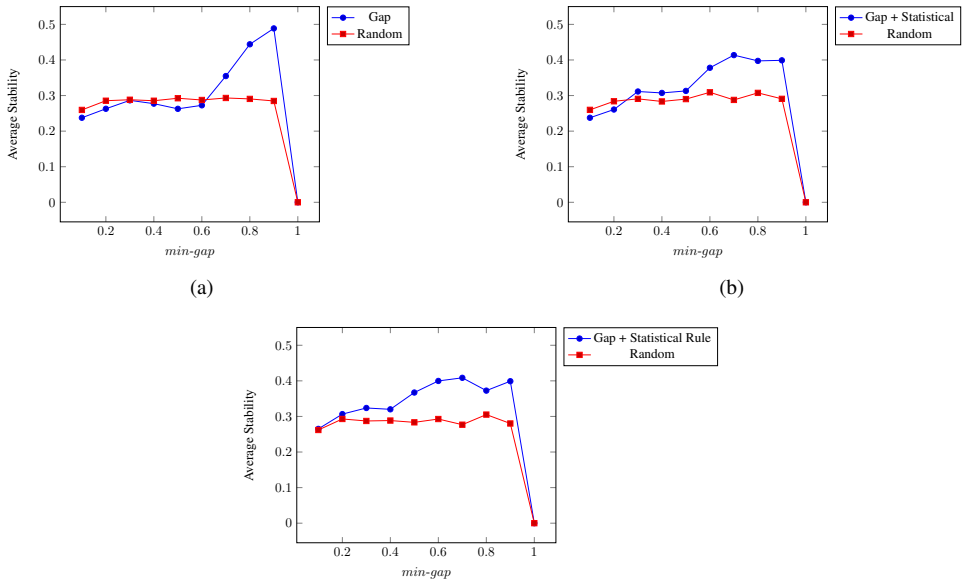


Figure 6.13: The comparison with Random method using the average stability.

Figure 6.14(a), 6.14(b) and 6.14(c) show the comparison with the random method using the similarity with the original classification. They show that Gap, "Gap+Statistical" and "Gap+Statistical Rule" give better similarity values than the random method.

Figure 6.15(a), 6.15(b) and 6.15(c) show the comparison with the random method using the $avg\text{-}p$ value. They show that "Gap+Statistical" and "Gap+Statistical Rule" achieve the best results as they give smaller $avg\text{-}p$ values than the random method, while Gap method gives higher $avg\text{-}p$ value than the random method.

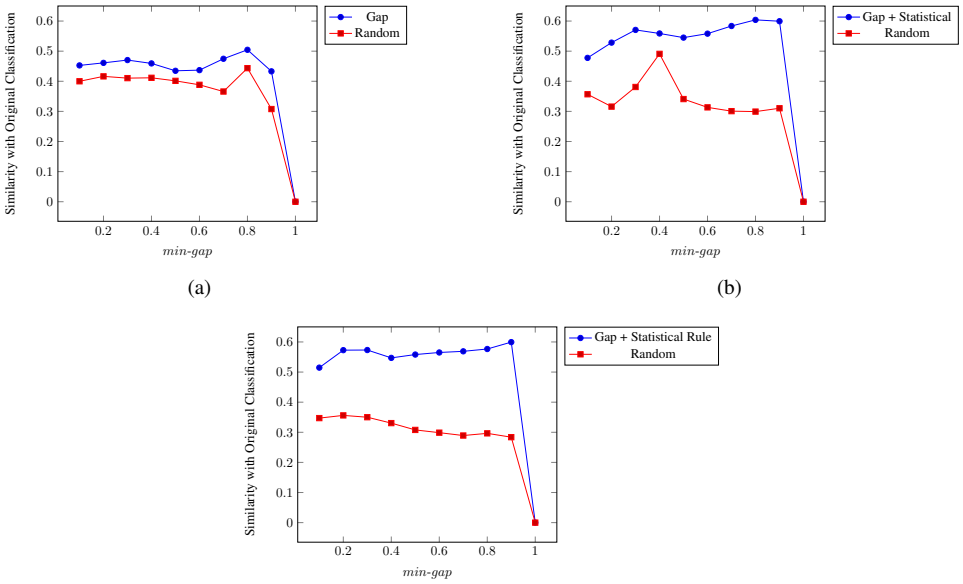


Figure 6.14: The comparison with Random method using the similarity with the original classification.

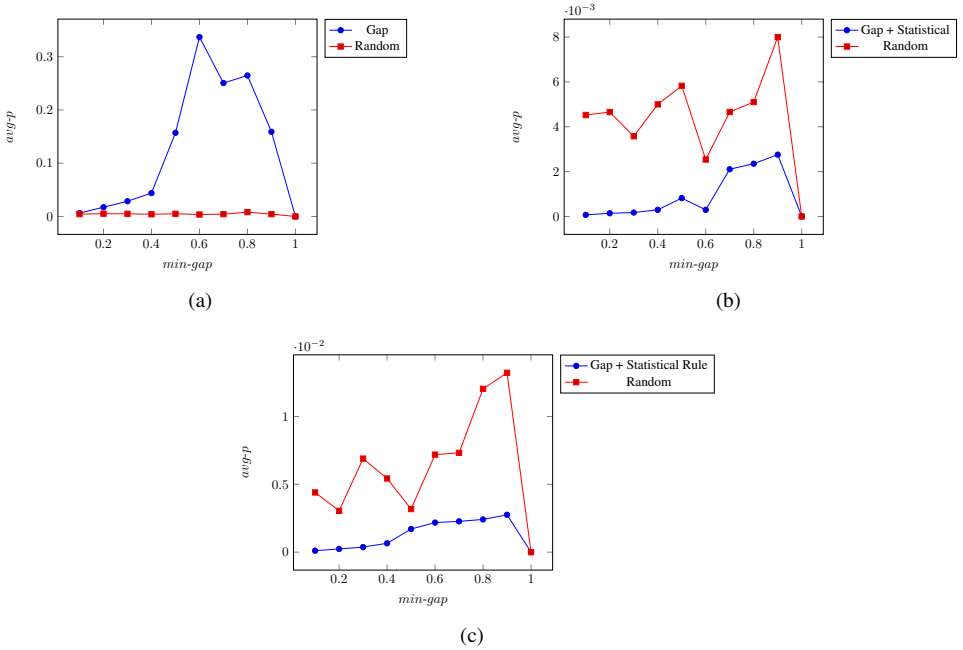


Figure 6.15: The comparison with Random method using $avg\text{-}p$ value.

6.6 Related Works

As this chapter proposes a pattern structure to classify diseases with respect to a phenotype ontology and to enrich an ontology, this section presents some related works about diseases classification and works that use FCA and pattern structures for classification.

Finding disease similarity for classifying diseases and their usage for data or ontology enrichment can be classified into different categories: function-based [SDC$^+$10, MD12, HSG15], semantic-based [LGC$^+$11], hybrid [CLJ$^+$14, OTO15, CWS$^+$16] and lattice-based [KK12b] approaches. This section details these different approaches.

Suthram *et al.* (2010) [SDC$^+$10] introduced a function-based approach for measuring the similarities between diseases. Their approach integrates multiple datasets including gene expression and protein-protein interaction networks. They employed the partial correlation coefficient to measure the similarity between two diseases based on their gene and protein lists. Their approach helps to discover human disease relationships in a systematic and quantitative way. Mathur *et al.* (2012) [MD12] defined a disease similarity approach based upon their common set of genes (gene-based disease similarity) and set of biological processes (process-based disease similarity). They presented a function based on co-occurrence and information content to measure the similarity between a pair of ontological terms (e.g., genes in GO [Ash00]), or entities annotated with them. Hoehndorf *et al.* [HSG15] developed a human diseasome, disease network, using the phenotype similarity between diseases. They used a text mining approach for extracting phenotypes of diseases from text. Then, they used scoring functions such as normalized point-wise mutual information for ranking these extractions. Finally, they build a disease network to cluster together diseases that have similar phenotypes. As a result, they create a resource that associates phenotypes with diseases in the Human Disease Ontology (DO).

Semantic-based approaches benefit from the ontology structure maintaining the concepts to compute their similarities [PFF$^+$09]. Li *et al.* (2001) [LGC$^+$11] presented a semantic similarity measure to compute disease similarity and gene similarity. They used Disease Ontology (DO) [SAN$^+$12] to compute the semantic similarity between diseases. Gene similarity is computed in terms of diseases. The gene is represented by its set of DO term annotations, and semantic similarity is calculated between terms in one set and terms in the other. This approach helps to detect disease-driven gene modules and also to annotate the modules for biological functions and pathways. Mungall *et al.* [MKR$^+$16] proposed an algorithm called k-BOOM for ontology construction. k-Boom aims at building a disease ontology by merging a mixture of disease ontologies, databases and vocabularies such as OMIM, DO, Orphanet and MESH. The algorithm first generates a probabilistic ontology with prior probability. Then, it estimates the most likely ontology by maximizing the posterior probability for the different combination of ontology axioms.

Cheng *et al.* [CLJ$^+$14] presented a hybrid approach, named SemFunSim, that combines both a function-based approach (FunSim) and a semantic-based approach (SemSim). For disease similarity, FunSim uses disease-related gene sets in a weighted network to calculate the disease similarity, while SemSim uses the relationship between two diseases in DO. SemFunSim helps to understand the associations between diseases. It also provides an effective way to find potential therapeutic chemicals for diseases. Cheng *et al.* [CWS$^+$16] developed an online system, named DisSim, for finding the disease similarity and offering their potential therapeutic drugs. DisSim provides semantic-based and functional-based methods (*e.g.,* SemFunSim, Resnik [Res95]) to compute the similarity between DO terms. It also provides the statistical significance of the similarity score in terms of $p\text{-}value$. Omura *et al.* [OTO15] developed a recommendation algorithm for clinical decision support system. This algorithm relies on measuring the similarity between diseases using a disease knowledge base that contains diseases, symptoms and disease-symptom relations. They introduced three different measures for disease similarity: *(a)* disease similarity based on the distance between diseases in ICD classification [201]; *(b)* probability-based approach using the symptom lists of diseases and their symptoms frequencies; and *(c)* machine learning approach hybridizes the

two approaches *(a)* and *(b)* and uses features list coming from these two approaches.

Keller *et al.* [KK12b] used FCA to study the disease similarity based on their shared genes. They developed a formal context (G, M, I) where G is a set of diseases, M is a set of genes and $(g, m) \in I$ if the disease g is associated with the gene m according to reference databases. The generated lattice from this formal context has been used to study the diseases similarity and identify the complexity of the relationships between them. Also, it has been used to identify new concepts from the lattice, representing the most strongly related disease families whose genes are candidates for further analysis.

Few works defined pattern structures over ontologies. They group objects by using the similarity of their descriptions defined in an ontology. Coulet *et al.* [CDKN13] used the "convex hull" operation to define the similarity between descriptions consisting of ontology classes in order to analyze ontology-based annotations of biomedical data. Alam *et al.* [ABNS15] revisited the pattern structures for structured attribute sets. They used the LCA to compute the concept lattice from the antichains in a taxonomy.

6.7 Conclusion

In this chapter, we introduced an approach based on pattern structures and ontologies for classifying RDs based on their phenotypic descriptions, which are organized in a phenotype ontology. Pattern structures use a similarity operator based on LCA to find the similarity between two RDs or more. This similarity operator considers both a disease ontology and a phenotype ontology for computing the similarity. The output of the pattern structures is a concept lattice that provides a new RD classification based on both their RD classes and their common phenotypes. The comparison of the resulting lattice with the original RD classification provides new RD classes that we suggest for enriching the original RD classification. We experimented our method on "Rare cardiac diseases" classification that contains 207 RDs. As only a few RDs have phenotypes in Orphadata, we used our SPARE* method that is a text mining approach for extracting RD phenotypes from relative PubMed abstracts. The resulting lattice contains 4,829 concepts, which are a large number of concepts to be considered by experts. As we are interested in finding the association between two RDs or more, we first discarded all concepts that have only one RD in their extents. This produces 4,662 concepts, which are still a large number of concepts. Therefore, we provided two different concept selection methods for finding the most interesting ones among these concepts. The first method used the $p\text{-}value$ which is a statistical significance measure that we used to measure the significance of how strong the association between two RDs based on their shared phenotypes. As the concept extent may contain more than two RDs, we compute its average $p\text{-}value$, denoted $avg\text{-}p$. We considered a concept is of interest if its $avg\text{-}p \leq 0.05$ and less than the $avg\text{-}p$ of any of its superconcepts. This selects 934 interesting concepts. The second method, called Gap, is based on the gap or the dissimilarity between a concept and its superconcepts. We discard the concepts that are very similar in term of their phenotypes (*i.e.,* have small gap or dissimilarity) to one of their superconcepts, as this superconcept has similar set of phenotypes but more RDs in its extent. We combine this method with the first method and we designed three possible methods: *(1)* "Gap", *(2)* "Gap + Significance" and *(3)* "Gap + Significance Rule". Their comparison with a random method shows that they give a higher average stability and similarity with the original classification. "Gap + Significance" and "Gap + Significance Rule" methods give lower $avg\text{-}p$ values than the random method, but Gap method gives higher values. This is because of the Gap method does not consider any $p\text{-}value$ computation in their concept selection like what is done in the other two methods.

7

Conclusion and Perspectives

Contents

This thesis discuss two main topics: *(1)* the information extraction from texts and *(2)* the classification and ontology enrichment using pattern structures. We summarize here our main contributions and present some perspectives of this thesis work.

7.1 Summary of the Contributions

7.1.1 Extracting Relationships from Texts

In Chapter 4, we introduced a hybrid method, called SPARE*, for extracting relationships from text. SPARE* combines a pattern-based method and a Machine Learning (ML) method. The pattern-based method, called SPARE, learns syntactic patterns from the dependency graphs (DGs) of sentences. These syntactic patterns are learned from the shortest paths between two entities (*e.g.,* disease, phenotype) in the DGs. To reduce the number of generated patterns, two shortest paths (or more) have been merged and represented in one generalized pattern. Different shortest paths are aggregated to one pattern if those share the same edges and directions. To learn high-quality patterns, we introduced a *positive-predictive value* for selecting a precise set of patterns. SPARE shows a good precision but low recall. Therefore, we used SVM as a ML method that shows the highest recall value among different ML methods (*e.g.,* rule-based methods, decision tree methods, Naïve Bayes, SVM) we experimented for classifying relationships. The relationship extraction task has been considered as a binary classification task where SVM classifies a relationship as True for correct relationship or False for incorrect relationship. SVM relies on multiple features that have been extracted from sentences containing a relationship. Finally, we experimented different possible combinations between SPARE and SVM to choose the combination that achieves the best $F\text{-}measure$. This hybrid approach shows an improvement in $F\text{-}measure$ with 16% and 3% over SPARE and SVM respectively. We applied SPARE* for extracting Disease-Phenotype (D-P) relationships from biomedical texts, where diseases and phenotypes are annotated in the text by a NER (MetaMap).

7.1.2 Identification of Complex Entities from Text

In Chapter 5, we described a relationship-based method for identifying complex entities (*e.g.,* phenotypes) that are in a relation with other entities (*i.e.,* diseases) from text. We employed SPARE* for achieving this goal. SPARE has been used for learning a set of high-quality syntactic patterns. Then, we introduced a $specificity$ measure to keep the patterns that are specific for relationships between two interesting entities *i.e.,* relationships only between a disease and a phenotype. Patterns with $specificity \geq min\_specificity$ are kept as they are considered specific to D-P relationships. We relaxed these patterns on the phenotype constraint to enable identifying phenotype candidates that are not identified by NER tools. SVM is combined with SPARE for classifying the candidates extracted (*e.g.,* phenotypes) by the syntactic patterns. Then, we compared these candidates with phenotypes listed in ontologies such as HPO to validate their correctness and to assess their novelty. This comparison is based on a compositional semantics model and a set of manually defined mapping rules. The results show the feasibility of our approach for discovering new phenotypes that were not referenced in phenotype databases and ontologies, and may involve complex phenotype descriptions.

We applied and evaluated SPARE* to enrich the content of Orphanet and Orphadata. SPARE* extracts phenotypes of Rare Diseases (RDs) from related PubMed articles. The extracted phenotypes are then compared to those listed in Orphanet and Orphadata. We proposed a compositional semantics model and manually-defined mapping rules for identifying which phenotypes are known, *i.e.,* they already exist in Orphanet and Orphadata, and which phenotypes are potentially new. New phenotypes are proposed to refine the content of Orphanet summary and Orphadata.

7.1.3 Pattern Structures for Classification and Ontology Enrichment

In Chapter 6, we used pattern structures for classifying RDs and for enriching the content of an initial RD ontology. We defined a meet operator that groups a set of RDs on the basis of their classes in the initial RD ontology and their common phenotypes. This operator respects the ontology structures of the RDs and phenotypes. To complete RD descriptions, we used SPARE for extracting RD phenotypes, from related PubMed abstracts, for RDs that do not have a phenotype in databases such as Orphadata. The lattice generated by pattern structures provides a new classification for RDs based on their phenotypic descriptions. This classification contains new RD classes (*i.e.,* concepts of the lattice) that we suggest to enrich the initial RD ontology. Indeed, the number of the concepts of the lattice is large, which makes their analysis a difficult task. Therefore, we provided two different selection methods for selecting a reduced set of interesting concepts among them. The first method is based on the $p\text{-}value$, named $Statistical$ method, to evaluate how strong is the association between the RDs grouped in the concept extent. The second method, called Gap, is based on the gap between the concepts and their superconcepts. This gap is measured by the dissimilarity between their phenotype sets of the concept intents. Several experiments have been conducted to evaluate the Gap method and its combination with the $Statistical$ method. This evaluation is achieved by the comparison of these methods with a random selection as a baseline method. The comparison is achieved on the basis of their average stability, the similarity with the initial ontology and the average $p\text{-}values$. Finally, we discussed the results of our methods.

7.2 Future Directions & Prospects

In chapter 4, we presented SPARE that is a pattern-based method for relationship extraction from text. Syntactic patterns are learned from a set of DGs. An interesting point that we would like to investigate in the future is to study the different representations of texts such as a sequence of tokens (*e.g.,* words, lemmas, POS), syntactic trees and semantic parsing for generating a different set of syntactic patterns. Another possibility could be defining a

method for generating a set of constrained patterns or rules that benefits from all these different text representations (DG, syntactic tree and semantic parsing). Also in this chapter, we used SVM that relies on a set of features for a relationship classification. This set of features is mainly based on the sequence of words, their POS and features generated from the DG. A possibility for the improvement is to extend the set of features to include other additional features generated from the syntactic parsing and the semantic parsing of the text.

Negation is a linguistic phenomenon where a negation word (*e.g.,* not, without) can change the meaning of a sentence. Identifying the scope of negations helps at qualifying a relationship. One pattern could extract a relationship that is a false relationship because of the negation. A possible solution could be enriching the patterns by adding constraints to handle the negation cases. For building these constraints, we could use negation clues such as the words No, not, neither, without, lack, fail, unable, absence, prevent and unlikely as an indicator for the negation. This information can be easily extracted from the dependency relations of a DG. Another method could be learning a binary negation classifier using features extracted from the discourse of the text.

Syntactic patterns have been learned and tested using a corpus retrieved from biomedical publications. Both learning and testing corpora are manually annotated by only one person to identify true and false relationships. The annotation task mainly requires linguistics and NLP skills. A future work is to involve several experts (from linguistics and biomedical domains) in the annotation process and then provide an Inter-Annotator agreement measure between the different annotations. This may produce a high-quality corpus that could enrich the RE process.

SPARE* was designed only for extracting binary relationships between two interesting entities. Interesting issue is to enrich SPARE* to extract n-ary relationships that relate more than two entities. A possible solution could be learning syntactic patterns that aggregate the relationships between n entities at the same time. Another solution could be learning different syntactic patterns that identify the relationships between each two entities.

In this thesis, we focus on extracting intra-sentential relationships, *i.e.,* relationships between two entities found within a single sentence. As a future perspective, we want to adapt our method to work on a cross-sentential RE where relationships between entities beyond sentence boundaries and can be asserted over many sentences. One possible solution could use co-reference resolution, which is the task of finding all expressions that refer to the same entity in a text. It could help at finding the reference of one entity in a sentence where the second entity is located and then we can use a RE method to check if they are associated or not.

In Chapter 5 we proposed a relationship-based approach for the identification of complex phenotypes. We used SPARE* for identifying new complex phenotypes that are related to RDs. This approach is used to enrich the content of Orphanet summary (and also the content of Orphadata) by suggesting new RD phenotypes. One interesting topic is to extend our approach not only for identifying RD phenotypes but also for identifying other entities that are related to RDs such as genes, drugs and treatments. This may be achieved by learning new syntactic patterns that are specific for these entities.

At the time of writing this thesis, deep learning is a promising approach in machine learning that already show its efficiency for several text mining tasks. In this thesis, we do not use deep learning but we believe it would be a good extension to our RE method that may improve performances. Deep learning is a numerical approach where its results are difficult to interpret, while our pattern-based approach is symbolic where its results are easy to interpret. In this thesis we combined SPARE with SVM and this combination showed good results. We think that deep learning could be a good alternative to SVM as a numerical approach to combine with SPARE.

In Chapter 6 we proposed an approach based on pattern structures for classifying RDs based on their phenotypes and for enriching an initial RD ontology. In the future, we propose to use more resources about RDs and phenotypes such as OMIM and HPO phenotype. These resources could provide a more complete descriptions about RDs. Also, we propose to add other medical terms such as treatments and genes as additional dimensions in the pattern structure context. This may help to easily associate RDs with similar phenotypes to similar diagnoses and treatments simultaneously. Pattern structures could be a useful tool for this kind of tasks as it shows its

ability to work with complex descriptions including several dimensions (*e.g.,* diseases, phenotypes) and involving knowledge resources (*e.g.,* ontologies).

In Appendix C, we presented some screenshots of our software for SPARE*. It is a prototype that will be demonstrated to curators at Orphanet, with the idea of guiding their annotation and population works. Also, we are packaging SPARE* to facilitate its reuse and to share with others in the community.

List of Figures

List of Tables

A

An Example of Orphanet Summary

Orphanet presents information related to RDs. In this appendix we present an example from the content of Orphanet for the Kennedy disease.

Figure A.1 presents meta-information, offered by Orphanet, about the Kennedy disease, including it's synonyms, the prevalence, inheritance and age of onset. It also provides the cross-references with other medical databases and resources such as ICD-10, OMIM, UMLS.

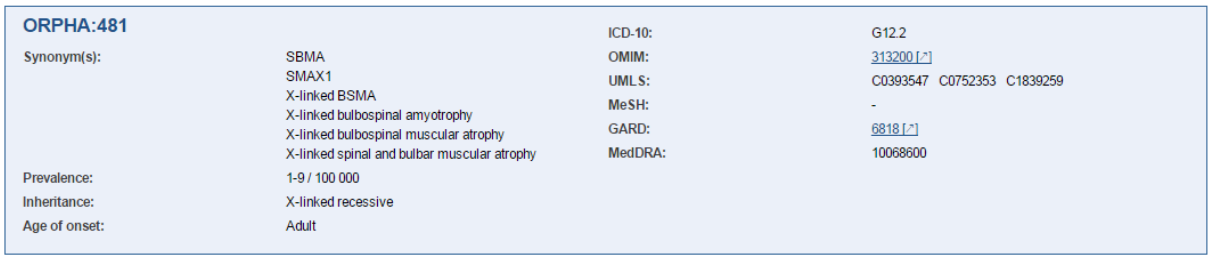


Figure A.1: Meta-information of the Kennedy disease in Orphanet

Figures A.2 shows the Orphanet summary of Kennedy disease. This summary is available at Orphanet website [Orp15b] and you can retrieve by searching with disease name "Kennedy disease" or by it's Orpha number 481. This summary is structured into different sections that provide different information, including disease definition, epidemiology, clinical description, etiology, diagnostic methods, differential diagnosis, antenatal diagnosis, genetic counseling, management and treatment and prognosis. These sections are different from disease to another disease. It's noted that the last updated date of this summary is July 2011.

This summary contains phenotypes related to Kennedy disease. An expert annotated manually these phenotypes from disease definition, clinical description and Etiology sections. Table A.1 shows the complete list of these phenotypes.

SUMMARY

Disease definition

Kennedy's disease, also known as bulbospinal muscular atrophy (BSMA), is a rare X-linked recessive motor neuron disease characterized by proximal and bulbar muscle wasting.

Epidemiology

The prevalence of BSMA is 1/30,000 male births. The incidence is 1/526,315 males/year.

Clinical description

Disease onset occurs between 30-60 years of age. Initial clinical manifestations include tremor, muscle cramps, muscle twitching, fatigue and slurred speech. With disease progression patients additionally develop weakness and wasting of the limb and bulbar muscles, manifesting as dysarthria, dysphonia, hanging jaw, tongue wasting, chewing difficulty and impaired mobility. Intellectual decline is minimal to none. In the terminal stages of the disease some patients may be unable to swallow or breathe. Non-neurological manifestations include gynecomastia, hypogonadism (leading to infertility and impotence) and in rare cases Dupuytren's contracture, or groin hernia.

Etiology

BSMA is caused by an unstable expansion of a CAG triplet repeat (40-62 repeats) in exon 1 of the androgen receptor (AR) gene on chromosome Xq11-12. The abnormally increased repetition of this CAG triplet leads to an expanded stretch of glutamines within the androgen-receptor (AR). Polyglutamine-expansion results in misfolding and proteolysis of the mutated AR, rendering it insensitive to androgen hormones. In the nucleus AR fragments are produced, which aggregate and these aggregates are believed to cause dysregulation of the transcription of various other proteins and consecutively lead to motor neuron degeneration. Without a sufficient number of motor neurons, initiation and maintenance of muscle contractions can no longer occur, leading to progressive muscle wasting. Recently, a BSMA phenotype with distal predominance of limb weakness and wasting has been reported, caused by mutations in a subunit of the dynactin 1 *DCTN1* gene.

Diagnostic methods

Diagnosis is established upon medical history, clinical examination, elevated creatine-kinase, testosterone, progesterone, follicle-stimulating hormone, luteinizing hormone, reduced nerve conduction velocities or reduced nerve action potential amplitudes, acute or chronic denervation and re-innervation on electromyography and documentation of the mutation.

Differential diagnosis

Differential diagnoses include hereditary spastic paraplegia, spinocerebellar ataxia (see these terms), other motor neuron diseases, myopathies, neuropathies, lead or aluminum poisoning, and cervical spondylosis.

Antenatal diagnosis

Antenatal diagnosis is possible for mothers carrying the mutation.

Genetic counseling

Female mutation carriers usually do not manifest clinically but transmit the mutation to 50% of their male and female offspring. Affected males do not transmit the disease but 100% of their daughters become mutation carriers.

Management and treatment

Symptomatic treatment includes physiotherapy and rehabilitation, agents against tremor and muscle cramps and hormone therapy or surgical treatment of gynecomastia. Recently, treatment of patients with the anti-testosterone leuprorelin was found to be beneficial. In advanced stages of the disease, tube feeding or ventilatory support may be indicated.

Prognosis

Disease progression is slow with only one third of patients requiring a wheelchair 20 years after diagnosis. Prognosis of BSMA is usually fair with only a small decrease in life expectancy.

Expert reviewer(s)

Pr Josef FINSTERER

Last update: July 2011

Figure A.2: The Orphanet summary of Kennedy disease. This screenshot is taken from the Orphanet website at 2/2017.

proximal muscle wasting	chewing difficulty	distal predominance of limb wasting
bulbar muscle wasting	impaired mobility	elevated creatine-kinase
tremor	intellectual decline	elevated testosterone
muscle cramps	unablility to swallow	elevated progesterone
muscle twitching	unablility to breathe	elevated follicle-stimulating hormone
fatigue	gynecomastia	elevated luteinizing hormone
slurred speech	hypogonadism	reduced nerve conduction velocities
weakness of the limb	infertility	reduced nerve action potential amplitudes
wasting of the limb	impotence	acute denervation
weakness of the bulbar muscles	Dupuytren's contracture	chronic denervation
wasting of the bulbar muscles	groin hernia	re-innervation
dysarthria	motor neuron degeneration	tremor
dysphonia	progressive muscle wasting	muscle cramps
hanging jaw	distal predominance of limb weakness	gynecomastia
tongue wasting		

Table A.1: The list of phenotypes that are identified by an expert form Orphanet summary of the Kennedy disease.

B

Linguistic and Syntactic Features Characterizing Considered Sentence

Feature	Description
NoWB	Number of words between DISEASE and PHENOTYPE
DL1, DL2, DLPOS1, DLPOS2	Lemma and POS tag of the first and second words left to DISEASE
DR1, DR2, DLPOS1, DLPOS2	Lemma and POS tag of the first and second words right to DISEASE
PL1, PL2, PLPOS1, PLPOS2	Lemma and POS tag of the first and second words left to PHENOTYPE
PR1, PR2, PLPOS1, PLPOS2	Lemma of the first and second words right to PHENOTYPE
DGLeastCommonRootLemma	The lemma of the root of DG between D-P
DGLeastCommonRootPOS	The POS of the root of DG between D-P
DGPathToDiseaseLemma	The path from the root to DISEASE (vertices are represented by the lemma of each word)
DGPathToPhenotypeLemma	The path from the root to PHENOTYPE (vertices are represented by the lemma of each word)
DGCompletePathLemma	The DG path between DISEASE and PHENOTYPE (vertices are represented by the lemma of each word)
DGPathToDiseasePOS	The path from the root to DISEASE (vertices are represented by the POS of each word)
DGPathToPhenotypePOS	The path from the root to PHENOTYPE (vertices are represented by the POS of each word)
DGCompletePathPOS	The DG path between DISEASE and PHENOTYPE (vertices are represented by the POS of each word)
e_walkToDisease	The DG relation sequence from the root to DISEASE (by keeping only the edges and removing all vertices)
e_walkToPhenotype	The DG relation sequence from the root to PHENOTYPE (by keeping only the edges and removing all vertices)
e_walkCompletePath	The DG relation sequence f between DISEASE and PHENOTYPE (by keeping only the edges and removing all vertices)
v_walkToDiseaseLemma	The DG vertex sequence from the root to DISEASE (by keeping only the vertices and removing all edges - vertex represented by lemma)
v_walkToPhenotypeLemma	The DG vertex sequence from the root to PHENOTYPE (by keeping only the vertices and removing all edges - vertex represented by lemma)
v_walkCompletePathLemma	The DG vertex sequence f between DISEASE and PHENOTYPE (by keeping only the vertices and removing all edges - vertex represented by lemma)
v_walkToDiseasePOS	The DG vertex sequence from the root to DISEASE (by keeping only the vertices and removing all edges - vertex represented by POS)
v_walkToPhenotypePOS	The DG vertex sequence from the root to PHENOTYPE (by keeping only the vertices and removing all edges - vertex represented by POS)
v_walkCompletePathPOS	The DG vertex sequence f between DISEASE and PHENOTYPE (by keeping only the vertices and removing all edges - vertex represented by POS)
Exact1VB	The DG path between D-P contains exactly one verb
Exact2VB	The DG path between D-P contains exactly two verbs
More2VB	The DG path between D-P contains more than two verbs
VB1	The lemma of the first verb in the DG path between D-P if exists
VB2	The lemma of the second verb in the DG path between D-P if exists

Table B.1: Features set for ML methods and their description. DG means Dependency Graph and POS means Part Of Speech. D and P stand respectively for the Disease and the Phenotype entities.

C

An Interactive Tool for Relationship Extraction

SPARE* is an automatic method for extracting information from texts. It is a hybrid of a pattern-based method, called SPARE, and a machine earnings method (SVM in our case). We implemented SPARE* method in a functional tool that consists of five modules: 1) visual annotation; 2) pattern learning; 3) relation extraction; 4) recognition and 5) application modules. This appendix presents these five modules from an application perspective.

C.1 System Modules

C.1.1 Visual Annotation Module

Visual annotation module (VAM) helps experts to annotate a set of texts (corpus) with the interesting entities and their relationships, *i.e.,* annotating a biomedical text with diseases, phenotypes and disease-phenotype (D-P) relationships. It provides a graphical user interface for facilitating the expert's work. It allows selecting a set of texts to annotate manually in addition to use annotations coming from other automatic annotation tool (*e.g.,* MetaMap) as well.

VAM provides a login frame as shown in Figure C.1. If the user is new, the module starts a new session. Otherwise, it opens the last session for the user. The main frame (see Figure C.2) presents the corpus sentence by sentence to be annotated by the users. One can navigate between the sentences by using "First", "Back", "Next", "Last" and "Go" buttons in the "Sentence Selection" panel. This panel shows the information of a given sentence, including the sentence itself, it's abstract PubMed ID, a list of diseases and the phenotypes in the sentence that are annotated manually or by MetaMap. It also provides the ability to add, delete and update a relation between a disease and a phenotype. VAM enables to mark a sentence to access it later for review. One can view the complete PubMed abstract (see FigureC.3) by pressing "Show Full Abstract" button. Figure C.4 shows the frame of marked sentences. Figures C.5 and C.6 show the frames for adding, deleting and updating the list of diseases and phenotypes. For instance, one can add a disease mention by typing directly the information in the input boxes or by highlighting the disease string within the sentence.

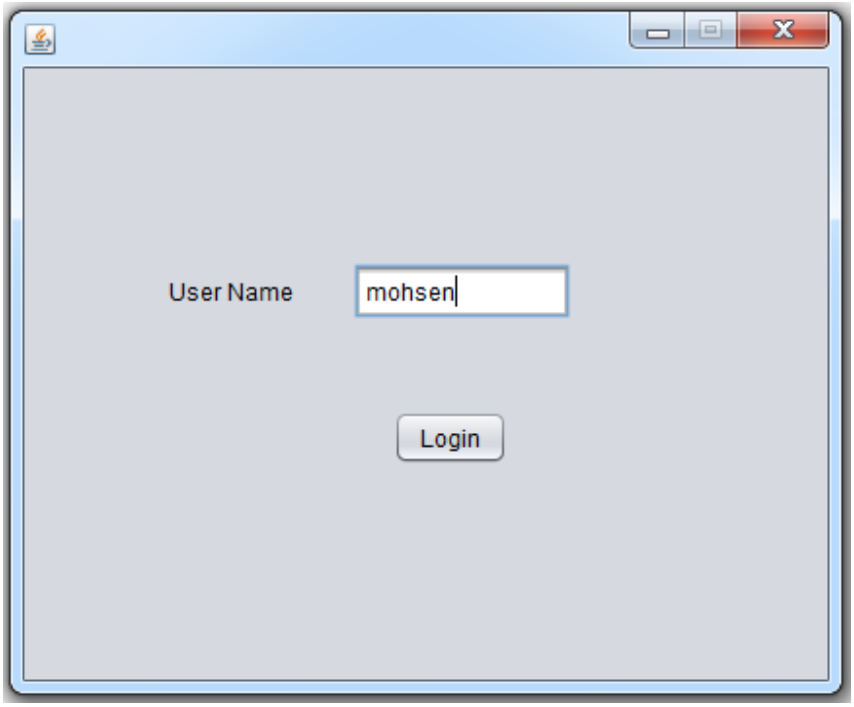


Figure C.1: Login Frame of VAM.

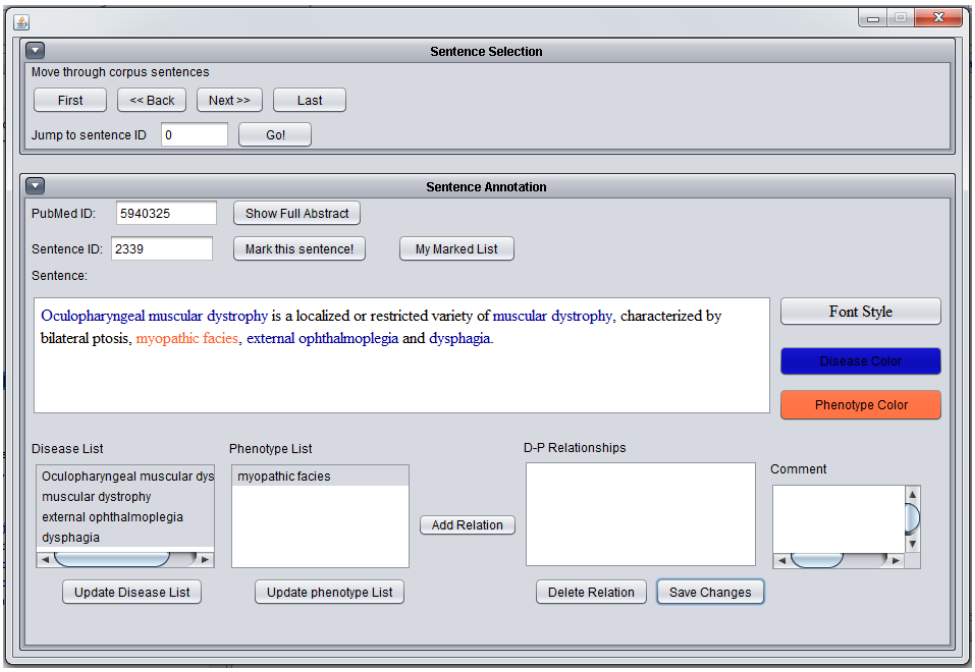


Figure C.2: Main Frame of VAM.

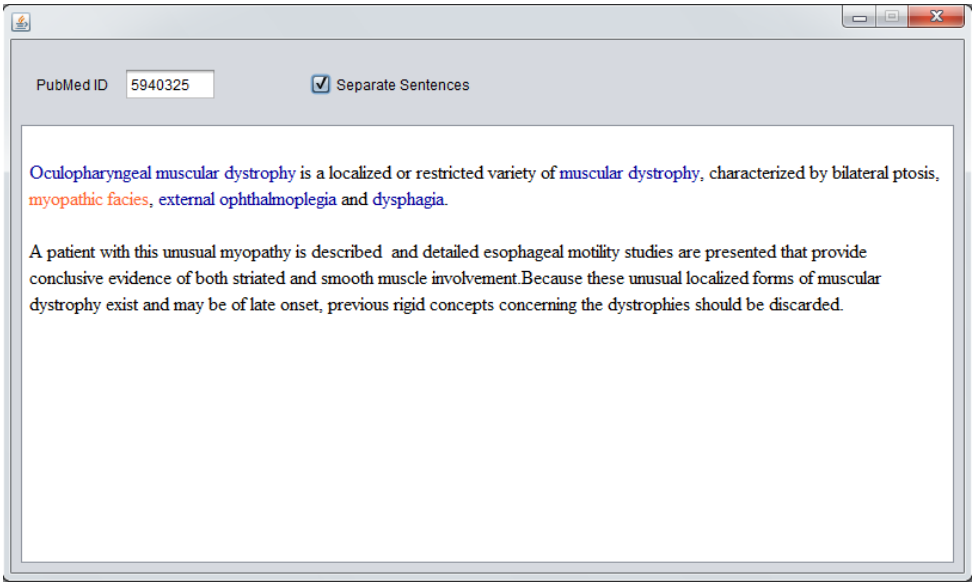


Figure C.3: Example of complete PubMed abstract annotated by diseases and phenotypes. Strings with blue color are diseases while strings with red color are phenotypes.

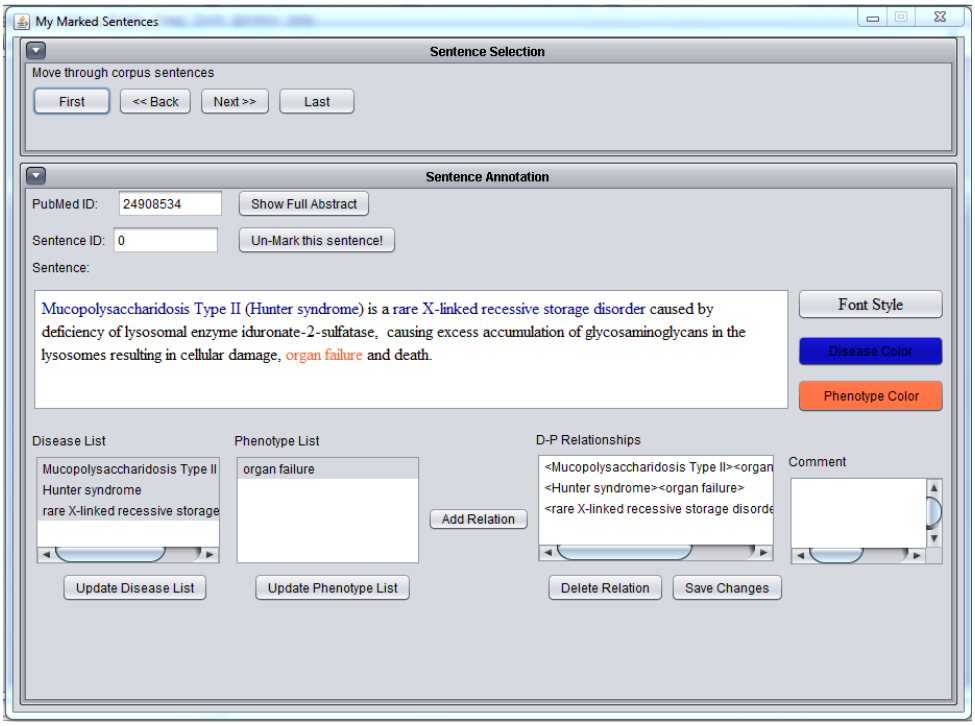


Figure C.4: The frame that shows the list of marked sentences.

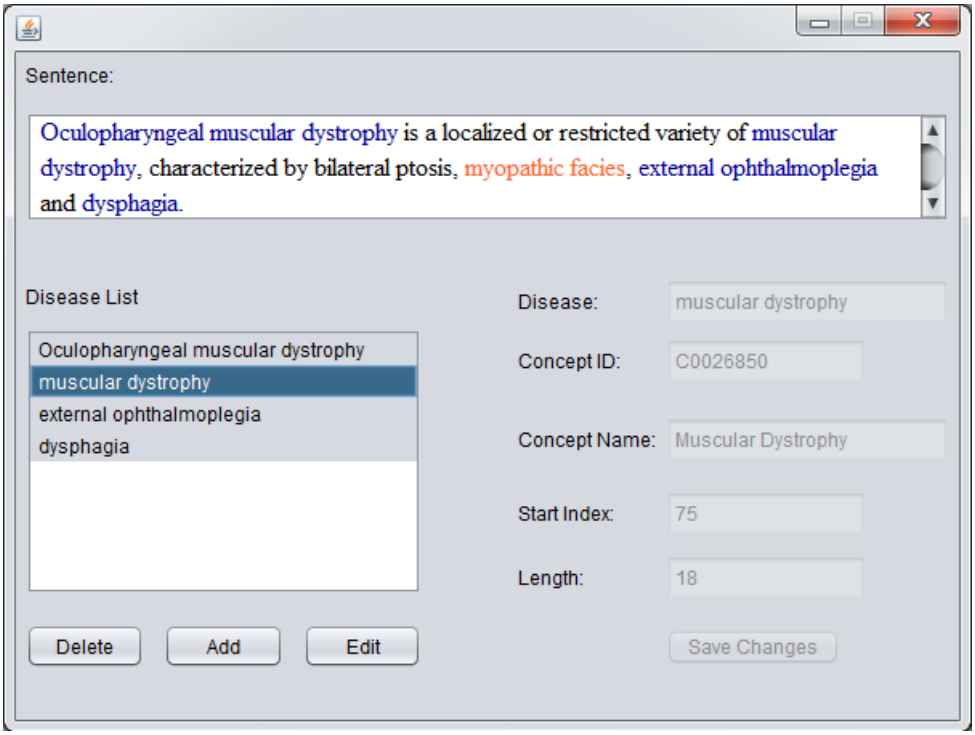


Figure C.5: The frame of editing disease mentions in a sentence.

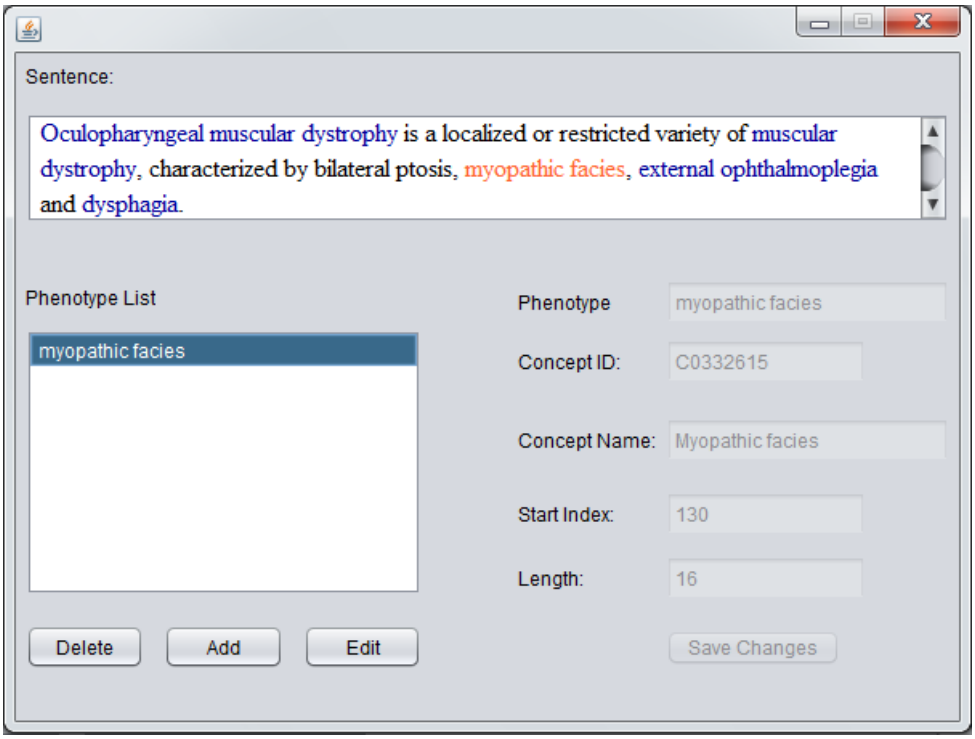


Figure C.6: The frame of editing disease mentions in a sentence.

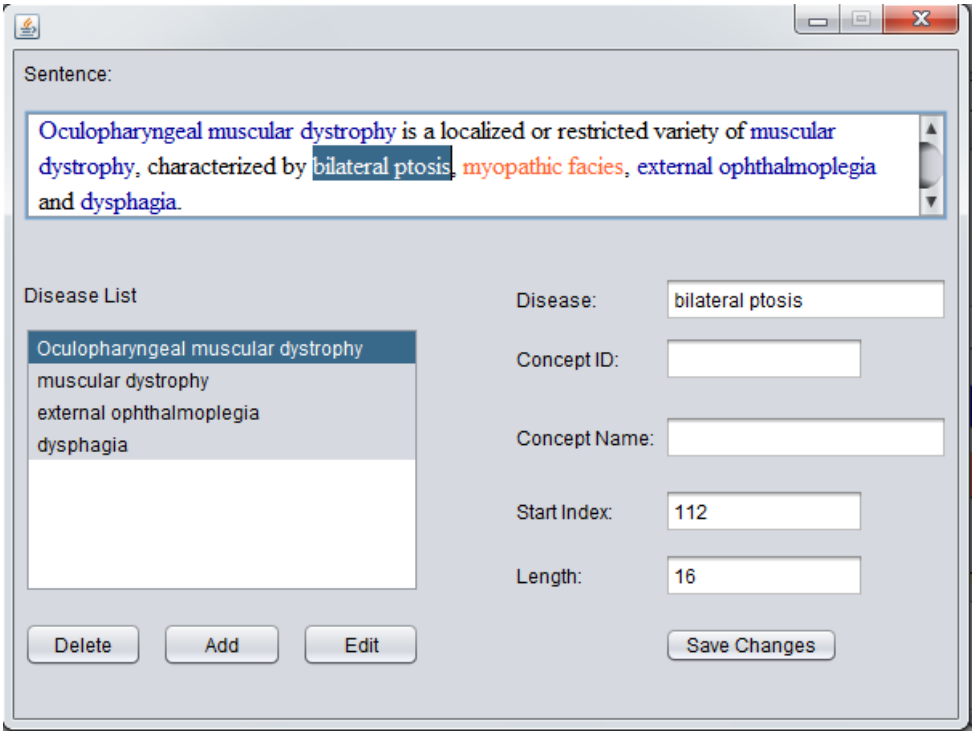


Figure C.7: Example of adding a disease mention.

C.1.2 Learning Module

Learning module learns the components of our SPARE⋆ method: *(1)* SPARE, *(2)* SVM and *(3)* the combination of SPARE and SVM. This subsection presents briefly the learning process of SPARE and SVM from and application perspective. More details about the approach are given in Chapter 4. Learning module uses an annotated corpus (*e.g.,* annotated corpus generated from VAM module or probably from any external tool). The corpus consists of a set of sentences containing at least one annotation of each entity composing the relationship, *i.e.,* at least one disease and at least one phenotype. It also provides annotations of relationships between the interesting entities, *i.e.,* annotations of D-P relationships. We implemented a software component, SPARE, to learn and evaluate syntactic patterns for relationship extraction from an annotated corpus. This component generates a set of patterns and defines their *support* and *PPV* values based on the learning corpus. We also implemented a software component to learn a SVM classifier for classifying relationships (*e.g.,* D-P relationships). The features defined in Table B.1 are extracted for each relationship in the learning corpus. Then, CfsSubsetEval, a feature selection method is available in Weka toolbox, is used to reduce the set of features and select the most important subset of features. Finally, the selected features are used as a feature vector for learning SVM classifier. We use LibSVM, which is available in Weka toolbox, for learning our SVM classifier. Finally, a combination component is used to combine the results of SPARE and SVM.

Currently, the learning module does not provide a visual interface. It is implemented as scripts of code that should be run in a sequence from any Java IDE (*e.g., NetBeans, Eclipse*) or from the command prompt.

C.1.3 Relation Extraction Module

Relation extraction module uses SPARE⋆ learned from the learning module for the relationship extraction task. A set of learned patterns is used to extract relationships from new texts (*e.g.,* extract D-P relationships where diseases

and phenotypes are previously identified). Then, SVM classifier is used to classify the extracted relationships. Finally, the combination component qualifies the final results.

C.1.4 Recognition Module

This module aims at learning a set of syntactic patterns for extracting relationships between specific entities, *i.e.,* extracting relationships between diseases and phenotypes only. To do that, we first get the learned patterns with high quality (*i.e.,* with PPV value higher than a specific threshold). Then, we relax the patterns on the second entity constraint of the relationship (*e.g.,* phenotype constraint) and then we computer their specificity based on the whole corpus. These patterns with high specificity are used for identifying the second entity of the relationships (*i.e.,* identifying phenotypes that are in a relation with diseases). Finally, they extracted candidates are then compared with terminologies of an ontology in order to validate and verify the correctness and the novelty of the extracted candidates. More details about this process are given in Chapter 5.

C.1.5 Application Module

We are describing here the application module that enables the comparison of SPARE*, Orphanet and Orphadata. It provides a graphic user interface that enables the user to select a target RD (*e.g.,* Kennedy disease). Then, the application shows some information related to the disease such as RD name, its orpha number, preferred name, a list of synonyms and the query submitted to PubMed to get RD abstracts from PubMed. When the user presses "Show Orphanet Summary Info" button, the application shows "Summary Info" frame. Figure C.10 shows the first part of this frame which contains the Orphanet summary of the RD (the user can get it online by pressing "Go online" button). The phenotypes selected by experts are highlighted in the summary text by red color (the user can change the font style of summary text and phenotypes). Figure C.11 shows the second part of "Summary Info" frame which presents a list of phenotypes annotated by experts from an Orphanet summary, a list of RD Orphadata phenotypes extracted from Orphadata and a list of SPARE* phenotypes extracted from its PubMed abstracts. Using Kennedy disease as an example, the figure shows that there are 43 (39 unique) phenotypes annotated by experts, 12 phenotypes extracted from Orphadata and 115 (100 unique) SPARE* phenotypes. Finally, the user can explore all possible mappings between these three phenotypes lists by pressing one of these three buttons "Mappings of SPARE* and Orphanet Phenotypes", "Mappings of SPARE* and Orphadata Phenotypes" and "Mappings of Orphanet and Orphadata Phenotypes". For instance, when the user presses the first one, a frame that presents the mappings between SPARE* and Orphanet phenotypes appear (see figure C.12). This frame contains all necessary information for each mapping such as the date and the PubMed ID of an abstract containing a phenotype, extracted phenotype by SPARE* and it's frequency in the abstracts, summary phenotype annotated by experts and its ID, the mapping/matching category, the similarity value between the two phenotypes and the comments that the expert can add to this mapping. In addition, it introduces a status of the information for the count of all different mapping categories. This frame also shows a recommendation of Orphanet summary improvements by providing a list of new SPARE* phenotypes extracted from PubMed abstracts. Similar frames are also available for SPARE*-Orphadata and Orphanet-Orphadata mappings by clicking the other two buttons.

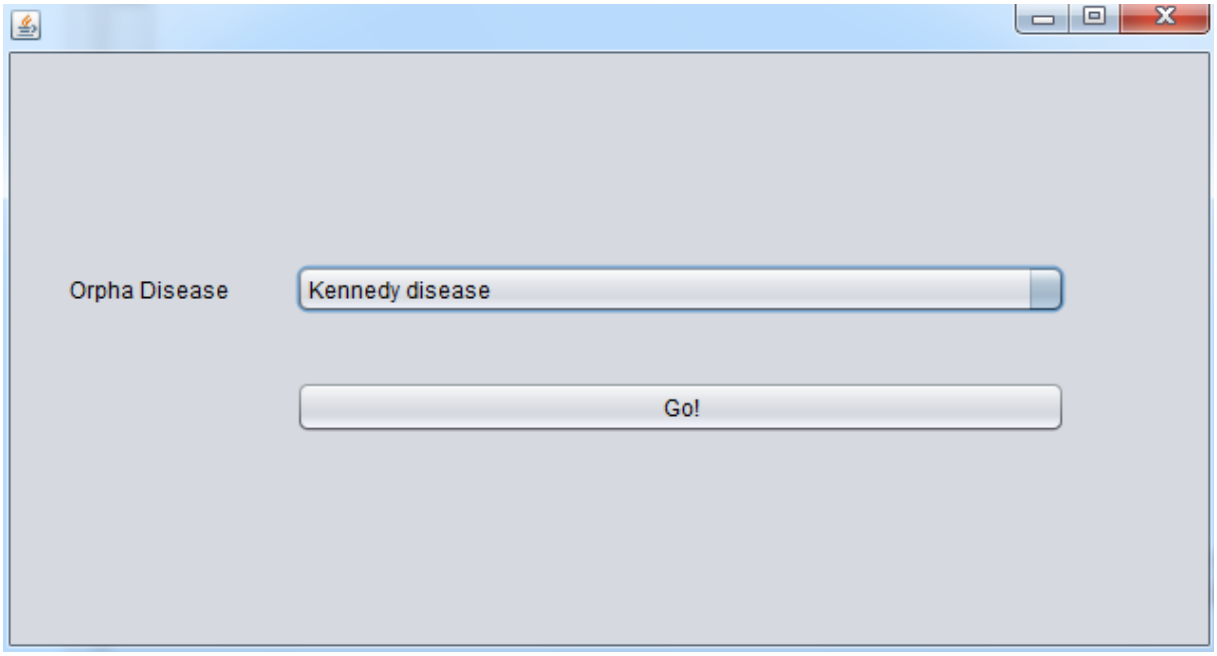


Figure C.8: This frame allows to select a RD from a list of RDs.

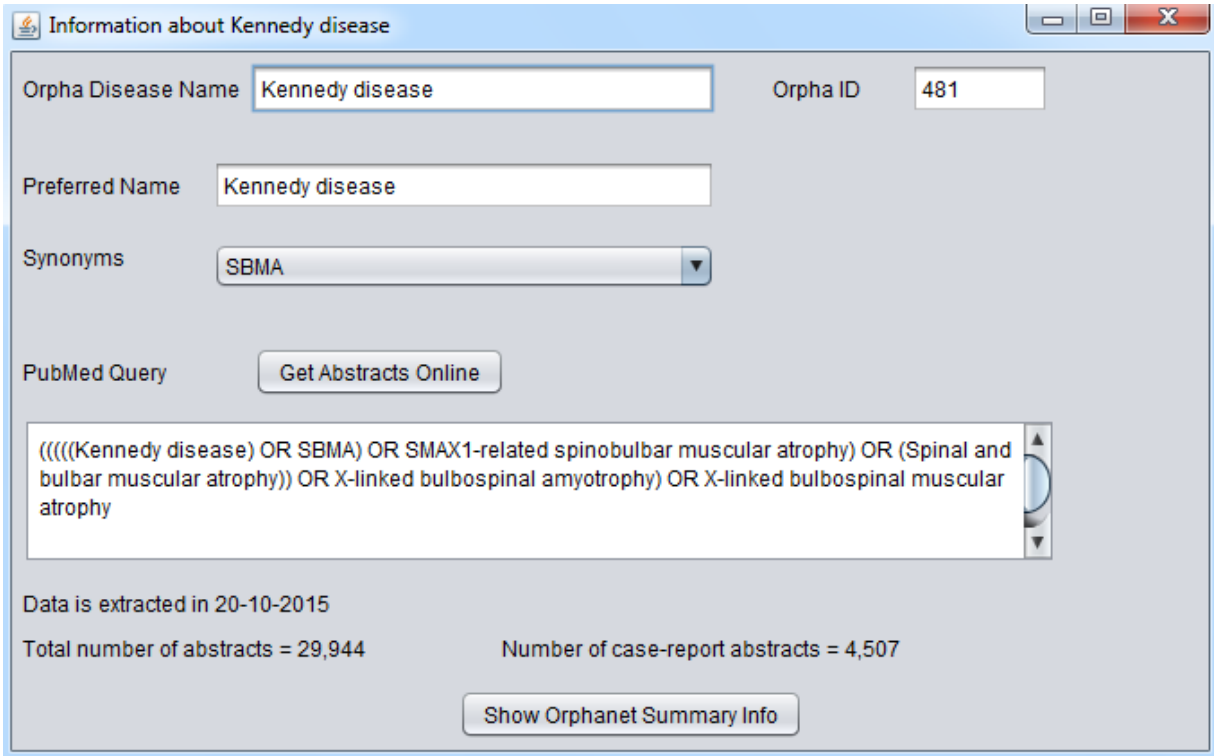


Figure C.9: This frame shows information related to Kennedy disease such as its orpha id, preferred name, a list of its synonyms, PubMed query ... etc.

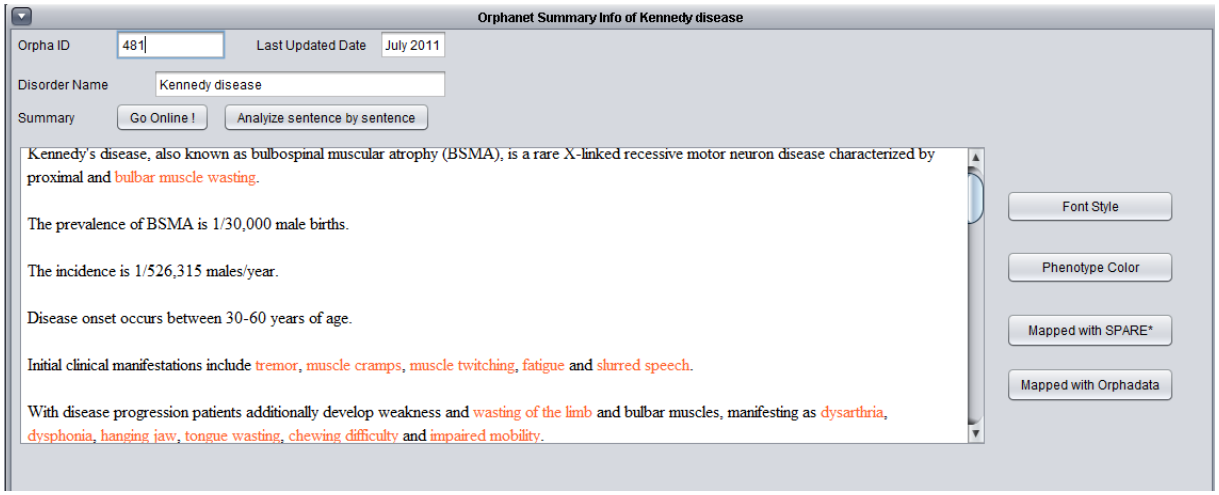


Figure C.10: This figure is a part of "Summary Info" frame that displays the Orphanet summary of Kennedy disease. Phenotypes annotated by the expert are in red color.

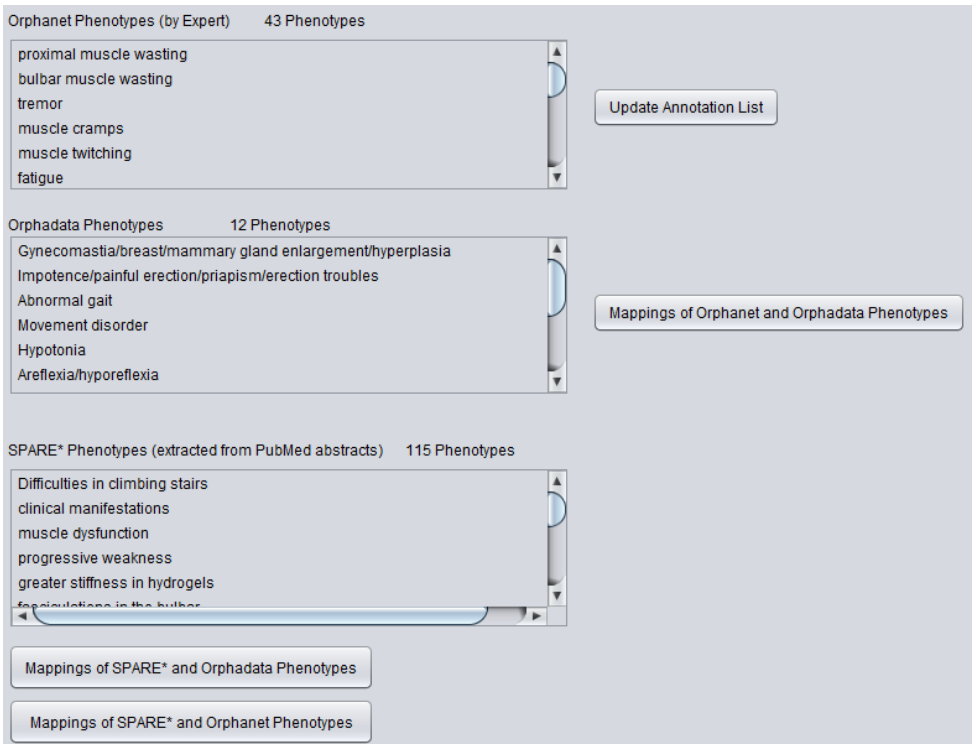


Figure C.11: This figure is the second part of "Summary Info" frame. It displays three lists of expert phenotypes, Orphadata phenotypes and SPARE⋆ phenotypes for Kennedy disease.

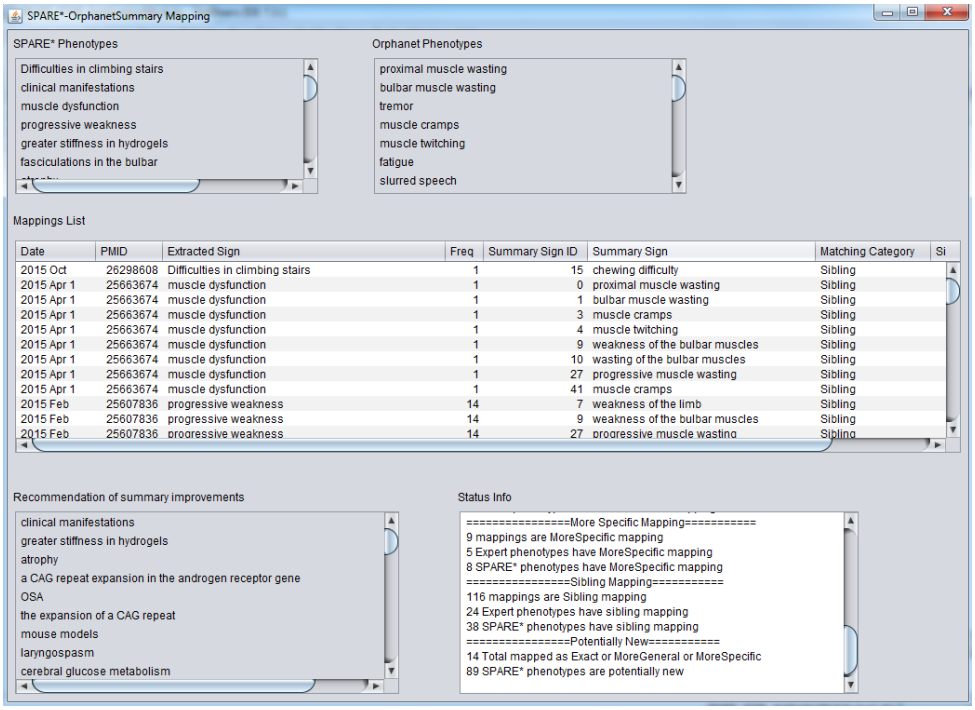


Figure C.12: This frame shows a table of mappings between SPARE⋆ and Orphanet phenotypes.

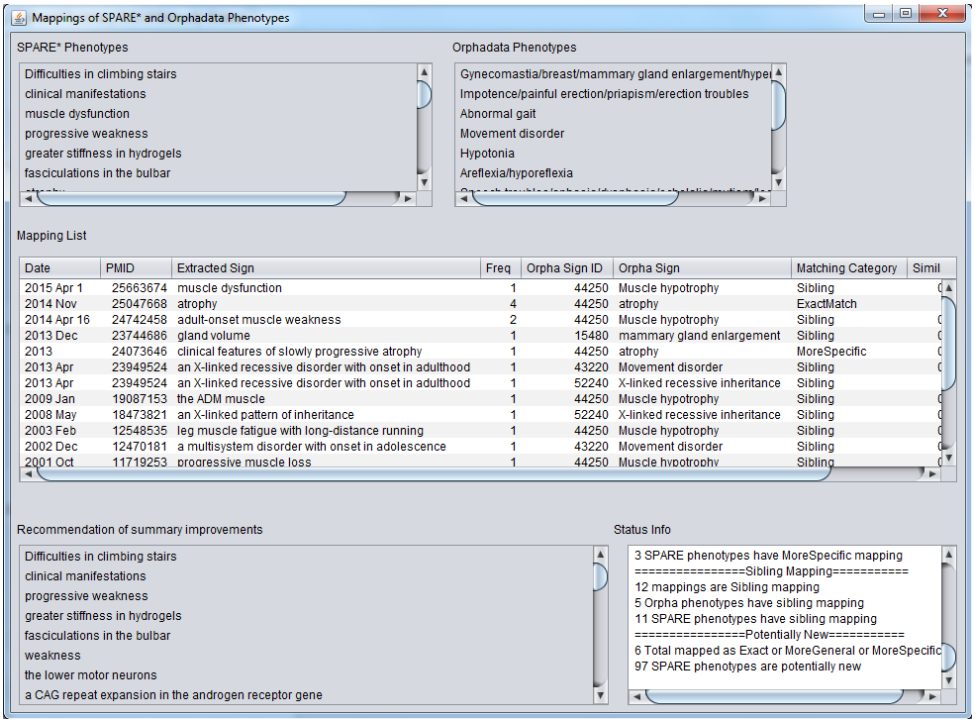


Figure C.13: This frame shows a table of mappings between SPARE⋆ and Orphadata phenotypes.

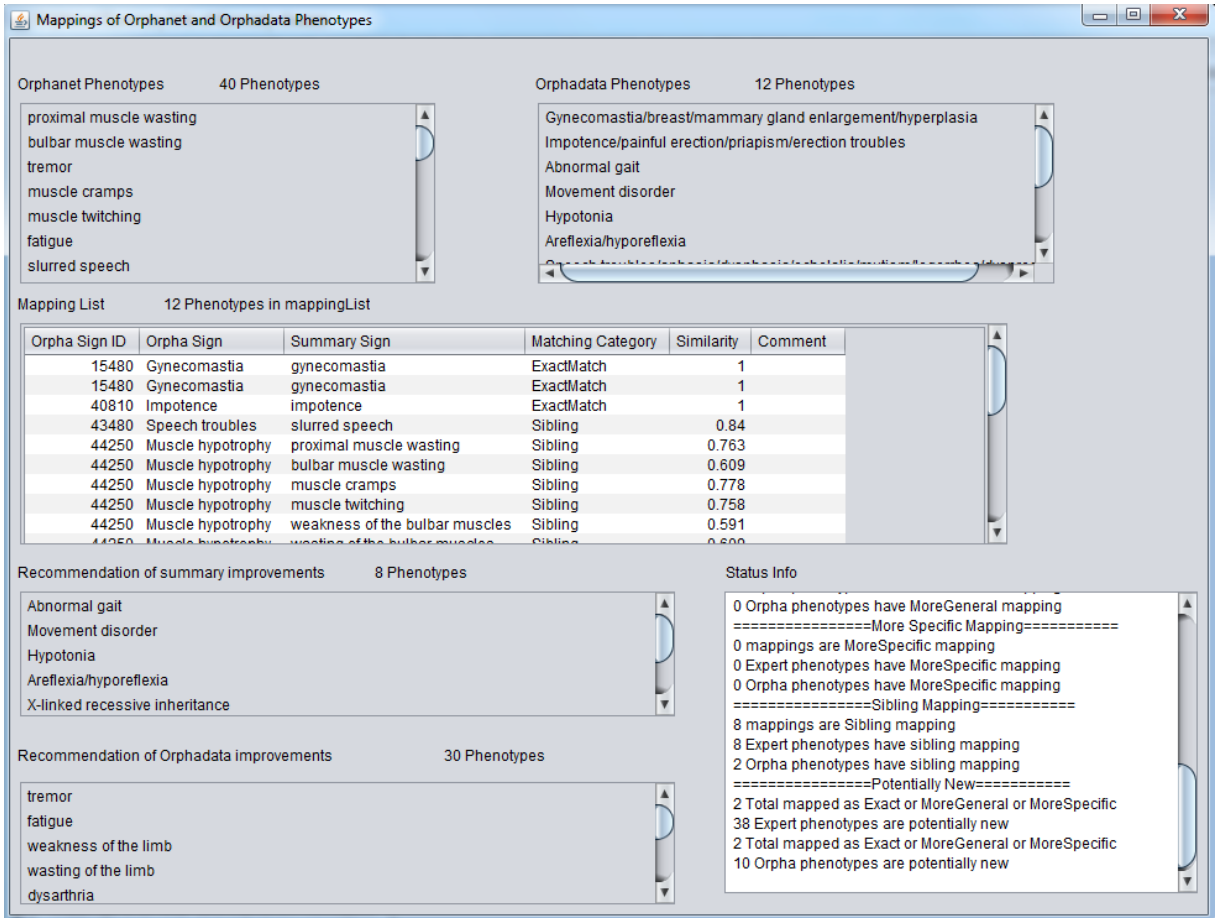


Figure C.14: This frame shows a table of mappings between Orphanet and Orphadata phenotypes.

C.2 Summary

In this appendix we described SPARE⋆ as a tool for information extraction. We presented the 5 modules of SPARE⋆. VAM is used to manually annotate a textual corpus with the interesting entities and their relationships. In addition, VAM could use annotations coming from automatic recognition tools (*e.g.,* MetaMap). Learning module learns the components of our SPARE⋆ method. It learns SPARE patterns from the DG dataset of the corpus and learns the SVM model classifier. Relation extraction module uses the configured SPARE⋆, the learned patterns and SVM model, to extract new relationships from texts where the entities of the relationships are previously recognized. Recognition module adopts SPARE⋆ to identify unrecognized entities that are in a relation with the other entities, *i.e.,* phenotypes that are in a relation with RDs. The application module is built upon the previous four modules. The development of this module depends on the application goal. In this thesis, we developed an application example for enriching the content of Orphanet summary and Orphadata.

SPARE⋆ modules can be used in a sequence, where the output of the first module is used as an input for the second module and so on. Also, they have been implemented as separated software components. Therefore, they can be used separately and independently from each other. In this case, each module should respect the input format of other modules.

Bibliography

[201] World Health Organization. 2011. International statistical classification of diseases and related health problems.

[ABB+12] Yasmine Asses, Aleksey Buzmakov, Thomas Bourquard, Sergei O. Kuznetsov, and Amedeo Napoli. A hybrid classification approach based on FCA and emerging patterns - an application for the classification of biological inhibitors. In *Proceedings of The Ninth International Conference on Concept Lattices and Their Applications, Fuengirola (Málaga), Spain, October 11-14, 2012*, pages 211–222, 2012.

[ABNS15] Mehwish Alam, Aleksey Buzmakov, Amedeo Napoli, and Alibek Sailanbayev. Revisiting pattern structures for structured attribute sets. In *Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015.*, pages 241–252, 2015.

[ABS+15] Joanna S. Amberger, Carol A. Bocchini, François Schiettecatte, Alan F. Scott, and Ada Hamosh. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Research*, 43(D1):D789–D798, January 2015.

[ADR17] Seyed Ziaeddin Alborzi, Marie-Dominique Devignes, and David W. Ritchie. Ecdomainminer: discovering hidden associations between enzyme commission numbers and pfam domains. *BMC Bioinformatics*, 18(1):107, 2017.

[AG00] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 85–94, New York, NY, USA, 2000. ACM.

[AHM01] S. Staab A. Hotho and A. Maedche. Ontology-based text clustering. In *In Proceedings of the IJCAI-2001 Workshop "Text Learning: Beyond Supervision*, 2001.

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[AKA91] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66, January 1991.

[AKBLN89] Hassan Aït-Kaci, Robert Boyer, Patrick Lincoln, and Roger Nasr. Efficient implementation of lattice operations. *ACM Trans. Program. Lang. Syst.*, 11(1):115–146, January 1989.

[AKM+03] Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, January 2003.

[AL10] Alan R. Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *JAMIA*, 17(3):229–236, 2010.

[AM02] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *In: Proceedings of the 1 st International Conference on General WordNet*, 2002.

[AMS+96] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Advances in knowledge discovery and data mining. chapter Fast Discovery of Association Rules, pages 307–328. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[AMT+09] Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, BioNLP '09, pages 185–192, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[APA08] Lourdes Araujo and José R. Pérez-Agüera. Improving query expansion with stemming terms: A new genetic algorithm approach. In *Proceedings of the 8th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'08, pages 182–193, Berlin, Heidelberg, 2008. Springer-Verlag.

[APB+08] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9 Suppl 11, 2008.

[Aro01] A. R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proc AMIA Symp*, pages 17–21, 2001.

[AS94a] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[AS94b] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[Ash00] M. Ashburner. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[ATA16] N. Alnazzawi, P. Thompson, and S. Ananiadou. Mapping phenotypic information in heterogeneous textual sources to a domain-specific terminological resource. *PLOS ONE*, 11(9):e0162287, 2016.

[ATBNA15] N. Alnazzawi, P. Thompson, R. Batista-Navarro, and S. Ananiadou. Using text mining techniques to extract phenotypic information from the phenochf corpus. *BMC Medical Informatics and Decision Making*, 15(Suppl. 2):S3, 2015.

[AXLU11] Peter Adolphs, Feiyu Xu, Hong Li, and Hans Uszkoreit. Dependency graphs as a generic interface between parsers and relation extraction rule learning. In *Proceedings of the 34th Annual German Conference on Advances in Artificial Intelligence*, KI'11, pages 50–62, Berlin, Heidelberg, 2011. Springer-Verlag.

[AZD15] Azdc corpus. Available on http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/corpora.html, oct Accessed October 2015.

[BBDR17] Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie, and Mathieu Roche. Xart: Discovery of correlated arguments of n-ary relations in text. *Expert Syst. Appl.*, 73:115–124, 2017.

[BBHN10] Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, pages 33–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[BCC+12] Nicolas Béchet, Peggy Cellier, Thierry Charnois, Bruno Crémilleux, and Marie-Christine Jaulent. Sequential pattern mining to discover relations between genes and rare diseases. In *CBMS*, pages 1–6, 2012.

[BDS+08] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, 9(1):207, 2008.

[BFANP14] Pedro Paulo Balage Filho, Lucas Vinicius Avanço, Maria das Graças Volpe Nunes, and Thiago Alexandre Salgueiro Pardo. NILC_USP: An improved hybrid system for sentiment analysis in twitter messages. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 428–432, Dublin, Ireland, 23–24 August 2014. Association for Computational Linguistics and Dublin City University.

[BFCP+05] Michael A. Bender, Martín Farach-Colton, Giridhar Pemmasani, Steven Skiena, and Pavel Sumazin. Lowest common ancestors in trees and directed acyclic graphs. *J. Algorithms*, 57(2):75–94, November 2005.

[BH01] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *IN WORKSHOP ON WORDNET AND OTHER LEXICAL RESOURCES, SECOND MEETING OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, 2001.

[BH06] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, March 2006.

[BHG+09] Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 10–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Bio15] Biotext corpus. Available on http://biocreative.sourceforge.net/bio_corpora_links.html, oct Accessed October 2015.

[BK99] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.*, 36(1-2):105–139, July 1999.

[BK05] Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society.

[BKN14] Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli. Scalable estimates of concept stability. In Cynthia Vera Glodeanu, Mehdi Kaytoue, and Christian Sacarea, editors, *Formal Concept Analysis - 12th International Conference, ICFCA 2014, Cluj-Napoca, Romania, June 10-13, 2014. Proceedings*, volume 8478 of *Lecture Notes in Computer Science*, pages 157–172. Springer, 2014.

[BL12] William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 546–556, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[BM05] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[BM06] Razvan Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In Y. Weiss, B. Schoelkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems, Vol. 18: Proceedings of the 2005 Conference (NIPS)*, 2006.

[BML+13] Meisam Booshehri, Abbas Malekpour, Peter Luksch, Kamran Zamanifar, and Shahdad Shariatmadari. Ontology enrichment by extracting hidden assertional knowledge from text. *CoRR*, abs/1308.0701, 2013.

[BMRM06] Razvan Bunescu, Raymond Mooney, Arun Ramani, and Edward Marcotte. Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline. In *Proceedings of the HLT-NAACL Workshop on Linking Natural Language Processing and Biology (BioNLP'06)*, pages 49–56, New York, NY, June 2006.

[BMUT97] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec.*, 26(2):255–264, June 1997.

[Bod04] O. Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270, 2004.

[Bre96a] Leo Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[Bre96b] Leo Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996.

[Bre01] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[BRO13] Boyan Bonev, Gema Ramírez-Sánchez, and Sergio Ortiz-Rojas. Statistical sentiment analysis performance in opinum. *CoRR*, abs/1303.0446, 2013.

[Bru11] Caroline Brun. Detecting opinions using deep syntactic analysis. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 392–398. RANLP 2011 Organising Committee, 2011.

[BS16] Mikhail Bogatyrev and Kirill Samodurov. Framework for conceptual modeling on natural language texts. In *Proceedings of the Third Workshop on Concept Discovery in Unstructured Data co-located with the 13th International Conference on Concept Lattices and Their Applications (CLA 2016), Moscow, Russia, July 18, 2016*, pages 13–24, 2016.

[BSJ15] Rakesh Chandra Balabantaray, Chandrali Sarma, and Monica Jha. Document clustering using k-means and k-medoids. *CoRR*, abs/1502.07938, 2015.

[BV02] Christian Blaschke and Alfonso Valencia. Automatic ontology construction from the literature. *Genome Informatics*, 13:201–213, 2002.

[CAM04] Michelangelo Ceci, Annalisa Appice, and Donato Malerba. Spatial associative classification at different levels of granularity: A probabilistic approach. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, pages 99–111, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[CC05] Ben Carterette and Fazli Can. Comparing inverted files and signature files for searching a large lexicon. *Inf. Process. Manage.*, 41(3):613–633, May 2005.

[CCP10] Peggy Cellier, Thierry Charnois, and Marc Plantevit. Sequential patterns to discover and characterise biological relations. In A. F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing (CICLing)*, LNCS 6008, pages 537–548. Springer, 2010.

[CDKN13] Adrien Coulet, Florent Domenach, Mehdi Kaytoue, and Amedeo Napoli. Using pattern structures for analyzing ontology-based annotations of biomedical data. In *Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings*, pages 76–91, 2013.

[CHD+07] M. Croitoru, B. Hu, S. Dasmahapatra, P. Lewis, D. Dupplaw, A. Gibb, M. Julia-Sape, J. Vicente, C. Saez, J. M. Garcia-Gomez, R. Roset, F. Estanyol, X. Rafael, and M. Mier. Conceptual graphs based information retrieval in healthagents. In *Twentieth IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, pages 618–623, June 2007.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[CL12] Md. Faisal Mahbub Chowdhury and Alberto Lavelli. Combining tree structures, flat features and patterns for biomedical relation extraction. In *EACL*, pages 420–429, 2012.

[CLJ+14] Liang Cheng, Jie Li, Peng Ju, Jiajie Peng, and Yadong Wang. Semfunsim: A new method for measuring disease similarity by integrating semantic and gene functional association. *PLoS ONE*, 9(6):1–11, 06 2014.

[CLM11] Faisal Md. Chowdhury, Alberto Lavelli, and Alessandro Moschitti. A study on dependency tree kernels for automatic extraction of protein-protein interaction. In *Proceedings of BioNLP 2011 Workshop*, pages 124–133, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[CLN14] Víctor Codocedo, Ioanna Lykourentzou, and Amedeo Napoli. A semantic approach to concept lattice-based information retrieval. *Ann. Math. Artif. Intell.*, 72(1-2):169–195, 2014.

[CMS09] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.

[CO01] John M. Conroy and Dianne P. O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 406–407, New York, NY, USA, 2001. ACM.

[COG15] Nigel Collier, Anika Oellrich, and Tudor Groza. Concept selection for phenotypes and diseases using learn to rank. *J. Biomedical Semantics*, 6:24, 2015.

[CR96] Claudio Carpineto and Giovanni Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2):95–122, 1996.

[CR04] Claudio Carpineto and Giovanni Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.

[CS04] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[CSG+10] Adrien Coulet, Nigam H. Shah, Yael Garten, Mark A. Musen, and Russ B. Altman. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics*, 43(6):1009–1019, 2010.

[CSL+13] Erik Cambria, Bjorn Schuller, Bing Liu, Haixun Wang, and Catherine Havasi. Knowledge-based approaches to concept-level sentiment analysis. *IEEE Intelligent Systems*, 28(2):12–14, March 2013.

[CTL+13] Nigel Collier, Mai-vu Tran, Hoang-quynh Le, Quang-Thuy Ha, Anika Oellrich, and Dietrich Rebholz-Schuhmann. Learning to Recognize Phenotype Candidates in the Auto-Immune Literature Using SVM Re-Ranking. *PLoS ONE*, 8(10):e72965+, October 2013.

[CWB+11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[CWS+16] Liang Cheng, Zhenzhen Wang, Hongbo Shi, Jie Sun, Haixiu Yang, Shuo Zhang, Yang Hu, and Meng Zhou. Dissim: an online system for exploring significant similar diseases and exhibiting potential therapeutic drugs. *Scientific Reports*, 2016.

[CYZH05] Jian Chen, Jian Yin, Jun Zhang, and Jin Huang. *Associative Classification in Text Categorization*, pages 1035–1044. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[DA05] A. Divoli and T. K. Attwood. BioIE: extracting informative sentences from the biomedical literature. *Bioinformatics*, 21(9):2138–9, 2005.

[DBK+97] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. In M. I. Jordan and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press, 1997.

[DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[DH07] Jie Dong and Min Han. Bittablefi: An efficient mining frequent itemsets algorithm. *Knowledge-Based Systems*, 20(4):329 – 335, 2007.

[Die00] Thomas G. Dietterich. Ensemble methods in machine learning. In *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857*, pages 1–15. Springer, 2000.

[DL12] Rezarta Islamaj Doğan and Zhiyong Lu. An improved corpus of disease mentions in pubmed citations. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, pages 91–99, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[DLL14] Rezarta Islamaj DoÄŸan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, (0):–, 2014.

[DMS06] Menno Van Zaanen Diego Mollá and Daniel Smith. Named entity recognition for question answering. In *In Lawrence Cavedon and Ingrid Zukerman, editors, Proceedings of the 2006 Australasian Language Technology Workshop*, pages 51–58, 2006.

[EJR+13] Elias Egho, Nicolas Jay, Chedy Raïssi, Gilles Nuemi, Catherine Quantin, and Amedeo Napoli. An approach for mining care trajectories for chronic diseases. In *Artificial Intelligence in Medicine - 14th Conference on Artificial Intelligence in Medicine, AIME 2013, Murcia, Spain, May 29 - June 1, 2013. Proceedings*, pages 258–267, 2013.

[Epp99] David Eppstein. Subgraph isomorphism in planar graphs and related problems. *CoRR*, cs.DS/9911003, 1999.

[ERI+12] Elias Egho, Chedy Raïssi, Dino Ienco, Nicolas Jay, Amedeo Napoli, Pascal Poncelet, Catherine Quantin, and Maguelonne Teisseire. Healthcare trajectory mining by combining multidimensional component and itemsets. In Annalisa Appice, Michelangelo Ceci, Corrado Loglisci, Giuseppe Manco, Elio Masciari, and Zbigniew W. Ras, editors, *New Frontiers in Mining Complex Patterns - First International Workshop, NFMCP 2012, Held in Conjunction with ECML/PKDD 2012, Bristol, UK, September 24, 2012, Revised Selected Papers*, volume 7765 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2012.

[ES13] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.

[FAR07] Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 57–60, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[FBS02] Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato. The FrameNet database and software tools. In *Proceedings of the Third International Conference on Languag Resources and Evaluation*, volume IV, Las Palmas, 2002. LREC, LREC.

[FCT04] Pierre Héroux Fabien Carmagnac and Eric Trupin. Distance based strategy for supervised document image classification. In *International Workshops on Statistical Pattern Recognition*, 2004.

[Fel98] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[FG13] Hai Fang and Julian Gough. dcgo: database of domain-centric ontologies on functions, phenotypes, diseases and more. *Nucleic Acids Research*, 41(D1):D536, 2013.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3):131–163, November 1997.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Fil76] Charles J. Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280(1):20–32, 1976.

[FKY+01] Carol Friedman, Pauline Kra, Hong Yu, Michael Krauthammer, and Andrey Rzhetsky. Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. *Comput. Appl. Biosci.*, 17(suppl_1):S74–82, June 2001.

[FKZ07] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, January 2007.

[FM06] Dmitriy Fradkin and Ilya Muchnik. Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:13–20, 2006.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Advances in knowledge discovery and data mining. chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[FS02] Andreas Faatz and Ralf Steinmetz. Ontology enrichment with texts from the www. In *In Semantic Web Mining, WS02*, 2002.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[GFW03] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *IN: CONFERENCE ON LEARNING THEORY*, pages 129–143, 2003.

[GG12] Michael R. Glass and Alfio Massimiliano Gliozzo. Structured term recognition in medical text. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 943–958, 2012.

[GGKS04] Bernhard Ganter, Peter A. Grigoriev, Sergei O. Kuznetsov, and Mikhail V. Samokhin. Concept-based data mining with scaled labeled graphs. In Karl Erich Wolff, Heather D. Pfeiffer, and Harry S. Delugach, editors, *Proceedings of the 12th International Conference on Conceptual Structures (ICCS 2004)*, volume 3127 of *Lecture Notes in Computer Science*, pages 94–108. Springer, 2004.

[GGNAM16] María del Mar Roldán García, José García-Nieto, and José F. Aldana-Montes. An ontology-based data integration approach for web analytics in e-commerce. *Expert Syst. Appl.*, 63(C):20–34, November 2016.

[GJ02] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288, September 2002.

[GK01] Bernhard Ganter and Sergei Kuznetsov. Pattern structures and their projections. In Harry Delugach and Gerd Stumme, editors, *Conceptual Structures: Broadening the Base*, volume 2120 of *Lecture Notes in Computer Science*, pages 129–142. Springer, Berlin/Heidelberg, 2001.

[GLR06] Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 3–7, Trento, Italy, April 2006. The Association for Computer Linguistics.

[Gra90] Robert M. Gray. *Entropy and Information Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[GW89] B. Ganter and R. Wille. Conceptual scaling. In F. Roberts, editor, *Applications of combinatorics and graph theory to the biological and social sciences*, pages 139–167. Springer-Verlag, 1989.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin – Heidelberg, 1999.

[GXCL09] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM.

[GZ01] Karam Gouda and Mohammed Javeed Zaki. Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society.

[Hal99] Mark A. Hall. Correlation-based feature selection for machine learning. Technical report, 1999.

[Har54] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[Has03] Martin Hassel. Exploitation of named entities in automatic text summarization for swedish. In *In Proceedings of NODALIDA 03 - 14 th Nordic Conference on Computational Linguistics, May 30-31 2003*, 2003.

[HCT14] Mohsen Hassan, Adrien Coulet, and Yannick Toussaint. Learning subgraph patterns from text for extracting disease - symptom relationships. In *Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, DMNLP@PKDD/ECML 2014, Nancy, France, September 15, 2014.*, pages 81–96, 2014.

[Hol93] Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.*, 11(1):63–90, April 1993.

[HPL+05] Jörg Hakenberg, Conrad Plake, Ulf Leser, Harald Kirsch, and Dietrich Rebholz-schuhmann. Lll'05 challenge: genic interaction extraction – identification . . . with alignments and finite state automata. In *IN PROC LEARNING LANGUAGE IN LOGIC WORKSHOP (LLL05) AT THE 22ND INT CONF ON MACHINE LEARNING*, pages 38–45, 2005.

[HPMA+00] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Freespan: Frequent pattern-projected sequential pattern mining. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 355–359, New York, NY, USA, 2000. ACM.

[HPO15] Hpo (the human phenotype ontology). Available on http://human-phenotype-ontology.github.io/downloads.html, oct Accessed October 2015.

[HSG15] Robert Hoehndorf, Paul N. Schofield, and Georgios V. Gkoutos. Analysis of the human diseasome using phenotype similarity between common, genetic, and infectious diseases. *Scientific Reports*, 5, 2015.

[HV06] Ian Horrocks and Andrei Voronkov. *Reasoning support for expressive ontology languages using a theorem prover*, volume 3861 of *Lecture Notes in Computer Science*, pages 201–218. Springer Verlag, Germany, 2006.

[HWP03] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 549–, Washington, DC, USA, 2003. IEEE Computer Society.

[HZGH08] Guobiao Hu, Shuigeng Zhou, Jihong Guan, and Xiaohua Hu. Towards effective document clustering: A constrained k-means based approach. *Inf. Process. Manage.*, 44(4):1397–1409, July 2008.

[HZL06] Minlie Huang, Xiaoyan Zhu, and Ming Li. A hybrid method for relation extraction from biomedical literature. *I. J. Medical Informatics*, 75(6):443–455, 2006.

[ICD07] International Classification of Diseases (ICD), 2007.

[IWM00] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 13–23, London, UK, UK, 2000. Springer-Verlag.

[JL95] George John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995.

[Joa98] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[KAF+15] Warren A. Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J. Mungall, Janos X. Binder, James Malone, Drashtti Vasant, Helen E. Parkinson, and Lynn M. Schriml. Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic Acids Research*, 43(Database-Issue):1071–1078, 2015.

[Kat16] Rohit J. Kate. Normalizing clinical terms using learned edit distance patterns. *JAMIA*, 23(2):380–386, 2016.

[KDKN09] Mehdi Kaytoue, Sébastien Duplessis, Sergei O. Kuznetsov, and Amedeo Napoli. *Two FCA-Based Methods for Mining Gene Expression Data*, pages 251–266. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[KDM+14] Sebastian Köhler, Sandra C. Doelken, Christopher J. Mungall, Sebastian Bauer, Helen V. Firth, Isabelle Bailleul-Forestier, Graeme C. Black, Danielle L. Brown, Michael Brudno, Jennifer Campbell, David R. FitzPatrick, Janan T. Eppig, Andrew P. Jackson, Kathleen Freson, Marta Girdea, Ingo Helbig, Jane A. Hurst, Johanna Jähn, Laird G. Jackson, Anne M. Kelly, David H. Ledbetter, Sahar Mansour, Christa L. Martin, Celia Moss, Andrew Mumford, Willem H. Ouwehand, Soo-Mi M. Park, Erin Rooney R. Riggs, Richard H. Scott, Sanjay Sisodiya, Steven Van Vooren, Ronald J. Wapner, Andrew O. Wilkie, Caroline F. Wright, Anneke T. Vulto-van Silfhout, Nicole de Leeuw, Bert B. de Vries, Nicole L. Washingthon, Cynthia L. Smith, Monte Westerfield, Paul Schofield, Barbara J. Ruef, Georgios V. Gkoutos, Melissa Haendel, Damian Smedley, Suzanna E. Lewis, and Peter N. Robinson. The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data. *Nucleic acids research*, 42(Database issue):D966–D974, January 2014.

[KK01] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 313–320, Washington, DC, USA, 2001. IEEE Computer Society.

[KK12a] Felix Eichinger Keller, Benjamin J. and Matthias Kretzler. Formal concept analysis of disease similarity. *AMIA Summits on Translational Science Proceedings.*, page 42–51, 2012.

[KK12b] Felix Eichinger Keller, Benjamin J. and Matthias Kretzler. Formal concept analysis of disease similarity. *AMIA Summits on Translational Science Proceedings.*, page 42–51, 2012.

[KKND11] Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sébastien Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989 – 2001, 2011. Special Issue on Information Engineering Applications Based on Lattices.

[KL02a] Latifur Khan and Feng Luo. Ontology construction for information selection. In *14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002), 4-6 November 2002, Washington, DC, USA*, page 122, 2002.

[KL02b] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[KLR+04] Krishna Kummamuru, Rohit Lotlikar, Shourya Roy, Karan Singal, and Raghu Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 658–665, New York, NY, USA, 2004. ACM.

[KLRPV08] Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome biology*, 9 Suppl 2(Suppl 2):S4+, 2008.

[KLYW15] Sun Kim, Haibin Liu, Lana Yeganova, and W. John Wilbur. Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. *Journal of Biomedical Informatics*, 55:23 – 30, 2015.

[KMR11] Maryam Khordad, Robert E. Mercer, and Peter Rogan. *Improving Phenotype Name Recognition*, pages 246–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[KOP+09] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 1–9, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[KPC95] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 68–73, New York, NY, USA, 1995. ACM.

[KPM+08] Izeta Kurbasic, Haris Pandza, Izet Masic, Senad Huseinagic, Salih Tandir, Fredi Alicajic, and Selim Toromanovic. The advantages and limitations of international classification of diseases, injuries and causes of death from aspect of existing health care system of bosnia and herzegovina. *Acta Inform Med*, 2008.

[KS05] Sergei O. Kuznetsov and Mikhail V. Samokhin. Learning closed sets of labeled graphs for chemical applications. In Stefan Kramer and Bernhard Pfahringer, editors, *ILP*, volume 3625 of *Lecture Notes in Computer Science*, pages 190–208. Springer, 2005.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[Kuz90] S. O. Kuznetsov. Stability as an estimate of the degree of substantiation of hypotheses on the basis of operational similarity. *Automatic Documentation and Mathematical Linguistics (Nauch. Tekh. Inf. Ser. 2)*, pages 62–75, 1990.

[Kuz13] Sergei O. Kuznetsov. Fitting pattern structures to knowledge discovery in big data. In *Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings*, pages 254–266, 2013.

[KW11] Lukasz Kobylinski and Krzysztof Walczak. Efficient mining of jumping emerging patterns with occurrence counts for classification. *Trans. Rough Sets*, 13:73–88, 2011.

[KYY08] Seonho Kim, Juntae Yoon, and Jihoon Yang. Kernel approaches for genic interaction extraction. *Bioinformatics*, 24(1):118–126, 2008.

[LAR02] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Min. Knowl. Discov.*, 6(1):83–105, January 2002.

[LBTN15] Artuur Leeuwenberg, Aleksey Buzmakov, Yannick Toussaint, and Amedeo Napoli. *Exploring Pattern Structures of Syntactic Trees for Relation Extraction*, pages 153–168. Springer International Publishing, Cham, 2015.

[LGC+11] Jiang Li, Binsheng Gong, Xi Chen, Tao Liu, Chao Wu, Fan Zhang, Chunquan Li, Xiang Li, Shaoqi Rao, and Xia Li. Dosim: An r package for similarity between diseases based on disease ontology. *BMC Bioinformatics*, 12(1):266, 2011.

[LGYW16] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. *Int J Hybrid Inf Technol*, 9:237–248, 2016.

[LHM98] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 80–86. AAAI Press, 1998.

[Li03] Xiaoyan Li. Syntactic features in question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 383–384, New York, NY, USA, 2003. ACM.

[Lin98] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, pages 768–774, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[Lin99] Chin-Yew Lin. Training a selection function for extraction. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, pages 55–62, New York, NY, USA, 1999. ACM.

[Lin15] Alias-i. 2008. lingpipe 4.1.0. http://alias-i.com/lingpipe, oct Accessed October 2015.

[Liu10] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*, 2010.

[LJ16] Jiewu Leng and Pingyu Jiang. A deep learning approach for relationship extraction from interaction context in social manufacturing paradigm. *Knowledge-Based Systems*, 100:188 – 199, 2016.

[LMC15] A. Leal, B. Martins, and Francisco Couto. Ulisboa: Recognition and normalization of medical concepts. In *9th International Workshop on Semantic Evaluation (SemEval)*, 2015.

[LTCW16] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Comp. Math. Methods in Medicine*, 2016:6918381:1–6918381:8, 2016.

[LV09] Wenhui Liao and Sriharsha Veeramachaneni. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, pages 58–65, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[LVC+13] Haibin Liu, Karin Verspoor, Donald C. Comeau, Andrew MacKinlay, and W John Wilbur. Generalizing an approximate subgraph matching-based system to extract events in molecular biology and cancer genetics. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 76–85, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[LvI10] Peder Olesen Larsen and Markus von Ins. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, pages 575–603, 2010.

[LW07] Jimmy Lin and W. John Wilbur. Pubmed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics*, 8(1):423, 2007.

[LYSZ16] Yonghong Luo, Zhifan Yang, Huike Shi, and Ying Zhang. *A Distributed Frequent Itemsets Mining Algorithm Using Sparse Boolean Matrix on Spark*, pages 419–423. Springer International Publishing, Cham, 2016.

[MBC14] Laure Martin, Delphine Battistelli, and Thierry Charnois. Symptom extraction issue. In *Proceedings of BioNLP 2014*, pages 107–111, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[Mcc06] Sabine Mcconnell. *Distributed Predictive and Descriptive Data Mining*. PhD thesis, Kingston, Ont., Canada, Canada, 2006. AAINR18534.

[MCS06] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006.

[MD12] Sachin Mathur and Deendayal Dinakarpandian. Finding disease similarity based on implicit semantic similarity. *J. of Biomedical Informatics*, 45(2):363–371, April 2012.

[MD14] Marghny H. Mohamed and Mohammed M. Darwieesh. Efficient mining frequent itemsets algorithms. *International Journal of Machine Learning and Cybernetics*, 5(6):823–833, 2014.

[MED17] Publication added to medline by year. https://www.nlm.nih.gov/bsd/stats/cit_added.html, feb Accessed February 2017.

[MKR+16] Christopher J Mungall, Sebastian Koehler, Peter Robinson, Ian Holmes, and Melissa Haendel. k-boom: A bayesian approach to ontology structure inference, with applications in disease ontology construction. *bioRxiv*, 2016.

[MKT+15] Sebastian Mate, Felix Köpcke, Dennis Toddenroth, Marcus Martin, Hans-Ulrich Prokosch, Thomas Bürkle, and Thomas Ganslandt. Ontology-based data integration between clinical and research systems. *PLOS ONE*, 10(1):1–20, 01 2015.

[ML02] José M. Martínez and Erwan Loisant. Browsing image databases with galois' lattices. In Gary B. Lamont, Hisham Haddad, George A. Papadopoulos, and Brajendra Panda, editors, *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC), March 10-14, 2002, Madrid, Spain*, pages 791–795. ACM, 2002.

[ML08] Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *In Proceedings of ACL-08: HLT*, pages 236–244, 2008.

[ML09] Jeff Mitchell and Mirella Lapata. Language models based on semantic composition. In *EMNLP*, pages 430–439. ACL, 2009.

[MPHT10] Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, BioNLP '10, pages 37–45, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[MPK+05] Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 491–498, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[MPRZ99] M. Munoz, V. Punyakanok, D. Roth, and D. Zimak. A learning approach to shallow parsing. In *EMNLP-VLC*, pages 168–178, 6 1999.

[MR01] Scott Mcdonald and Michael Ramscar. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *In Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 611–6, 2001.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[MS00] Alexander Maedche and Steffen Staab. Discovering conceptual relations from text. In *Proc. of ECAI'2000*, pages 321–325, 2000.

[MSB+14] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[MSMT09] Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 121–130, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[MTM+12] M. Miwa, P. Thompson, J. McNaught, D. B. Kell, and S. Ananiadou. Extracting semantically enriched events from biomedical literature. *BMC Bioinformatics*, 13:108, 2012. Highly Accessed.

[MTV94] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Efficient algorithms for discovering association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pages 181–192. AAAI Press, 1994.

[MVGaN13] Rodrigo Moraes, JoãO Francisco Valiati, and Wilson P. GaviãO Neto. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Syst. Appl.*, 40(2):621–633, February 2013.

[NCB15] The ncbi disease corpus. Available on http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/, oct Accessed October 2015.

[NCMO13] Tiago Nunes, David Campos, Sérgio Matos, and José Luís Oliveira. Becas: biomedical concept recognition services and visualization. *Bioinformatics*, 29(15):1915–1916, 2013.

[Nea03] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[NG15] Thien Huu Nguyen and Ralph Grishman. *Relation extraction: Perspective from convolutional neural networks*. 2015.

[NK05] Siegfried Nijssen and Joost N. Kok. The gaston tool for frequent subgraph mining. *Electr. Notes Theor. Comput. Sci.*, 127(1):77–87, 2005.

[NKT11] N. T. H. Nguyen, J.-D. Kim, and J. Tsujii. Overview of bionlp 2011 protein coreference shared task. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 74–82, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[NVHT12] Loan T. T. Nguyen, Bay Vo, Tzung-Pei Hong, and Hoang Chi Thanh. Classification based on association rules: A lattice-based approach. *Expert Syst. Appl.*, 39(13):11357–11366, October 2012.

[OMI15] Online mendelian inheritance in man, omim®. mckusick-nathans institute of genetic medicine, johns hopkins university (baltimore, md). Available on http://www.omim.org/, oct Accessed October 2015.

[OPG13] Alina Onishchenko Olga Prokasheva and Sergey Gurov. Classification methods based on formal concept analysis. *Proceedings of the Workshop Formal Concept Analysis Meets Information Retrieval (FCAIR 2013).*, pages 95–104, 2013.

[ORD15] Orphanet rare disease ontology. http://www.bioportal.bioontology.org/ontologies/ORDO, oct Accessed October 2015.

[Orp15a] Orphadata: Free access data from orphanet. © INSERM 1997. Available on http://www.orphadata.org, oct Accessed October 2015.

[Orp15b] Orphanet: an online rare disease and orphan drug data base. © INSERM 1997. Available on http://www.orpha.net, oct Accessed October 2015.

[OSG10] Sonia Ordoñez-Salinas and Alexander Gelbukh. *Information Retrieval with a Simplified Conceptual Graph-Like Representation*, pages 92–104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[OTO15] Mai Omura, Yuka Tateishi, and Takashi Okumura. Disease similarity calculation on simplified disease knowledge base for clinical decision support systems. In Ingrid Russell and William Eberle, editors, *FLAIRS Conference*, pages 501–506. AAAI Press, 2015.

[PCY95] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash-based algorithm for mining association rules. *SIGMOD Rec.*, 24(2):175–186, May 1995.

[PFF+09] Catia Pesquita, Daniel Faria, André O. Falcão, Phillip Lord, and Francisco M. Couto. Semantic similarity in biomedical ontologies. *PLoS Comput Biol*, 5(7):1–12, 07 2009.

[Phe15] Phenochf corpus. Available on http://www.nactem.ac.uk/PhenoCHF/, oct Accessed October 2015.

[PHMA+04] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16(11):1424–1440, November 2004.

[PL07] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199, June 2007.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[PUB15] Pubmed. Available on https://www.ncbi.nlm.nih.gov/pubmed, oct Accessed October 2015.

[PY13] Jakub Piskorski and Roman Yangarber. Information extraction: Past, present and future. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 23–49. Springer Berlin Heidelberg, 2013.

[QGYH14] Hongjian Qiu, Rong Gu, Chunfeng Yuan, and Yihua Huang. Yafim: A parallel frequent itemset mining algorithm with spark. In *IPDPS Workshops*, pages 1664–1671. IEEE Computer Society, 2014.

[QMR+16] Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100:55 – 61, 2016.

[Qui86] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[Qui93] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RBMM05] A.K. Ramani, R.C. Bunescu, Raymond J. Mooney, and E.M. Marcotte. Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biology*, 6(5):r40, 2005.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[RH04] Barbara Rosario and Marti A. Hearst. Classifying semantic relations in bioscience texts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[RHM02] Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4):399–408, December 2002.

[RKK15] Sanjay Rathee, Manohar Kaul, and Arti Kashyap. R-apriori: An efficient apriori based algorithm on spark. In *Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management*, PIKM '15, pages 27–34, New York, NY, USA, 2015. ACM.

[RP08] Chedy Raïssi and Marc Plantevit. *Mining Multidimensional Sequential Patterns over Data Streams*, pages 263–272. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[RRNS14] Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. Semeval-2014 task 9: Sentiment analysis in twitter. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 73–80. The Association for Computer Linguistics, 2014.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.

[SA00] Guus T. Schreiber and Hans Akkermans. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, Cambridge, MA, USA, 2000.

[SAN+12] Lynn M. Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei W. Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren A. Kibbe. Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, 40(D1):D940–D946, January 2012.

[SBJ+09] Nigam H. Shah, Nipun Bhatia, Clement Jonquet, Daniel Rubin, Annie P. Chiang, and Mark A. Musen. Comparison of concept recognizers for building the Open Biomedical Annotator. *BMC bioinformatics*, 10 Suppl 9, 2009.

[SBMdPS11] Isabel Segura-Bedmar, Paloma Martínez, and César de Pablo-Sánchez. Using a shallow linguistic kernel for drug-drug interaction extraction. *Journal of Biomedical Informatics*, 44(5):789–804, 2011.

[Sch97] Bruce Schatz. Information retrieval in digital libraries: Bringing search to the net. *Science*, 275:327–333, 1997.

[Sch01] R. E. Schapire. The Boosting Approach to Machine Learning: An Overview. In MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA, 2001.

[SDC+10] Silpa Suthram, Joel T. Dudley, Annie P. Chiang, Rong Chen, Trevor J. Hastie, and Atul J. Butte. Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets. *PLoS Comput Biol*, 6(2):1–10, 02 2010.

[SDG15] Jayakrushna Sahoo, Ashok Kumar Das, and A. Goswami. An efficient approach for mining association rules from high utility itemsets. *Expert Syst. Appl.*, 42(13):5754–5778, August 2015.

[Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[Set04] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA '04, pages 104–107, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[Set05] B. Settles. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[SF16] Tony C. Smith and Eibe Frank. *Statistical Genomics: Methods and Protocols*, chapter Introducing Machine Learning Concepts with WEKA, pages 353–378. Springer, New York, NY, 2016.

[SHK+14] Sung Jeon Song, Go Eun Heo, Ha Jin Kim, Hyo Jung Jung, Yong Hwan Kim, and Min Song. Grounded feature selection for biomedical relation extraction by the combinative approach. In *Proceedings of the ACM 8th International Workshop on Data and Text Mining in Bioinformatics*, DTMBIO '14, pages 29–32, New York, NY, USA, 2014. ACM.

[SHMN12] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[SIL07] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[SK03] A. J. Smola and I. R. Kondor. Kernels and regularization on graphs. In *Proceedings of the Annual Conference on Computational Learning Theory*, 2003.

[SK11] Björn Schuller and Tobias Knaup. Learning and knowledge-based sentiment analysis in movie review key excerpts. In *Proceedings of the Third COST 2102 International Training School Conference on Toward Autonomous, Adaptive, and Context-aware Multimodal Interfaces: Theoretical and Practical Issues*, pages 448–472, Berlin, Heidelberg, 2011. Springer-Verlag.

[SKSS08] Dimitrios Skoutas, Verena Kantere, Alkis Simitsis, and Timos Sellis. *Ontology-Based Data Sharing in P2P Databases*, pages 117–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[SMO+10] G.K. Savova, J.J. Masanz, P.V. Ogren, J. Zheng, S. Sohn, K.C. Kipper-Schuler, and C.G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[SNV07] Laszlo Szathmary, Amedeo Napoli, and Petko Valtchev. Towards rare itemset mining. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), October 29-31, 2007, Patras, Greece, Volume 1*, pages 305–312. IEEE Computer Society, 2007.

[Sow84] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[SPH+11] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[SS10] Kumutha Swampillai and Mark Stevenson. Inter-sentential relations in information extraction corpora. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *LREC*. European Language Resources Association, 2010.

[SS11] Kumutha Swampillai and Mark Stevenson. Extracting relations within and across sentences. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 25–32. RANLP 2011 Organising Committee, 2011.

[SSM11] Jiang Su, Jelber Sayyad Shirab, and Stan Matwin. Large scale text classification using semisupervised multinomial naive bayes. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 97–104. Omnipress, 2011.

[Ste06] Mark Stevenson. Fact distribution in information extraction. *Language Resources and Evaluation*, 40(2):183–201, 2006.

[STG15] Balaji Jagan S. Thenmalar and T. V. Geetha. Semi-supervised bootstrapping approach for named entity recognition. *CoRR*, abs/1511.06833, 2015.

[STML09] Joseph Sill, Gábor Takács, Lester W. Mackey, and David Lin. Feature-weighted linear stacking. *CoRR*, abs/0911.0460, 2009.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[TBT+11] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June 2011.

[THE00] Keng-Woei Tan, Hyoil Han, and Ramez Elmasri. Web data cleansing and preparation for ontology extraction using wordnet. In *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)-Volume 2 - Volume 2*, WISE '00, pages 2011–, Washington, DC, USA, 2000. IEEE Computer Society.

[TL10] Lei Tang and Huan Liu. Graph mining applications to social network analysis. In Charu C. Aggarwal and Haixun Wang, editors, *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*, pages 487–513. Springer, 2010.

[TMK05] Pang-Ning Tan, Steinbach Michael, and Vipin Kumar. *Association Analysis: Basic Concepts and Algorithms*, chapter 6, pages 327–404. Addison-Wesley, 2005.

[TNLM05] Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Muñoz. Improving question answering using named entity recognition. In *Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, Alicante, Spain, June 15-17, 2005, Proceedings*, pages 181–191, 2005.

[TP10] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010.

[TT04] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Improving the performance of dictionary-based approaches in protein name recognition. *Journal of Biomedical Informatics*, 37(6):461 – 470, 2004. Named Entity Recognition in Biomedicine.

[Tur01] Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK, 2001. Springer-Verlag.

[UG96] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *KNOWLEDGE ENGINEERING REVIEW*, 11:93–136, 1996.

[vdMOK04] Dean van der Merwe, Sergei Obiedkov, and Derrick Kourie. *AddIntent: A New Incremental Algorithm for Constructing Concept Lattices*, pages 372–385. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[vJO+06] Jasmin Šarić, Lars Juhl Jensen, Rossitza Ouzounova, Isabel Rojas, and Peer Bork. Extraction of regulatory gene/protein networks from medline. *Bioinformatics*, 22(6):645–650, March 2006.

[VSKB10] S. V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *J. Mach. Learn. Res.*, 11:1201–1242, August 2010.

[WDSS06] Hai H. Wang, Jin S. Dong, Jing Sun, and Jun Sun. Reasoning support for Semantic Web ontology family languages using Alloy. *Multiagent and Grid Systems*, 2(4), 2006.

[WNK10] Nikil Wale, Xia Ning, and George Karypis. Trends in chemical graph data mining. In Charu C. Aggarwal and Haixun Wang, editors, *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*, pages 581–606. Springer, 2010.

[XL08] Shasha Xie and Yang Liu. Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4985–4988, March 2008.

[YH02] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 721–, Washington, DC, USA, 2002. IEEE Computer Society.

[YH03] Xifeng Yan and Jiawei Han. Closegraph: Mining closed frequent graph patterns. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 286–295, New York, NY, USA, 2003. ACM.

[YHA03] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177, 2003.

[YL05] Yongwook Yoon and Gary Geunbae Lee. Practical application of associative classifier for document classification. In *Proceedings of the Second Asia Conference on Asia Information Retrieval Technology*, AIRS'05, pages 467–478, Berlin, Heidelberg, 2005. Springer-Verlag.

[YLL10] Zhihao Yang, Hongfei Lin, and Yanpeng Li. Bioppisvmextractor: A protein–protein interaction extractor for biomedical literature using {SVM} and rich feature sets. *Journal of Biomedical Informatics*, 43(1):88 – 96, 2010.

[ZAC02] Osmar R. Zaïane, Maria-Luiza Antonie, and Alexandru Coman. Mammography classification by an association rule-based classifier. In *Proceedings of the Third International Conference on Multimedia Data Mining*, MDMKDD'02, pages 62–69, London, UK, UK, 2002. Springer-Verlag.

[Zak01] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.*, 42(1-2):31–60, January 2001.

[ZAR03] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, March 2003.

[ZH05] Mohammed J. Zaki and Ching-Jui Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Trans. on Knowl. and Data Eng.*, 17(4):462–478, April 2005.

[ZHY+16] Li Zhao, Minlie Huang, Ziyu Yao, Rongwei Su, Yingying Jiang, and Xiaoyan Zhu. Semi-supervised multinomial naive bayes for text classification by leveraging word-level statistical constraint. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2877–2883. AAAI Press, 2016.

[ZS02] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 473–480, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[ZSZZ05] Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[ZZA08] Min Zhang, GuoDong Zhou, and Aiti Aw. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. *Inf. Process. Manage.*, 44(2):687–701, March 2008.

[ZZH14] Deyu Zhou, Dayou Zhong, and Yulan He. Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics*, 30(11):1587–1594, 2014.

Résumé

De part leur grand nombre et leur sévérité, les maladies rares (MR) constituent un enjeu de santé majeur. Des bases de données de référence, comme Orphanet et Orphadata, répertorient les informations disponibles à propos de ces maladies. Cependant, il est difficile pour ces bases de données de proposer un contenu complet et à jour par rapport à ce qui est disponible dans la littérature. En effet, des millions de publications scientifiques sur ces maladies sont disponibles et leur nombre augment de façon continue. Par conséquent, il serait très fastidieux d'extraire manuellement et de façon exhaustive des informations sur ces maladies. Cela motive le développement des approches semi-automatiques pour extraire l'information des textes et la représenter dans un format approprié pour son utilisation dans d'autres applications.

Cette thèse s'intéresse à l'extraction de connaissances à partir de textes et propose d'utiliser les résultats de l'extraction pour enrichir une ontologie de domaine. Nous avons étudié trois directions de recherche: *(1)* l'extraction de connaissances à partir de textes, et en particulier l'extraction de relations maladie-phénotype (M-P); *(2)* l'identification d'entité nommées complexes, en particulier de phénotypes de MR; et *(3)* l'enrichissement d'une ontologie en considérant les connaissances extraites à partir de texte.

Tout d'abord, nous avons fouillé une collection de résumés d'articles scientifiques représentés sous la forme graphes pour un extraire des connaissances sur les MR. Nous nous sommes concentrés sur la complétion de la description des MR, en extrayant les relations M-P. Cette trouve des applications dans la mise à jour des bases de données de MR telles que Orphanet. Pour cela, nous avons développé un système appelé SPARE* qui extrait les relations M-P à partir des résumés PubMed, où les phénotypes et les MR sont annotés au préalable par un système de reconnaissance des entités nommées. SPARE* suit une approche hybride qui combine une méthode basée sur des patrons syntaxique, appelée SPARE, et une méthode d'apprentissage automatique (les machines à vecteurs de support ou SVM). SPARE* bénéficié à la fois de la précision relativement bonne de SPARE et du bon rappel des SVM.

Ensuite, SPARE* a été utilisé pour identifier des phénotypes candidats à partir de textes. Pour cela, nous avons sélectionné des patrons syntaxiques qui sont spécifiques aux relations M-P uniquement. Ensuite, ces patrons sont relaxés au niveau de leur contrainte sur le phénotype pour permettre l'identification de phénotypes candidats qui peuvent ne pas être références dans les bases de donnés ou les ontologies. Ces candidats sont vérifiés et validés par une comparaison avec les classes de phénotypes definies dans une ontologie de domaine comme HPO. Cette comparaison repose sur une modèle sémantique et un ensemble de règles de mises en correspondance définies manuellement pour cartographier un phénotype candidate extrait de texte avec une classe de l'ontologie. Nos expériences illustrent la capacité de SPARE* à des phénotypes de MR deja répertoriés ou complètement inédits. Nous avons appliqué SPARE* à un ensemble de résumés PubMed pour extraire les phénotypes associés à des MR, puis avons mis ces phénotypes en correspondence avec ceux deja répertoriés dans l'encyclopedie Orphanet et dans Orphadata ; ceci nous a permi d'identifier de nouveaux phénotypes associés à la maladie selon les articles, mais pas encore listés dans Orphanet ou Orphadata.

Enfin, nous avons appliqué les structures de patrons pour classer les MR et enrichir une ontologie préexistante. Tout d'abord, nous avons utilise SPARE* pour completer les description en terme de phénotypes de MR disposibles dans Orphadata. Ensuite, nous proposons de compter et grouper les MR au regard de leur description phénotypique, et ce en utilisant les structures de patron. Les structures de patron permettent de considérer d'une part des connaissances de domaine, ici une ontologie des MR et une des phénotypes ; et d'autre part les relations M-P issues des textes ou des bases de données. Le treillis généré a partir des structures de patron suggère une nouvelle classification des RD, et ainsi de nouvelles classes MR qui n'existent pas dans l'ontologie d'origine. Comme

leur nombre est important, nous avons proposé différentes méthodes pour ne sélectionner qu' un ensemble réduit intéressantes que nous suggérons aux experts pour une analyse plus poussée.

Mots-clés: Traitement automatique du langage naturel, extraction d'information, analyse formelle de concepts, structure de patron, enrichissement d'ontologie

Abstract

Even if they are uncommon, Rare Diseases (RDs) are numerous and generally sever, what makes their study important from a health-care point of view. Few databases provide information about RDs, such as Orphanet and Orphadata. Despite their laudable effort, they are incomplete and usually not up-to-date in comparison with what exists in the literature. Indeed, there are millions of scientific publications about these diseases, and the number of these publications is increasing in a continuous manner. This makes the manual extraction of this information painful and time consuming and thus motivates the development of semi-automatic approaches to extract information from texts and represent it in a format suitable for further applications.

This thesis aims at extracting information from texts and using the result of the extraction to enrich existing ontologies of the considered domain. We studied three research directions *(1)* extracting relationships from text, *i.e.,* extracting Disease-Phenotype (D-P) relationships; *(2)* identifying new complex entities, *i.e.,* identifying phenotypes of a RD and *(3)* enriching an existing ontology on the basis of the relationship previously extracted, *i.e.,* enriching a RD ontology.

First, we mined a collection of abstracts of scientific articles that are represented as a collection of graphs for discovering relevant pieces of biomedical knowledge. We focused on the completion of RD description, by extracting D-P relationships. This could find applications in automating the update process of RD databases such as Orphanet. Accordingly, we developed an automatic approach named SPARE*, for extracting D-P relationships from PubMed abstracts, where phenotypes and RDs are annotated by a Named Entity Recognizer. SPARE* is a hybrid approach that combines a pattern-based method, called SPARE, and a machine learning method (SVM). It benefited both from the relatively good precision of SPARE and from the good recall of the SVM.

Second, SPARE* has been used for identifying phenotype candidates from texts. We selected high-quality syntactic patterns that are specific for extracting D-P relationships only. Then, these patterns are relaxed on the phenotype constraint to enable extracting phenotype candidates that are not referenced in databases or ontologies. These candidates are verified and validated by the comparison with phenotype classes in a well-known phenotypic ontology (*e.g.,* HPO). This comparison relies on a compositional semantic model and a set of manually-defined mapping rules for mapping an extracted phenotype candidate to a phenotype term in the ontology. This shows the ability of SPARE* to identify existing and potentially new RD phenotypes. We applied SPARE* on PubMed abstracts to extract RD phenotypes that we either map to the content of Orphanet encyclopedia and Orphadata; or suggest as novel to experts for completing these two resources.

Finally, we applied pattern structures for classifying RDs and enriching an existing ontology. First, we used SPARE* to compute the phenotype description of RDs available in Orphadata. We propose comparing and grouping RDs in regard to their phenotypic descriptions, and this by using pattern structures. The pattern structures enable considering both domain knowledge, consisting in a RD ontology and a phenotype ontology, and D-P relationships from various origins. The lattice generated from this pattern structures suggests a new classification of RDs, which in turn suggests new RD classes that do not exist in the original RD ontology. As their number is large, we proposed different selection methods to select a reduced set of interesting RD classes that we suggest for experts for further analysis.

Keywords: Natural Language Processing, Information Extraction, Formal Concept Analysis, Pattern Structures, Ontology Enrichment