Universidade Estadual de Campinas
Instituto de Computação

Fagner Leal Pantoja

Generating Knowledge Networks from Phenotypic Descriptions

Gerando Redes de Conhecimento a partir de Descrições de Fenótipos

CAMPINAS
2016

Fagner Leal Pantoja

Generating Knowledge Networks from Phenotypic Descriptions

Gerando Redes de Conhecimento a partir de Descrições de Fenótipos

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. André Santanchè
Co-supervisor/Coorientador: Prof. Dr. Julio Cesar dos Reis

Este exemplar corresponde à versão final da Dissertação defendida por Fagner Leal Pantoja e orientada pelo Prof. Dr. André Santanchè.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): CAPES, 1406900

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

P196g

Pantoja, Fagner Leal, 1987-
Generating knowledge networks from phenotypic descriptions / Fagner Leal Pantoja. – Campinas, SP : [s.n.], 2016.

Orientador: André Santanchè.
Coorientador: Júlio César dos Reis.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Banco de dados. 2. Reconhecimento de entidades mencionadas. 3. Web semântica. 4. Fenótipo. 5. Processamento de linguagem natural (Computação). I. Santanchè, André,1968-. II. Reis, Júlio César dos,1979-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Gerando redes de conhecimento a partir de descrições de fenótipos
Palavras-chave em inglês:
Database
Named entity recognition
Semantic web
Phenotype
Natural language processing (Computer science)
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
André Santanchè [Orientador]
Ariadne Maria Brito Rizzoni Carvalho
Maria da Graça Campos Pimentel
Data de defesa: 05-08-2016
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Fagner Leal Pantoja

Generating Knowledge Networks from Phenotypic Descriptions

Gerando Redes de Conhecimento a partir de Descrições de Fenótipos

Banca Examinadora:

- Prof. Dr. André Santanchè
  Instituto de Computação - Universidade Estadual de Campinas
- Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
  Instituto de Computação - Universidade Estadual de Campinas
- Profa. Dra. Maria da Graça Campos Pimentel
  Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 05 de agosto de 2016

Vita brevis,
ars longa,
occasio praeceps,
experimentum periculosum,
iudicium difficile.

(Hippocrates)

Acknowledgements

I would like to thank my parents, Amarildo and Francisca, who taught me the value of hard work, for the support and dedication throughout my life. My brother Albert and my sister Linda for their warmth over the years and for encouraging me to pursue my dreams. My nieces (Sophia and Maitê), my sisters in law, my grandparents, grandmothers, cousins (I would like to highlight Beto, Elian, Juliana, and Gilberto), aunts and uncles. All of them – without exception – contributed in several and important ways.

I really need to thank my advisor professor André Santanchè, not just for the supervision, but also for being a friend that I owe so much (including the running times). Without him, I may never have gotten where I am. I thank Julio dos Reis and Patrícia Cavoto, who also support me to guide this work.

I thank my friends from PB2 fraternity (Fernando, Guilherme, Lucas, Júnior, Jefferson), LIS laboratory (Lucas, Jacqueline, Matheus, Leandro, Luana, Márcio, Lucas Augusto, Felipe, Francisco, Kleber, Ive, Victor, Celso) and Campinas city (Davidson, Denise, Ederlon, Ana, Jefferson). Each place where we shared moments gave me strength to continue in this city. I also to thank my friends from Belém (Suanne, Marcelo, Leonardo, Reinaldo, Igor, Marcel, Fabrício Mercy, Ana Luiza, Camilla), whose friendship did not decrease even along all this time of separation. Specifically, I need to cite Rafael de Sousa who has shared with me the initial and hardest moments.

Finally, I would like to thank the partial grants from[1] of CNPq (134205/2015-4), FAPESP (2014/14890-0), FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the Microsoft Research FAPESP Virtual Institute (NavScales project), CNPq (MuZOO Project), FAPESP-PRONEX (eScience project), INCT in Web Science, and individual grants from CNPq.

[1] The opinions expressed in this work do not necessarily reflect those of the funding agencies

Resumo

Diversos sistemas computacionais usam informações sobre seres vivos, tais como chaves de identificação – artefatos criados por biólogos para identificar espécimes de seres vivos seguindo uma cadeia de questões acerca das suas características observáveis (fenótipos). Tais questões estão em formato de texto livre, por exemplo, "Possui olhos grandes e pretos". Contudo, texto livre dificulta a interpretação de informação por máquinas, limitando sua capacidade de realização de tarefas de busca, integração e comparação de termos. Esta dissertação propõe um método para extrair informação a respeito de fenótipos a partir de textos escritos em linguagem natural, colocando-os no formato de Entidade-Qualidade – um formato de dados biológicos para representar estruturas anatômicas (Entidade) e o seu modificador (Qualidade). A proposta permite que Entidades e Qualidades, reconhecidas automaticamente a partir de informação do nível textual, sejam relacionadas com conceitos presentes em ontologias de domínio. Ela adota ferramentas de Processamento de Linguagem Natural existentes, bem como contribui com novas técnicas que exploram as características de escrita e estruturação implícitas em textos presentes nas chaves de identificação. A abordagem foi validada utilizando os dados da base *FishBase*, sobre a qual foram conduzidos experimentos explorando um conjunto de testes anotado manualmente para avaliar a precisão e aplicabilidade do método de extração proposto. Os resultados obtidos mostram os benefícios da técnica e as possibilidades de estudos científicos utilizando a rede de conhecimento extraída.

Abstract

Several computing systems rely on information about living beings, such as identification keys – artifacts created by biologists to identify specimens following a flow of questions about their observable characters (phenotype). These questions are described in a free-text format, *e.g.*, “big and black eye”. Free-texts hamper the automatic information interpretation by machines, limiting their ability to perform search and comparison of terms, as well as integration tasks. This thesis proposes a method to extract phenotypic information from natural language texts from biology legacy information systems, transforming them in an Entity-Quality formalism – a format to represent each phenotype character (Entity) and its state (Quality). Our approach aligns automatically recognized Entities and Qualities with domain concepts described in ontologies. It adopts existing Natural Language Processing techniques, adding an extra original step, which exploits intrinsic characteristics of phenotypic descriptions and of the organizational structure of identification keys. The approach was validated over the *FishBase* data. We conducted extensive experiments based on a manually annotated Gold Standard set to assess the precision and applicability of the proposed extraction method. The obtained results reveal the feasibility of our technique, its benefits and possibilities of scientific studies using the extracted knowledge network.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Nowadays, several knowledge bases contain information about living beings, including descriptive information to support the work of biologists. Phenotype descriptions play a key role in this context denoting the visible properties of an organism, which are consequence of the interaction of a genotype and the environment [28]. The increasing amount of available phenotype descriptions, on one hand, expands the possibilities of analysis; on the other hand, the consequent complexity requires more involvement and assistance of computers automating tasks. The interpretation of semantics by computers regarding phenotype descriptions in Biology implies in several benefits to the analyses of biological knowledge. Since most of the available descriptions rely on textual descriptions in natural language, lacking explicit semantics apt to be interpreted by machines, it is necessary to convert the information carried by them into a format that enables to automate analysis tasks. Our work contributes to this challenge through a system that interprets phenotype descriptions in free text format and automatically translates them to Semantic Web open standards.

In the following sections, we describe the scenario of our project, define our research problem and introduce our methodology.

1.1 Research Scenario

This work departed from a previous project [7] involving *FishBase*[1], which is a global information system that records a vast amount of information about fishes known to science. It currently contains data about 33,000 registered species encompassing several aspects of fishes – *e.g.*, taxonomic classification and ecosystems – with more than 2 million records [19]. Scientists, fisheries managers, zoologists, and others explore information from *FishBase* to support their activities.

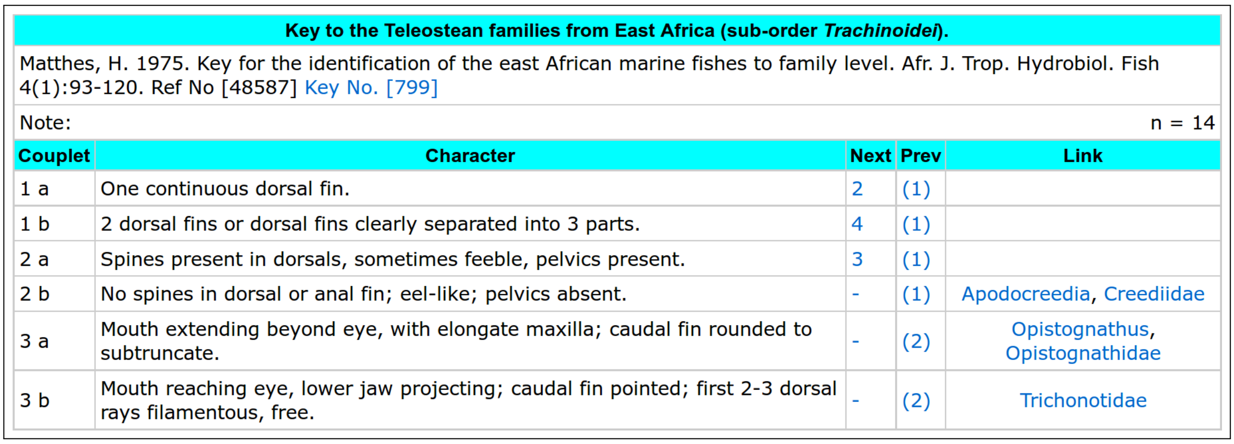
Among several types of data managed by *FishBase*, *Identification Keys* (IKs) consist in artifacts created by biologists to identify species or any other taxonomic group (called taxon) of an observed specimen [38]. An IK denotes a structured set of phenotype descriptions of organisms.

To identify a living being using an IK, users might navigate through a series of multiple

[1] www.fishbase.org

choice questions about the specimen characteristics. According to the picked answers, the path leads to the respective taxon. Currently, *FishBase* has 1,668 IKs of fishes containing 25,542 phenotype description sentences. They are one of the most relevant artifacts to support biological research involving fishes.

As an example of IK usage, Figure 1.1 presents an IK to identify the *Teleostean* families, from East Africa (sub-order *Trachinoidei*). The identification process begins with question 1, which has the pair of options 1a and 1b, with their descriptive texts in the *Character* column. According to the picked answer, the user might navigate to either question 2 or 4, indicated in the *Next* column. Each descriptive text inside the *Character* column is called *Key Question* (KQ). This process is repeated until the biologist reaches a row that does not lead to another question. At this stage, the specimen is identified and its respective taxon appears at the *Link* column.



Key to the Teleostean families from East Africa (sub-order *Trachinoidei*).

Matthes, H. 1975. Key for the identification of the east African marine fishes to family level. Afr. J. Trop. Hydrobiol. Fish 4(1):93-120. Ref No [48587] Key No. [799]

Note: n = 14

Couplet	Character	Next	Prev	Link
1 a	One continuous dorsal fin.	2	(1)	
1 b	2 dorsal fins or dorsal fins clearly separated into 3 parts.	4	(1)	
2 a	Spines present in dorsals, sometimes feeble, pelvics present.	3	(1)	
2 b	No spines in dorsal or anal fin; eel-like; pelvics absent.	-	(1)	Apodocreedia, Creediidae
3 a	Mouth extending beyond eye, with elongate maxilla; caudal fin rounded to subtruncate.	-	(2)	Opistognathus, Opistognathidae
3 b	Mouth reaching eye, lower jaw projecting; caudal fin pointed; first 2-3 dorsal rays filamentous, free.	-	(2)	Trichonotidae

Figure 1.1: Fragment of the *Identification Key* to the *Teleostean* families from East Africa (sub-order *Trachinoidei*). Source: `http://fishbase.org/keys/allkeys.php`

IKs can be organized as trees, where each root establishes the taxa to be identified and the internal nodes are the KQs containing descriptive texts. The edges conduct to alternative choices. The leaves specify the identified taxa, which are part or specializations of the taxa represented in the root. Figure 1.2 shows the same IK of the *Teleostean* families (sub-order *Trachinoidei*) in a tree-format representation.

Data in *FishBase* are stored in a set of relational tables. Handling all these data manually is a huge challenge for scientists, who face difficulties to analyse some scenarios involving the network of relations (links) among taxa and their characteristics. The overwhelming amount of phenotype descriptions is in free-text format. This format is more flexible and easier to produce, having advantages in the narrative structure and providing better expressiveness. However, this free-text format is inappropriate for some computational tasks, mainly when it involves the interpretation and comparison of the content by machines. It hampers tasks involving information retrieval and integration with other sources, since the description components are “locked” within the text.

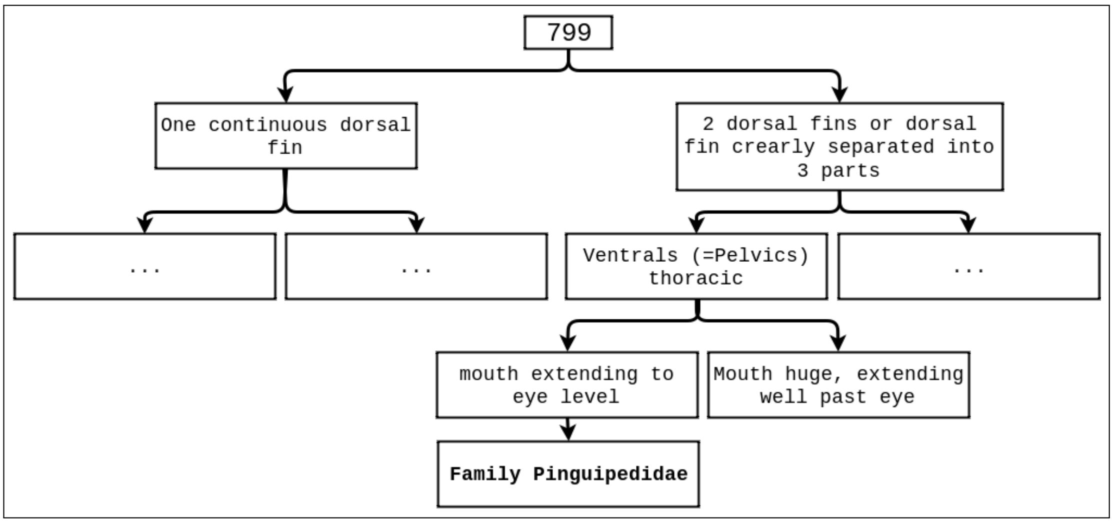

799
One continuous dorsal fin
2 dorsal fins or dorsal fin crearly separated into 3 parts
...
...
Ventrals (=Pelvics) thoracic
...
mouth extending to eye level
Mouth huge, extending well past eye
Family Pinguipedidae

Figure 1.2: Tree format representation of part of *Identification Key* 799 to *Teleostean* families from East Africa (sub-order *Trachinoidei*)

This observation motivated the previous project, developed by Cavoto *et al.* [7], to look for an alternative storage representation that favors this kind of analysis. It involved the transformation of *FishBase* data in a graph format to support network driven analysis – the *FishGraph*. Figure 1.3 shows its graph model, highlighting the types of nodes (SPECIES, GENUS, *etc.*) and relationships among them (SPECIES *belongs_to* GENUS, *etc.*). The *KEY* type node represents the IK comprising all its complex information (the set of *Key Questions* organized in a tree structure).

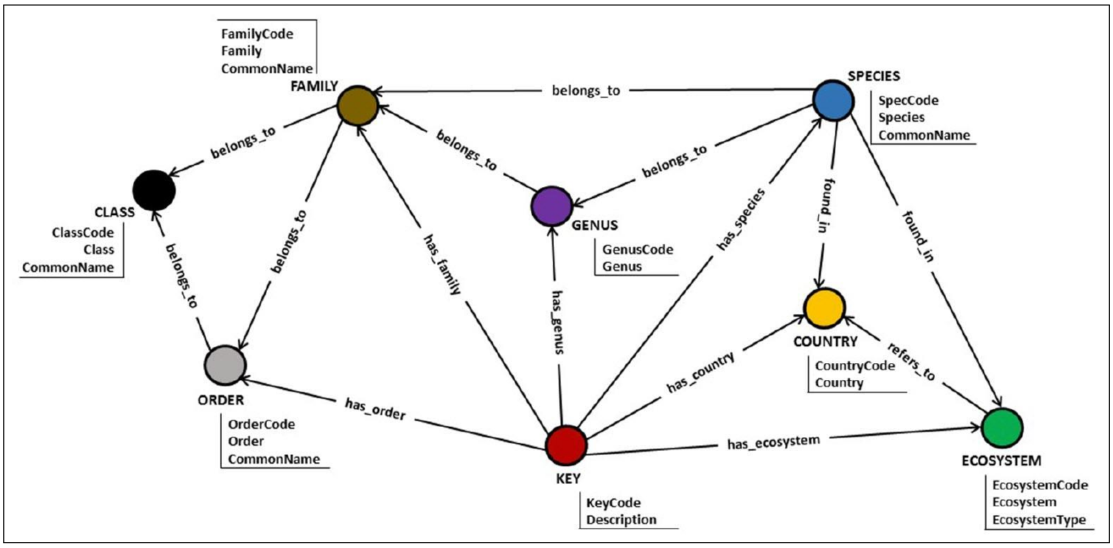


Figure 1.3: *FishGraph*: graph representation of *FishBase*. Source: Cavoto *et al.* [7]

1.2 Problem Definition

Phenotype descriptions inside *Key Questions* in *FishBase* are written in a textual format and are mainly composed by morphological characters of fishes, *e.g.*, "*eye*", and related qualifiers, *e.g.*, "*big*". Many problems arise when descriptions are written in free-text format, as follows:

- **Lack of a common and formal language:** the absence of a standardized terminology opens the possibility of writing the same description in different ways. For instance, "*median fin skeleton*" can be written as "*unpaired fin skeleton*" and "*axial fin skeleton*". In these cases, textual comparisons are insufficient. It is necessary to compare them on a semantic level, using the concepts behind them. This is a hard task without a language to formalize concepts, able to relate the syntactical structures with their respective explicit semantics, as ontology-based descriptions [8]. The subsequent problems are consequences of this one;

- **Consumable only by humans:** the lack of a mechanism to support a formal conceptualization hampers computer agents of interpreting the information carried by textual sentences and limits the scope of their role in operations like data aggregation, information retrieval, and reasoning;

- **Lack of interoperability with other systems:** there is a vast amount of systems relying on phenotype descriptions. The current *Linked Data* scenario makes it possible to share information among them [6]. However, their free-text format limits the capacity of exchanging data in a semantic level among different systems, resources, and applications.

In summary, the IKs usefulness does not depend only on the availability and accuracy of data, but also on a common and formal language (ontology) used to specify a set of core concepts, fostering the fully semantic interpretation of data by computers and expanding their capacity of performing more accurate and richer analyses over the information.

The problem of representing phenotype descriptions on a semantic level has been addressed by Grand *et al.* [21]. They synthesized the current scenario related to the description of organisms (*cf.* Figure 1.4), organizing the existing approaches in a series of progressive layers: (1) Textual description; (2) Structured description: splits the sentence into *Character* and *Character State* parts; (3) Semantic description: adopts the *Entity-Quality* (EQ) approach [41] to represent the elements (*Character* and *Character State*, from the previous layer) in a more detailed way (*Entity* and *Quality*).

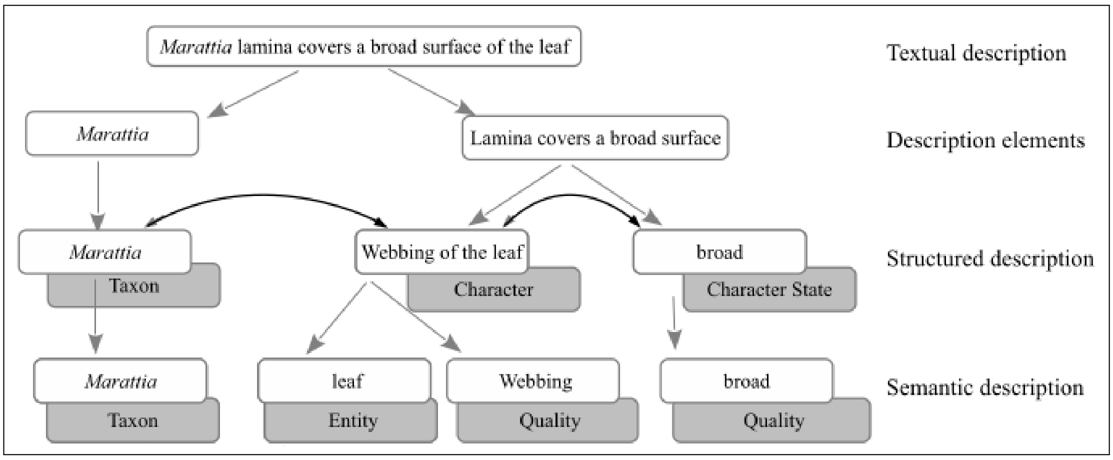


Figure 1.4: From textual and structured descriptions to semantic descriptions with specialized ontologies. Source: adapted from Grand *et al.* [21]

In the EQ representation, the *Entity* refers to the morphological or anatomical structure (*e.g.*, "*leaf*") and the *Quality* stands for a qualifier (*e.g.*, "*broad*") that specifies a given state of the *Entity*. Both refer to concepts defined in ontologies. The *Entity* comes from anatomical ontologies according to the organism and the *Quality* comes from a specialized ontology of *Qualities*. An EQ element is a relation between these two components.

The EQ model has been widely accepted and adopted by modern systems to make explicit the semantics of their phenotypic descriptions. In spite of the importance of this model, there is still a vast amount of legacy descriptions in free-text format – as in *FishBase*. Techniques to automatically transform these descriptive sentences into EQ representations are still an open problem and can play an important role to leverage the exploitation of existing descriptions, as well as to support the creation of semantically richer descriptions. This thesis is motivated by this problem proposing algorithms for automatic EQ recognition. It includes a strategy to distinguish, as automatically as possible, the anatomical entities and their qualifications inside a text of a phenotype description, making it possible to:

- **Reuse of Entities:** if phenotypic *Entities* are duly unified in a semantic level, it is possible to identify which IKs refer to the same *Entities*, making it explicit the interrelation network among IKs and *Entities*;

- **No need of previous knowledge:** in *FishBase*, IKs are segmented according to the taxa that they identify, like the sub-order *Trachinoidei* (*cf.* Figure 1.1). Therefore, users must know beforehand the specimen's taxon to pick a correct IK. This process is laborious and error-prone; in addition, it limits the use of the system only to expert biologists, who could not have previous clues about the specimen to be identified. An explicit and standard semantic representation might enable to correlate EQ elements of several IKs and combine them in a unified identification tree;

- **Relation between taxa and keys:** unified and semantic-enriched descriptions will enable to perform analyses to understand facts including: (i) which IKs identify similar taxonomic groups; (ii) which EQ elements are determinant to discriminate a taxon of a specimen; (iii) which EQ elements define a specific taxon.

1.3 Objective and contributions

This thesis aims to provide formal semantic expressiveness over the *Identification Keys* of *FishBase*. The resulting representation has been built over *FishGraph* [7]. As it is shown in Figure 1.5, we have improved the *FishGraph* database – presented in Figure 1.3 – adding new nodes that expand the *Key* node (which represents an *Identification Key*) as follows:

- **KeyQuestion:** unity that composes the tree of an *Identification Key*;
- **EQ:** *Entity-Quality* element extracted from a *Key Question*;
- **Entity:** part of the EQ representing the morphological structure;
- **Quality:** part of the EQ that qualifies an *Entity*.

We have also linked the existing taxonomic classification – species, genus, family, order, and class – to their respective EQs extracted from the IKs (*cf.* Figure 1.5).

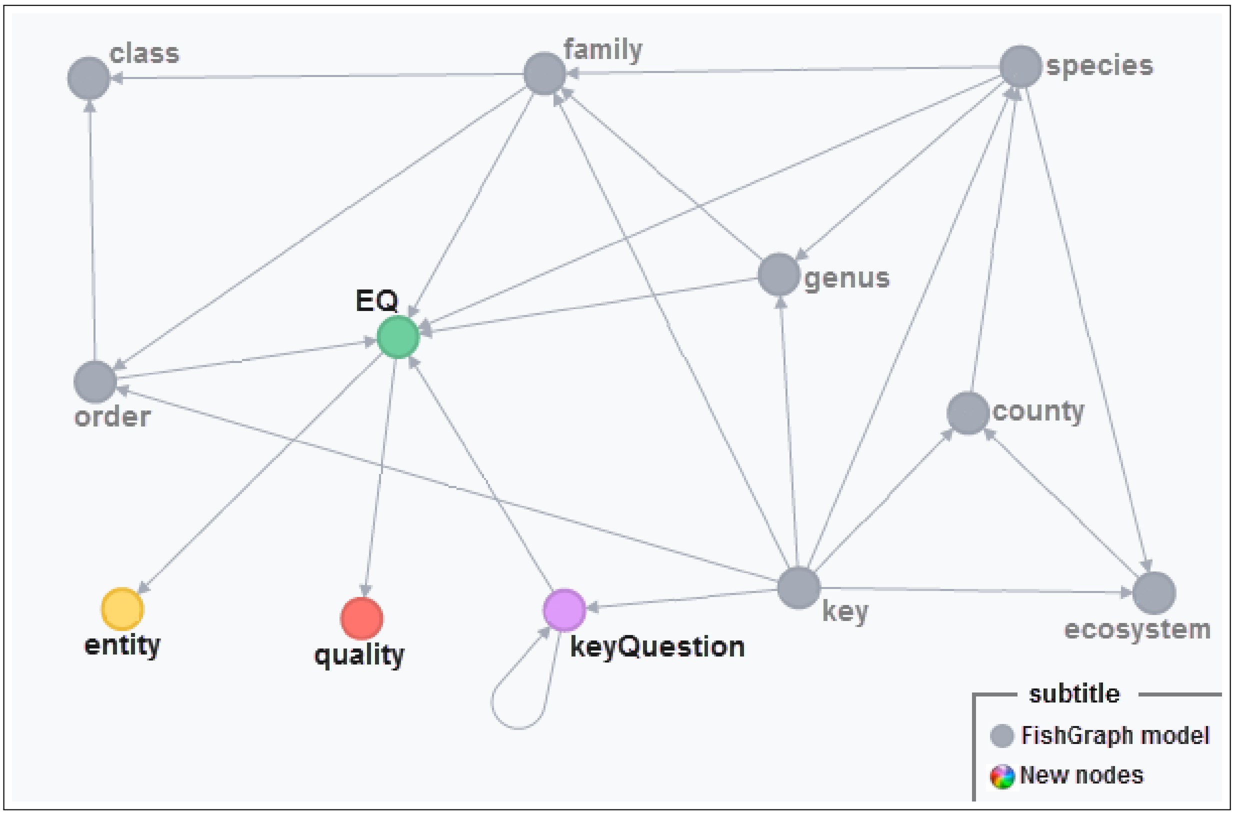


Figure 1.5: New nodes added in the *FishGraph* model.

In this work, we propose a method to detect the phenotype descriptions expressed in *Identification Keys*. Our proposal transforms the recognized elements into a semantic-based representation aligned to the EQ approach, going from the textual description layer to the semantic description layer according to Figure 1.4. It covers the following aspects:

- It requires a minimum human effort to recognize phenotypes, being accessible to users without biological expertise;
- It takes advantage of existing tools and resources;
- It links the recognized EQ elements to concepts of domain ontologies, in order to integrate *FishBase* to the Semantic Web scenario;
- It takes advantage of the characteristic way in which phenotype descriptions are written and structured to improve the automatic recognition.

This investigation defines a two-step method. The first step analyses the sentence using a Natural Language Processing technique that produces a *Dependency Tree*, establishing dependency relations between the sentence terms. It extracts EQ elements computing matches between ontology concepts and terms of the tree. We assume that the relations among terms in the *Dependency Tree* have latent *Entity-Quality* statements. They reflect the biologists approach to write phenotype descriptions: a term (or a set of terms) representing a given *Entity* has specific kinds of dependency with a term representing its *Quality*.

The second step takes advantage of the way that biologists relate and structure the phenotype descriptions. This step explores the correlations between sentences inside the IK. The identified *Entities* and *Qualities* are connected to domain ontologies to make their semantic explicit.

This thesis has the main contribution:

- An approach to extract phenotypic information from texts combining a Natural Language Processing technique with algorithms that explore the structure behind the interrelated descriptive texts;

1.4 Thesis outline

This thesis is organized in five chapters. Chapter 2 presents the foundations of our research and related work. It covers foundations about phenotype descriptions and information extraction. It also discusses open issues that related work fail to address in the EQ recognition.

Chapter 3 describes the proposed method for extraction and semantic linking of EQs. We detail the steps and algorithms that explore the characteristics of phenotypic description texts and the structure of IKs in the automatic recognition task.

Chapter 4 reports our experimental evaluation. We present scenarios of exploratory analyses, over the graph produced by us, aimed to examine the effectiveness of our proposal. The scenarios include a network analysis over the graph resulting from the integration of species with the recognized EQs. It discusses the achievements and limitations.

Chapter 5 provides a summary of the results and presents future work.

Chapter 2

Foundations and Related Work

This chapter presents some foundations and related work of our research. Section 2.1 provides an overview of systems to describe phenotypes and shows the importance of bringing these data to a machine-interpretable format. Section 2.2 presents the background of Phenotype Extraction. Section 2.3 presents some foundations of Information Extraction methods focusing in those related to the biology context. Section 2.5 makes a comparative presentation of the related work to indicate the original aspects explored in this investigation.

2.1 Phenotype Descriptions

There is a huge amount of biological data available in free-text format. As the process of producing biological data is expensive and complex, it is necessary to leverage the capability of automatically computing existing data. Thus, there is a challenge of migrating such vast amount of data into machine-interpretable formats, in order to produce semantically explicit knowledge.

These machine-interpretable data can be used by generic identification systems to improve their process and results. These systems implement different identification processes, such as: by descriptive characteristics, by pictures, by morphological measures, *etc.* Besides the generic identification systems, some information systems specialized in specific kinds of organisms may also offer support to build and publish IKs. For example, *FishBase* for fishes and *Bird Id* (`http://www.birdid.co.uk`) for birds. In the *FishBase* case, the identification process can be conducted in distinct ways, such as by images, by ecosystems, through descriptive characteristics, *etc.*

Several systems allow people to digitally create and publish *Identification Keys* for organisms [17], for example, *Intkey*, *IdentifyIt*, *Linnaeus II*, *Lucid* [27], *MEKA*, *NaviKey*, *PollyClave*, *XID*, *xPer* [44], *ActKey*, *eFloras*, *SLIKS*, and *KeyToNature* [31]. Technical reviews of some of these tools can be found in Dallwitz *et al.* [15].

Farnsworth *et al.* [17] give an overview of technical innovations and trends in the area and highlight the importance of ontologies and semantics. They also compiled a list of 50 species identification systems and concluded:

- 96% provide detailed data about the described taxa, including range maps, information on life histories, and distinguishing characteristics;

- 76% enable users to search or browse for a particular species of interest;
- 43% offer a glossary of technical terms or a dedicated help page with tips for usage;
- 39% allow users either to upload data to a central repository or to select other users to share data.

This list indicates the high quality achieved by systems to manage and publish phenotype descriptions. However, there is still space for improvements on the usage of these data, concerning data analysis and the correlation of phenotypes across different taxa and systems.

Existing investigations consider the use of phenotype descriptions in a machine-interpretable format. *Phenoscape*[1] addresses this issue adopting the *Entity-Quality* (EQ) approach to describe phenotypes and developing a scalable infrastructure that enables linking phenotypes across different fields of biology by the semantic similarity of their descriptions.

In the context of morphological characters data usage, Grosser *et al.* [22] proposed a method to identify taxa, based on K-Nearest-Neighbors. The algorithm consists in the computation of neighborhoods, based on a dissimilarity function. It handles taxa and specimen descriptions as complex objects, containing structured descriptions with characters and values. Although this method enables to identify species by proximity, it considers that there are beforehand structured descriptions as input to the algorithm, *e.g.*, characters following a controlled vocabulary will better perform in a comparison. Then, our work can be situated as an input to this method, since it will transform free-texts in structured descriptions – the complex objects treated by [22] – whose characters/states are more strictly defined by EQs.

2.2 Phenotype Extraction

Concerning how to make explicit the semantics of biological data, Dahdul *et al.* [13] investigated techniques for transforming descriptive biology texts into a format that enables large-scale computation. Based on a previous study, they claim that large-scale computation can benefit from annotating characters with ontology terms. Therefore, they advocate the need of efficient methods to automatically extract and annotate phenotypes from descriptions and consider that NLP tools can be used in the process. Natural Language Processing (NLP) tools can be used in the process.

In the following subsection, we present foundations about some of NLP tasks related to this work, emphasizing information extraction.

2.3 Natural Language Texts in Information Extraction

Information Extraction (IE) refers to a research area, which addresses the transformation of natural language texts into a format interpretable by machines. It aims to provide

[1] http://phenoscape.org/wiki/Main Page

structured and useful facts for information integration and retrieval. This research area is also known by the most general label Text Data Mining (or simply Text Mining) [43].

Among several IE tasks, the biological and biomedical research areas recurrently investigate: Named Entity Recognition, Relation Extraction, Event Extraction, Summarization, Question Answering, Literature Based Discovery, *etc.* The specialized literature (*e.g.*, [25], [40], [46], [9]) provides surveys and studies on current problems, progresses, and challenges.

This thesis is interested, on one hand, in Named Entity Recognition (NER) to identify entities inside texts, and on other hand, in Relation Extraction to identify ocurrences of particular types of relationships between pairs of entities [40]. Therefore, these two tasks are covered in this subsection.

NLP tools and external resources aid the extraction methods, as illustrated in Figure 2.1. The following subsections cover NLP tools, External Resources, NER methods, and Relation Extraction methods.

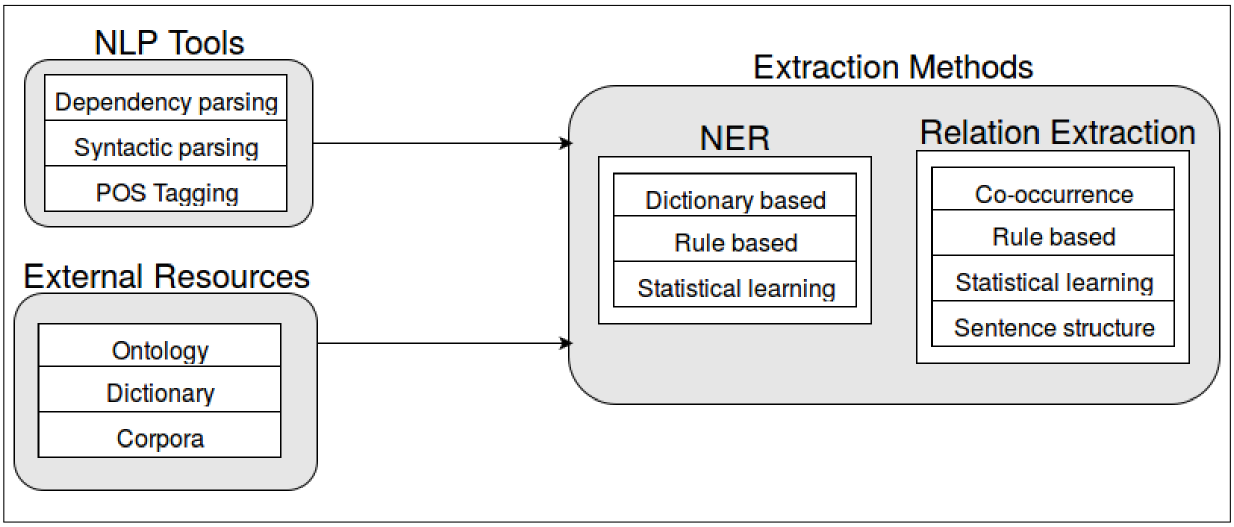


Figure 2.1: General view of Extraction Methods and its association with NLP Tools and External Resources.

2.3.1 NLP Tools

There are several NLP tools varying at sophistication and language interpretation. The levels of interpretation start with words, progressing to sentence structure (syntax) towards the sentence meaning (semantics).

POS tagging. A Part-Of-Speech (POS) Tagger is a tool that recognizes and assigns parts of speech annotations to elements of a given text. The simplest kind of POS tagging, called Shallow Parsing, recognizes and assigns the most basic tags to each individual word, *e.g.*, noun, verb, and adjective. A more sophisticated POS tagging (Deep Parsing) uses more fine-grained tags, *e.g.*, noun-plural. They are also able of tagging compound noun phrases. Penn Treebank Tag Set is the most common collection of tags used by such taggers, having labels for different parts of speech such as adjective phrases (ADJP), plural nouns (NNP), and so forth [43].

Dependency parsing. Dependency parsers model a sentence as a set of relationships between words. As illustrates Figure 2.2, they produce a graph, called Dependency Tree, for each sentence, where each node represents a word and each edge denotes a grammatical dependency – as the dependency *nsubjpass*, which relates the verb *isolated* and its subject *homologues* (*cf.* Figure 2.2). All the dependencies are binary relations. A grammatical dependency involves a governor (also known as regent or head) and a dependent [16].

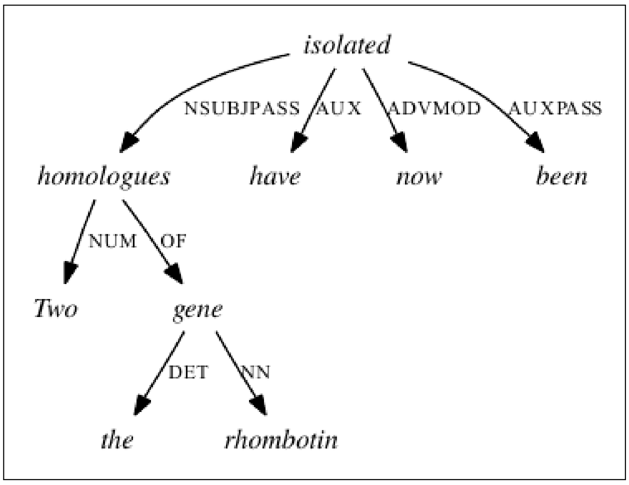


Figure 2.2: The dependency graph of the sentence: "Two homologues of the rhombotin gene have now been isolated". Source: Clegg and Shepherd [11].

2.3.2 Support of External Resources

As shows Figure 2.1, external resources give support to extraction methods. Common nouns listed in a dictionary, for instance, are useful in the disambiguation of capitalized words in ambiguous positions (*e.g.*, sentence beginning). Other examples of external resources include glossaries, gazetteers, vocabularies, ontologies, *etc.*

There are also text collections in the biological domain. Some of them are specialized or annotated corpora. An example is MEDLINE, the primary resource in biomedical text mining, which contains bibliographic references to journal articles in life sciences, with a concentration on biomedicine [43].

2.3.3 Named Entity Recognition

Named Entity Recognition (NER) consists of identifying and classifying types of information elements, called *Named Entities* (NE). There are several and different definitions for Named Entity (NE). In this investigation, we define NE according to its purpose and application domain, *i.e.*, the goal of the NER task defines what is considered a NE [30].

The most common types of entities are proper names (names of persons, organizations, locations), numeric expressions (quantities, monetary amounts), and dates. Important entities in the biology domain include genes, proteins, diseases, drugs, body parts, *etc.* Many entities are expressed as terms (*e.g., "polyostotic fibrous dysplasia"*) rather than proper names (*e.g., "McCune-Albright"*). The notion of named entity is also applied to them [46].

There are several and different classifications for NER extraction methods. Considering the scenario of this research, the NER approaches can be classified into three categories: (1) Dictionary-based, (2) Rule-based, and (3) Statistical Learning approaches. They are further detailed.

Dictionary Based. Use resources – *e.g.*, a dictionary, an ontology, a list of terms, *etc.* – containing known entities in order to identify NE occurrences in a text. Such approaches determine whether a word or group of words, identified in the text, exactly matches a term from the resource.

Methods using only the dictionary-based approach generally exhibit reasonable precision, but suffer from poor recall due to spelling mistakes and morphological variants [40]. Dictionaries are also seldom complete lacking variants and new names, limiting such approaches [46]. Another bottleneck is the high rate of false positives, *i.e.*, entities mistakenly recognized by the method. They are inherent to the use of short names, which significantly degrade the overall accuracy. Exclusion of short names from the dictionary may resolve this issue, but it is not the ultimate solution since it disallows the recognition of entities with a short name, reducing the recall [42].

Hirschman *et al.* [24], Ono *et al.* [36], and Aronson *et al.* [4] are examples of works this approach. One way to improve the result involves generating spelling variants for the listed terms appending them to the word list.

Rule-Based. Rule-based approaches act through rules that describe pattern structures for certain classes, based on their morphological, orthographic, and syntactic characteristics [34, 23]. Generally, the rule-based approach can be improved using contextual information and syntactic parsers to determine the NE boundaries. This approach typically achieves better results than the dictionary based approach. Nevertheless, the manual generation of the required rules is a time-consuming process. The rules are usually very specific in order to achieve high precision and must be customized to each domain [40].

Statistical Learning. Such approaches are based on a learning process. Nadeau and Sekine [33] classify them in three categories:

- **Supervised learning (SL):** uses only labeled data (feature/label pairs) for training. The main shortcoming of the SL is the requirement of a large annotated corpus as input. SL techniques include Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models (ME), Support Vector Machines (SVM), and Conditional Random Fields (CRF).

- **Semi-supervised learning (SSL):** combines a large amount of unlabeled data with a small amount of labeled data as a set of seeds, to start the learning process. One of the main techniques for SSL is called Bootstrapping.

- **Unsupervised learning (UL):** used when labeled data are not available for training. Clustering is a typical approach. It looks for hidden patterns grouping the data. The clusters can be computed using a measure of similarity, which is defined by metrics such as Euclidean or probabilistic distances.

Machine learning techniques are seen as an alternative to overcome the problems inherent to dictionary-based approaches [2]. They depend on the existence of either training data (for SL and SSL) – which is often difficult, expensive or time-consuming to obtain, as they require the efforts of experienced human annotators – or complementary resources, such as lexical resources (for UL).

2.3.4 Relation Extraction

Consists in identifying relationships among NEs, going beyond the NER task. Relation Extraction is defined as a task that copes only with associations between two entities [46, 40], *i.e.*, binary relations. When the association involves three or more NEs, called complex associations, it is treated as an Event Extraction, another kind of NLP task.

Relation extraction faces many challenges according to the chosen approach, such as the generation of rules or the creation of annotated corpora for training and evaluating relation extraction systems. These tasks are considerably more complicated in a Relation Extraction task than in a NER task [40].

There are many Relation Extraction approaches. We further present one possible classification to them [40].

Co-occurrence. It is the simplest approach. The basic principle is: if a given NE frequently occurs together with another NE, then it is likely that they are related. This approach does not determine the type and direction of the extracted relation. Commonly, it exhibits high recall and low precision.

Rule-Based. Performs the extraction using linguistic patterns previously defined for particular relations. The rules can be manually inferred by domain experts or they can be derived from an annotated corpora. For example, the pattern *modal verb + participle + preposition* is likely to express a relation, such as "is composed by", "is generated with" [18]. Typically, this approach results in high precision and low recall.

Statistical Learning. Approaches under this category use statistical learning methods, which can be supervised, unsupervised or semi-supervised, similar to those in NER. They can be trained on a tagged corpus to learn which combinations of cues are the most effective in detecting relations.

Sentence Structure Based. In this category, approaches use syntactic parsers in order to take advantage of the sentence structure. The syntactic parser outputs dependency trees or graphs, which encode grammatical relations between phrases or words [11].

2.4 NER, Relation Extraction, and Hybrid Approaches

NER and Relation Extraction can be seen in this work as complementary tasks since to identify the relations it is necessary previously to recognize the involved entities. Each presented method for NER and Relation Extraction has limitations and advantages. The decision for a method will consider the research scenario and requires to take into consideration the trade-off among performance in terms of precision and recall, availability of external resources (such as ontologies), the cost of annotating data, and so on.

An alternative is to combine approaches in order to balance their shortcomings. Several systems rely on multiple (hybrid) techniques plus several resources, such as the combination of dictionary matching with either rule-based or statistical methods to reduce the number of false positives [40].

2.5 Related Work

This subsection presents related work of NER and Relation Extraction in the biology context. Table 2.1 summarizes the related work presented in this subsection, emphasizing three aspects:

- Extraction: kind of extraction preformed: NER, Relation Extraction, or both;
- Context: focus of the extraction;
- Approach: specifies which approach each related work explores, as previously described.

Related work concerning phenotype extraction are mostly concentrated in interactions among genes, proteins, drugs, and diseases, as it is shown in the first five investigations in Table 2.1. Although the domains are similar to our work, we exploit specific peculiarities on organisms morphological descriptions to improve the results of our extraction.

Ciaramita *et al.* [10] present an unsupervised model for learning to recognize relations between concepts of a molecular biology ontology, inside an input text. Relations are extracted and learned from the GENIA corpus – an annotated corpus coming from research abstracts of the MEDLINE databases [35]. They do not propose a NER method, so they manually tagged the entities that are used in the RE process. While Ciaramita *et al.* [10] depart from existing entities in ontologies to learn their relations, our approach recognizes and extracts *Entities* and look for their relations with *Qualities*. Our NER approach applies a similar process of existing methods (*e.g.*, Song *et al.* [42], Pyysalo and Ananiadou [38], and Ramakrishnan *et al.* [39]) to identify entities representing anatomical structures in the biomedical domain, as further detailed.

Song *et al.* [42] present a hybrid dictionary-based extraction technique. The entities are recognized by matching the sentences against the *Medical Subject Headings* (MeSH)[2] tree and *PubMed* records. In order to overcome the problems related to short names and spelling variations (issues mentioned in Section 2.3.3), they use an edit-distance algorithm

[2] A hierarchically-organized terminology concerning biomedical information

Table 2.1: Related Work

Reference	**Extraction**	**Context**	**Approach**
Ciaramita *et al.* [10]	RE	Interactions in molecular biology	Unsupervised Statistical Learning and Rules over Dependency Trees
Song *et al.* [42]	NER	Biomedical anatomic entities	Dictionary-based
Pyysalo and Ananiadou [38]	NER	Biomedical Anatomic entities	Supervised statistical learning
Ramakrishnan *et al.* [39]	NER, RE	Biomedical Anatomical entities	Dictionary-based, Rules over Dependencies Trees and Statistical Learning
Fundel *et al.* [20]	NER, RE	Gene and Protein Interaction	Rules over Dependency Trees
Cui [12]	NER, RE	Morphological structures of organisms	Statistical Learning - Unsupervised

called *Shortest Path Edit Distance* (SPED). To address similar concepts, they use rules acting in the lexical and syntactic levels. Our approach also exploits the syntactic relations between words, but they are used to drive our process to discover entities, which are connected and confirmed as concepts in ontologies.

Pyysalo and Ananiadou [38] propose a statistical learning method, called *Anatomy-Tagger*, to extract anatomical entities from a corpus of scientific papers. It identifies all contiguous non-overlapping sequences of characters that refer to anatomical entities in an unstructured text. The approach assigns each entity to exactly one type of a given set of ontological categories. Figure 2.3 presents their proposed workflow. After applying classical processes such as *Segmentation and preprocessing*, as well as *Morphosyntactic analysis*, the ontologies aid in the generation of features (*Lexical resources* boxes) to be used in the NER statistical learning phase starting in *1st stage* box. The 2nd stage incorporates non-local features coming from the 1st stage. Like Pyysalo and Ananiadou [38], our investigation also matches the entity mentions against ontology concepts, but it is not limited to contiguous sequences of characters, once the entities may not be constituted by continuous words in the sentence.

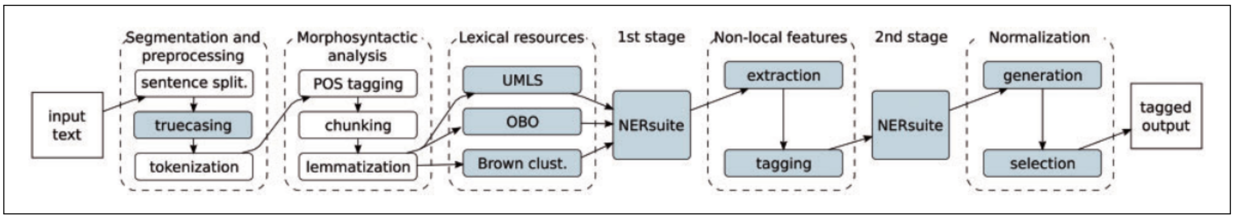


Figure 2.3: *AnatomyTagger* Architecture. Source: Pyysalo and Ananiadou [38].

In order to handle non-adjacent tokens, our work explores the Dependency Trees in an approach inspired in Ramakrishnan *et al.* [39]. It departs from the grammatical dependency tree of the sentence. The method iterates over each dependency relation and decides whether it is either part of given entity or is part of a relation between entities, based on a set of rules. Figure 2.4 shows the tree of the sentence "*Anti-Ro(SSA) autoantibodies are associated with T cell receptor beta genes in systemic lupus erythematosus patients*". The dependency *nsubjpass* (passive nominal subject) is considered a cut point since it bridges the term *autoantibodies*, marked as the head of a subject (a possible entity), and the term *associated*, marked as the head of a predicate (the sentence part modifying the subject). The application of the rule set to the dependencies *prep_ with* and *prep_ in* combined with their upper term *associated* – result in the relational roles *associated_ with* and *associated_ in*, between *associated* and their dependents *genes* and *patients*. The method recursively expands, looking for candidate entities in the other side of the relation collecting modifiers to compose a token sequence that could reveal a compound entity. The recursive expansion procedure results in the sentence candidates for entities "*Anti-Ro(SSA) autoantibodies*" "*T cell receptor beta genes*", and "*systemic lupus erythematosus patients*".

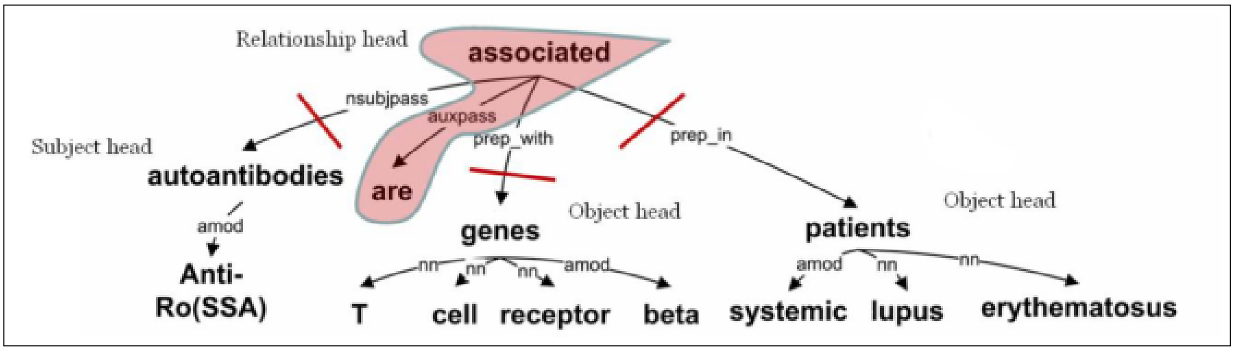


Figure 2.4: Dependency parse tree of a given sentence. Source: Ramakrishnam *et al.* [39].

The second phase of the Ramakrishnan' algorithm matches the head term of each candidate sentence with single-word MeSH terms and further with classes in the *Unified Medical Language System* (UMLS) – a language that systematizes terms in biomedicine and health. Our work analyses the relations in the dependency tree following an approach similar to Ramakrishnan *et al.* [39], adapted to our scenario – phenotypes. In particular, we have extended the analysis to identify relations between *Entities* and *Qualities*.

Furthermore, this investigation is similar to ours as it looks for terms in a knowledge base, but we differ in the match process, considering all possible combinations of terms to form an *Entity*, not only the head.

Fundel *et al.* [20] also use dependency tree. Their method combines dictionaries, dependency parse trees, and rules in a process showed in Figure 2.5.

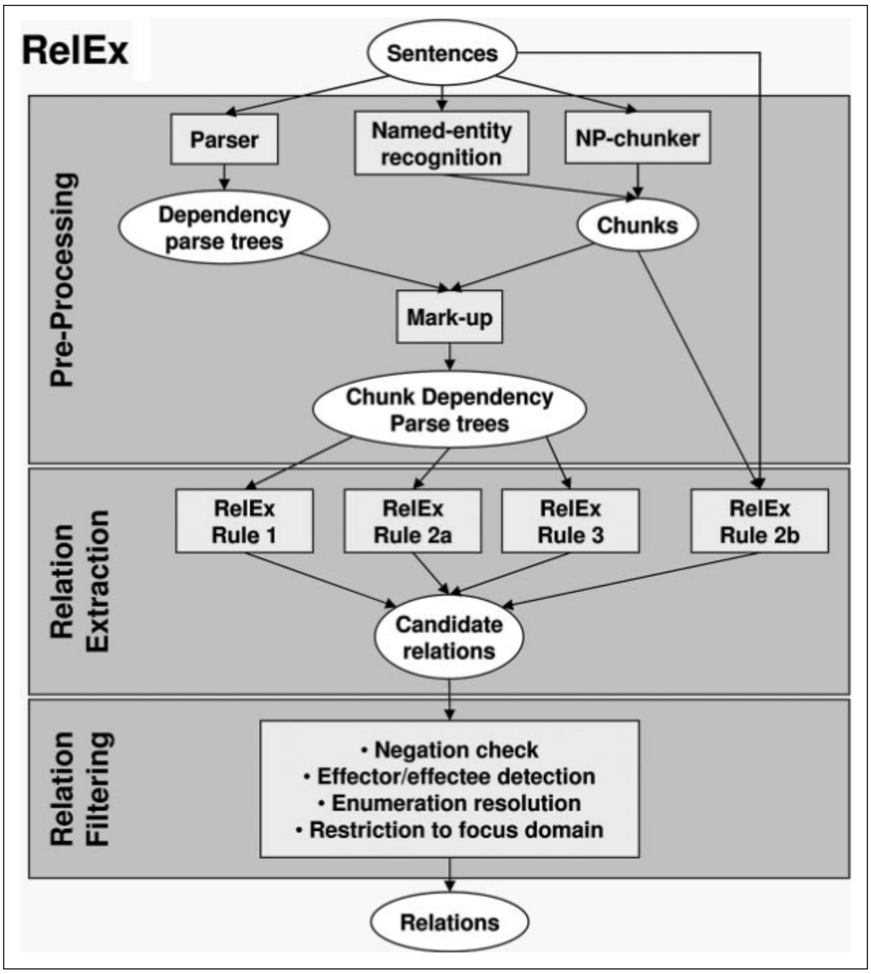


Figure 2.5: *RelEx work-flow*, subdivided into pre-processing, relation extraction, and relation filtering. Source: Fundel *et al.* [20].

At the Pre-Processing step, the NEs are recognized by a matching process involving the terms of the input sentence and a synonym dictionary of gene and protein names. The Relation Extraction and the Relation Filtering steps follow a rule-based approach similar to the previous work of Pyysalo and Ananiadou [38]. A set of rules establishes as input patterns to be recognized in the dependency tree and as output inferred relations. We further show some examples of rules to illustrate the approach. The Relation Extraction step defines three rules, Where the terms effector and effectee are NEs found in the previous step:

1. effector-relation-effectee (*e.g.*, A activates B);

2. relation-of-effectee-by-effector (*e.g.*, Activation of A by B);

3. relation-between-effector-and-effectee (*e.g.*, Interaction between A and B).

The Relation Filtering step treats specific relations, like negations and enumerations. The rule set of both steps is manually generated, *i.e.*, in the scenario of the study conducted in [20], it requires a deep study of all kinds of interactions between proteins and genes. In our approach, we still address simple forms of relationships between *Entity* and *Quality*. They are usually an 'is' relationship (*e.g.*, in the sentence *big eye* the relationship is eye-is-big). Therefore, we focused our attention on this kind of relation, but we consider that rules could expand our spectrum of analysis and we are considering it in future work.

The literature presented so far address the biomedical domain. There is a smaller set of contributions related to the domain of our research – recognition of phenotypic descriptions of organisms – as show the two last rows of Table 2.1.

Cui [12] presents a method to extract phenotypes that describe leaves, fruits, and nuts of plants. He uses two key techniques: (a) an unsupervised learning algorithm to annotate descriptions at the sentence level, to build a lexicon (step 1 on Figure 2.6); (b) the learned lexicon, enhanced by a human user, feeds a parser that recognizes biological characters in descriptive sentences and annotates them (step 2 on Figure 2.6). Our work differs since it does not require human intervention during the process, in such a way that a non-expert can use the system.

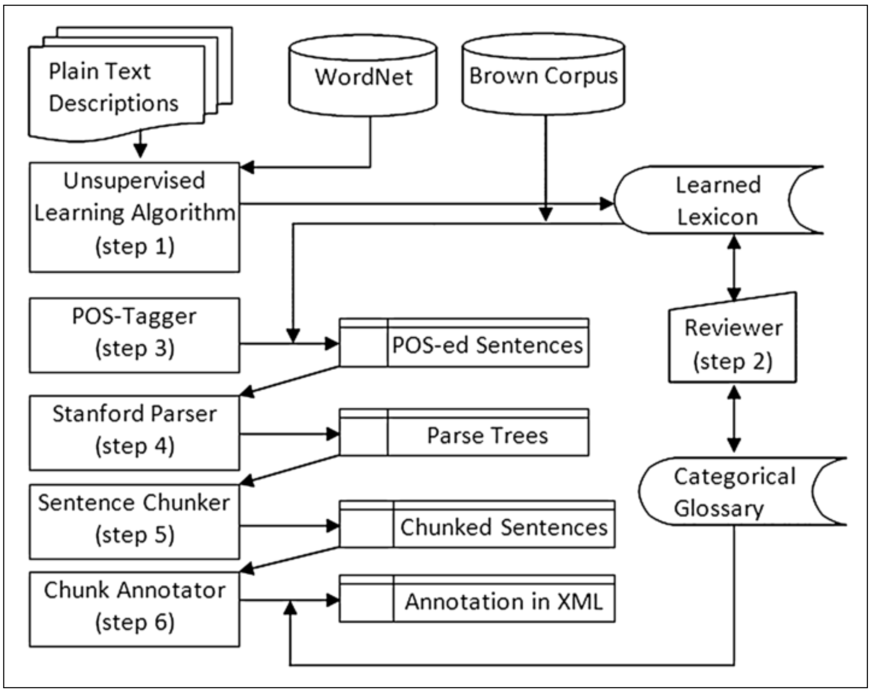


Figure 2.6: *CharaParse* System Architecture. Source: Cui [12].

Alnazzawi *et al.* [1] compare several statistical learning methods against a curated corpus made by experts, called *PhenoCHF*. This corpus contains annotations about phenotypic information related to Congestive Heart Failure (CHF). One of their objectives is to demonstrate how the well-known methods perform better when a curated corpus is available. However, the creation of a corpus is a hard and expensive task. Our approach was developed to serve in contexts in which such corpus are unavailable.

2.6 Summary

This chapter provided a literature review concerning phenotype description and extraction. We reported relevant techniques of information extraction useful to this research context. In particular, we focused on the task of Named Entity Recognition and Relation Extraction. For both tasks, we described a categorization of solution approaches. We explained the NLP basic tools and background resources, which support the existing approaches. We presented the related work with their advantages and drawbacks. This chapter demonstrated how our proposal differs from existing ones and to which extent we rely on defined techniques.

The literature on phenotype extraction presents a concentration in the Biomedical field, looking for interactions between genes, proteins, drugs, diseases and so on. These phenotype descriptions are typically written in the standard English syntax (due to the nature of papers and other documents containing them), as opposed to the syntax of the language seen in phenotype description in *Identification Keys* (as will be further detailed), *e.g.*, they usually omit function words. Although they are similar domains, there are peculiarities on the latter to be considered in order to improve the results on such scenario.

Our approach extracts texts of *Identification Keys* (IK), which enabled us to consider an extra factor to improve the extraction not addressed by related work: the characteristics structure of the IK. It is aligned with Wong *et al.* [45], which states that multiple noncontent cues, such as fonts and layout information, may be used to assist extraction. Next chapter presents the description of our solution for EQ recognition from phenotype description texts.

Chapter 3

Entity-Quality Recognition Method in Identification Keys

This chapter details our approach to recognize and to make explicit *Entity-Quality* (EQ) elements, which are part of textual descriptions inside semi-structured *Identification Keys* (IKs). The method involves mapping them to a more formal representation with explicit semantics, based on domain ontologies. The approach departs from natural language text sentences (phenotype descriptions) and produces a graph representation of the recognized EQs. Figure 3.1 shows the general view of our approach, which encompasses two steps:

- **Step 1:** recognizes EQ elements through an algorithm that analyses the text of the sentence;

- **Step 2:** improves the results of Step 1 recognizing more EQ elements through an algorithm that analyses the relations of sentences according to the structure of the IK.

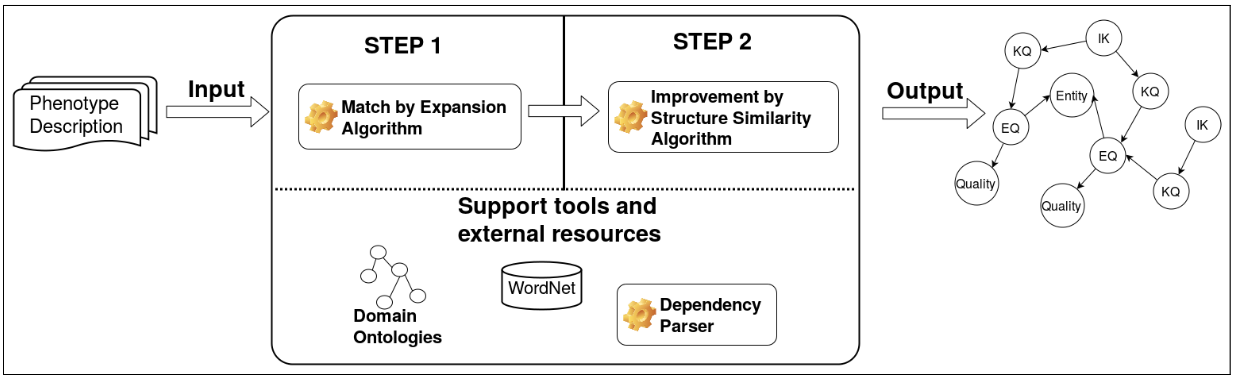

Phenotype Description
Input
STEP 1
Match by Expansion Algorithm
STEP 2
Improvement by Structure Similarity Algorithm
Output
Support tools and external resources
Domain Ontologies
WordNet
Dependency Parser
IK
KQ
Entity
EQ
Quality

Figure 3.1: General view of the proposed approach.

Both steps rely on external tools and resources throughout the process. This proposal is founded on the two following assumptions that synthesize the principles behind our method.

Assumption 1 *The typical way in which a phenotypic description is written can guide the extraction of EQ elements.*

Assumption 2 *The way in which a set of phenotype descriptions is organized and structured holds implicit relations that can be exploited to improve the extraction of EQ elements.*

Steps 1 and 2 implement algorithms based on Assumptions 1 and 2, respectively (*cf.* Sections 3.1 and 3.2). We further define a notation to be used throughout this chapter, which will support the explanation of the method.

$E[e_x]$ = an Entity

$Q[q_y]$ = a Quality

$EQ[e_x, q_1, q_2, ..., q_n]$ = an Entity-Quality

$S[s_x]$ = sentence in free-text format

$V[v_1, v_2, ..., v_n]$ = vertexes of a Dependency Tree

As presented in Introduction (*cf.* Chapter 1), we extract EQs from *Identification Keys* (IKs), which are decision trees where users navigate as they answer *Key Questions* (KQs). More specifically, the EQs are extracted from the descriptive texts inside KQs.

Figure 3.2 gives an overview of the process through an example of our approach running over an IK. Part *(a)* shows a fragment of an IK, which is the input of the process. Part *(b)* shows Step 1, which focuses on the relations among terms within one sentence of a KQ. Part *(c)* presents Step 2, which focuses on exploring the correlations among KQs and their respective sentences in the organizational structure of an IK. Step 2 improves the results of Step 1. The method generates as output a graph, containing the phenotype descriptions that distinguishes *Entities* and their related *Qualities* in an EQ format.

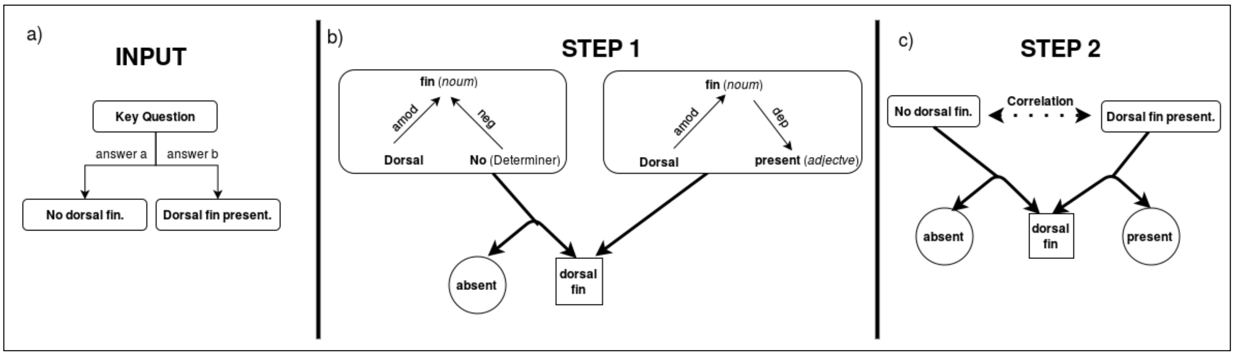


Figure 3.2: Steps of our method to extract EQs from semi-structured texts.

To exemplify the outcome of our method, Figure 3.3 shows an IK where the EQ elements and their respective *Entities* and *Qualities* are explicitly represented. The original sentences extracted of the KQs are S[*No dorsal fin*] and S[*Dorsal fin present*]. From these sentences, "*dorsal fin*" is recognized as E[*dorsal fin*]. The terms "*no*" and "*present*" are recognized as Q[*absent*] and Q[*present*], derived from the first and second sentences respectively. The EQs are the bridges among an *Entity* and its *Qualities*, expressing a statement. For example, the EQ[*dorsal fin, absent*] – *i.e.*, the node EQ connected to the node E[*dorsal fin*] and to the node Q[*absent*] – expresses the sentence "no dorsal fin".

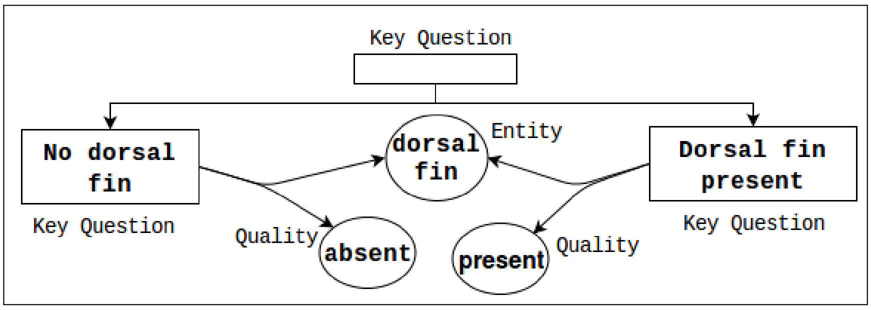


Figure 3.3: Example of the method output.

3.1 Step 1: Exploiting the Writing Characteristics of Phenotypic Descriptions

Following Assumption 1, in order to guide the extraction task, this step exploits the typical approach followed by biologists to write phenotypic descriptions. This principle was previously exploited by other authors like Cui [12], who listed out some writing characteristics observed in Biology description texts:

1. Generally, morphological descriptions are constituted by two elements: Characters and Character States (C/CS);

2. Omission of Function Words – it is usual the omission of words that do not carry relevant meaning, such as articles and auxiliary verbs (*e.g.*, a, an, the, is, are);

3. Morphological characters are often not explicitly stated in the descriptions. For example, in the sentence "*Black and big eyes*", the characters color and size are not explicitly stated.

As stated in the first item, description sentences are mainly composed by Nouns and Adjectives. However, it is not a trivial computational task to identify grammatical classes. Even though this identification could be conducted by a Part-Of-Speech (POS) tagger, they could assign a wrong tag to a term, *e.g.,* "curved" should be tagged as an adjective, but it is frequently tagged as a verb. To deal with the first item, we have chose to work with Dependency Trees in order to reveal relations between sentence terms reflecting C/CS relations. The relations can reveal non-local dependencies within sentences – *i.e.*, dependencies between words that are far apart in a sentence – which is a common case in phenotype description sentences, for example, in S[*Dorsal and anal fins*] the word *dorsal* is related to word *fins*, although the distance between them.

In order to identify these relations, this step uses a *Typed Dependencies Parser* (TDP), which is based on a NLP technique that captures grammatical relations between words inside a sentence [16] producing a dependency tree, as demonstrated in Section 2.3.1, where each node represents a word, and each edge denotes a grammatical dependency. The dependencies are all binary relations [16]. This task is executed by the *Stanford Typed Dependencies Parser*[1] (STDP), which belongs to the *Stanford Core NLP*

[1] http://nlp.stanford.edu/software/stanford-dependencies.shtml

toolkit. The parser generates a Dependency Tree for the input text. Figure 3.4 shows the dependency trees generated by the parser for the sentences *S*[*No dorsal fin*] and *S*[*34-37 scales in lateral line*].

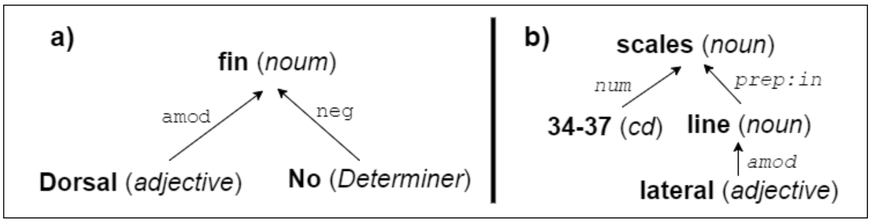


Figure 3.4: Dependency trees of the sentences (a) S[*No dorsal fin*] and (b) S[*34-37 scales in lateral line*].

The STDP contains approximately 50 grammatical dependencies [29]. We selected a subset of these relations that reflects the written characteristics to be analized. For example, *amod* is a dependency that has an adjective qualifying a noun. Figure 3.4.a presents the term "dorsal", which refers to an adjective, while "fin" refers to a noun. The exploitation of these grammatical dependencies reflects the first observation of the typical way to write phenotypic descriptions. Some of the relations cited along this text are detailed below:

- **amod:** an adjective modifying the meaning of the noun, *e.g.*, $amod(fin, dorsal)$;
- **nn:** a noun that serves to modify a head noun;
- **nsubj:** a nominal subject is a noun which is the syntactic subject of a clause;
- **dobj:** the direct object of a verb is a noun which is the object of a verb;
- **prep:** a word introducing a finite clause subordinate to another clause, *e.g.,* $prep(20, to)$;
- **conj:** a conjunct is a relation between two elements connected by a coordinating conjunction, such as "and", "or", *etc.*, *e.g.,* $conj\_and(head, body)$;
- **number:** it specifies a compound number, in a specialized kind of multi-word expression, *e.g.*, $number(20, 15)$;
- **dep:** a dependency is labeled as *dep* when the STDP is unable to determine a more precise dependency relation between two words. This may be a result of an unusual grammatical construction; a limitation in the Stanford Dependency conversion software; a parser error, *etc.*

In the Dependency Tree, Function Words are represented as edges (*e.g.*, the word "*in*" in Figure 3.4.b) instead of vertexes. Our match algorithm does not consider the edge labels. Therefore, these Function Words are ignored, which is conform the second writing characteristic of phenotype descriptions, observed by Cui [12] since it does not carry relevant meaning.

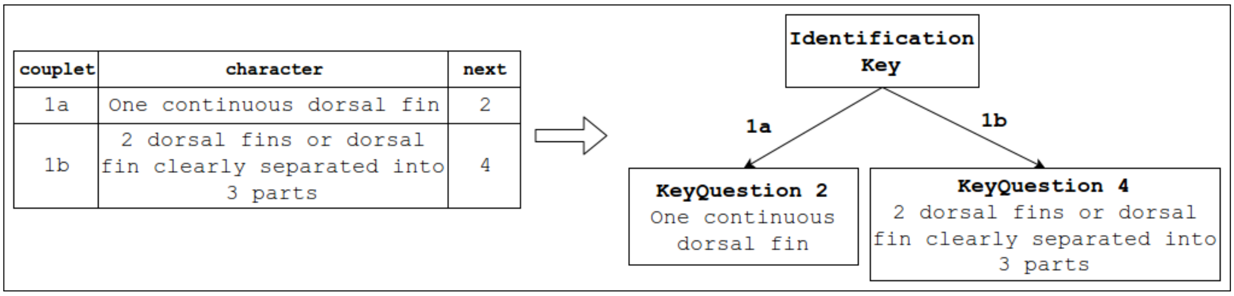
In order to obtain a semantic description of EQs, the Dependency Relations are matched with domain ontologies. We used ontologies widely adopted by the community: (1) *Teleost Anatomy Ontology* (TAO) [14] – an ontology that formalizes the knowledge about teleostean fishes anatomy; (2) *Phenotypic Quality Ontology* (PATO) – an ontology that defines *Qualities* to be related to *Entities* and their respective values.

Domain ontologies aid the recognition of implicit phenotype characters, such as those mentioned in the third item of the list regarding writing characteristics. PATO states *big* as an instance of *size*, as well as *black*, as an instance of *color*, even though the original text *S*[*Black and big eyes*] does not explicitly mention the words *size* or *color*.

We further detailed the algorithms that implement the activities of Step 1.

3.1.1 Algorithms in Step 1

We started transforming the relational data contained in *FishBase* into a graph representation, as shows Figure 3.5. It shows a fragment of the IK previously presented in Figure 1.1. In the graph representation, nodes are KQs containing descriptive texts and edges are links among KQs according to answer choices.



couplet	character	next
1a	One continuous dorsal fin	2
1b	2 dorsal fins or dorsal fin clearly separated into 3 parts	4

Identification Key
1a
1b
KeyQuestion 2
One continuous dorsal fin
KeyQuestion 4
2 dorsal fins or dorsal fin clearly separated into 3 parts

Figure 3.5: Transforming the relational data into a graph representation.

Each *KeyQuestion* is then processed by Algorithm 1, which in turn calls Algorithm 2. These algorithms are presented in order to facilitate the explanation of their characteristics and execution. Details concerning tests to handle exception failures (*e.g.*, *null*, empty, *etc.*) are omitted here for simplicity, but they are considered in the prototype implementation.

3.1.1.1 Algorithm 1: Entity-Quality recognition

Algorithm 1 processes the natural language sentence text in order to recognize EQ elements and link the produced subgraph to the original *KeyQuestion*. The method *getDependencyTree()* in Line 2 applies the parser over the sentence text.

The loop starting at Line 3 recognizes the *Entities*, matching each vertex of the dependency tree with the TAO ontology through the Algorithm 2, whose return is stored in the variable *entity* (Line 4). If this matching process is successful (test of Line 5), then the *Entity* is added to the graph (Line 6), through the link to the KQ node.

Algorithm 1

```
INPUT:    KeyQuestion                               ▷ node containing text sentence
          KeyQuestion
OUTPUT:   List of Entities               ▷ Entity nodes connected to the KeyQuestion
          List of Qualities                   ▷ Quality nodes conneted to each Entity
 1: procedure STEP1($KeyQuestion$)
 2:     $dependencyTree \leftarrow KeyQuestion$.getDependencyTree()
 3:     for each $vertex \in dependencyTree$ do
 4:         $entity \leftarrow$ matchByExpansion($vertex, TAO$)
 5:         if $entity \neq Null$ then
 6:             $KeyQuestion$.attach($entity$)
 7:
 8:     for each $entity \in KeyQuestion$.getEntities() do
 9:         $candidateQualities \leftarrow entity$.relatedVertexes()
10:         for each $vertex \in candidateQualities$ do
11:             $quality \leftarrow$ matchByExpansion($vertex, PATO$)
12:             if $quality \neq Null$ then
13:                 $entity$.attach($quality$)
```

Part (a) of Figure 3.6 illustrates the execution of Algorithm 1 over the sentence S[*No dorsal fin*]. The E[*dorsal fin*] is recognized in the execution until Line 6. The gray region in Figure 3.6.a emphasizes the part of the dependency tree which was recognized as *Entity*. From this point, the algorithm recognizes the *Qualities* related to *Entities* previously recognized.

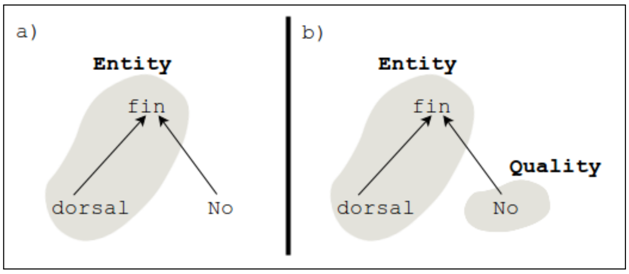


Figure 3.6: Example of Algorithm 1 execution.

The loop initiated at Line 8, iterates over each *Entity* detected in the previous loop. Line 9 gets the vertexes related to each recognized *Entity*. Each vertex is a possible *Quality* candidate (*e.g.,* in our example, V[*No*]). It is also recognized by Algorithm 2, but using the *PATO* ontology (Line 11). Figure 3.6.b shows the result of the execution regarding Lines 8-13.

Special Dependency Relations. Algorithm 1 was described in general terms. However, it is important to detail its approach to handle two special dependency relations: *number* and *conj*. Figure 3.7 presents the dependency trees of two sentences S[*vertebrae 119 to 132.*] and S[*dorsal and anal fins.*]

The relation *number* connects two extremes of a range of values. It can be mapped to a *Quality* modifying an *Entity* in a range of possible quantities. For example, when the

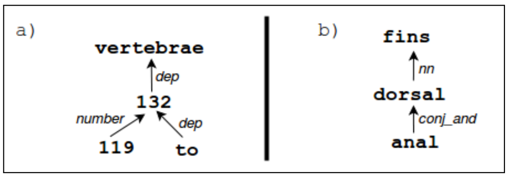


Figure 3.7: Special Relations. (a) S[*vertebrae 119 to 132*]. (b) S[*dorsal and anal fin*].

relation $dep(vertebrae, 132)$ is processed, the vertex $V[132]$ is transformed into $V[count]$, since the *PATO* ontology defines a class *Counter*. In Line 9, instead of sending the vertex V[*count: 132*] to Algorithm 2, we previously group the vertexes $V[119]$, $V[132]$ and $V[to]$ into a single vertex V[*count:119 to 132*], resulting in the *Quality* Q[*count: 119 to 132*]. This treatment might further improve the recognition of Entities.

The relation *conj* can represent a group of two *Entities*. In Figure 3.7.b, the relation *conj(dorsal, anal)* groups both terms, which are jointly related to $V[fins]$. Therefore, they are converted into two vertexes V[*dorsal fins*] and V[*anal fins*], since $V[dorsal]$ and $V[anal]$ are related to $V[fins]$, even not directly in the second case.

3.1.1.2 Algorithm 2: Match in Ontology by Expansion

Algorithm 2 (invoked at Lines 4 and 11 of Algorithm 1) refers to a recursive function that performs a search over a domain ontology (TAO or PATO: *Entity* or *Quality*, respectively). The function looks for a concept in the ontology that has the highest similarity with a given subgraph. Similarity refers to which degree ($similarity \in [0, 1]$) an existing ontology concept is similar to the terms of a given subgraph. In each recursive call, the algorithm visits all the vertexes connected to the subgraph (neighbor vertexes) and try to match the combination against the ontology.

Algorithm 2

```
INPUT:   subgraph          ▷ a subgraph of the dependency tree being analyzed
         ontology          ▷ ontology to be queried
OUTPUT:  concept           ▷ concept node
function matchByExpansion($subgraph, ontology$)
    return matchByExpansion($vertexes, Null, ontology$)

function matchByExpansion($subGraph, bestConcept, ontology$)
    $concept$ ← ontology.getClosestConcept($subgraph$)

    if $bestConcept == Null \,||\, concept.similarity > bestConcept.similarity$ then
        $bestConcept \leftarrow concept$

    $neighborVertexes \leftarrow subgraph.getNeighborVertexes()$
    for each $vertex \in neighborVertexes$ do
        $newSubgraph \leftarrow subgraph.connected(vertex)$
        $bestConcept$ ← matchByExpansion($newSubgraph, bestConcept, ontology$)
    return $bestConcept$
```

Figure 3.8 illustrates the execution of Algorithm 2 looking for an *Entity*. The algorithm receives the parameters: "*fin*" as subgraph and TAO as ontology. At each iteration, the algorithm expands the subgraph adding vertexes connected to the current subgraph (neighbors). After recursively traversing all the Dependency Tree, it looks for the most similar concept $E[\textit{dorsal fin}]$.

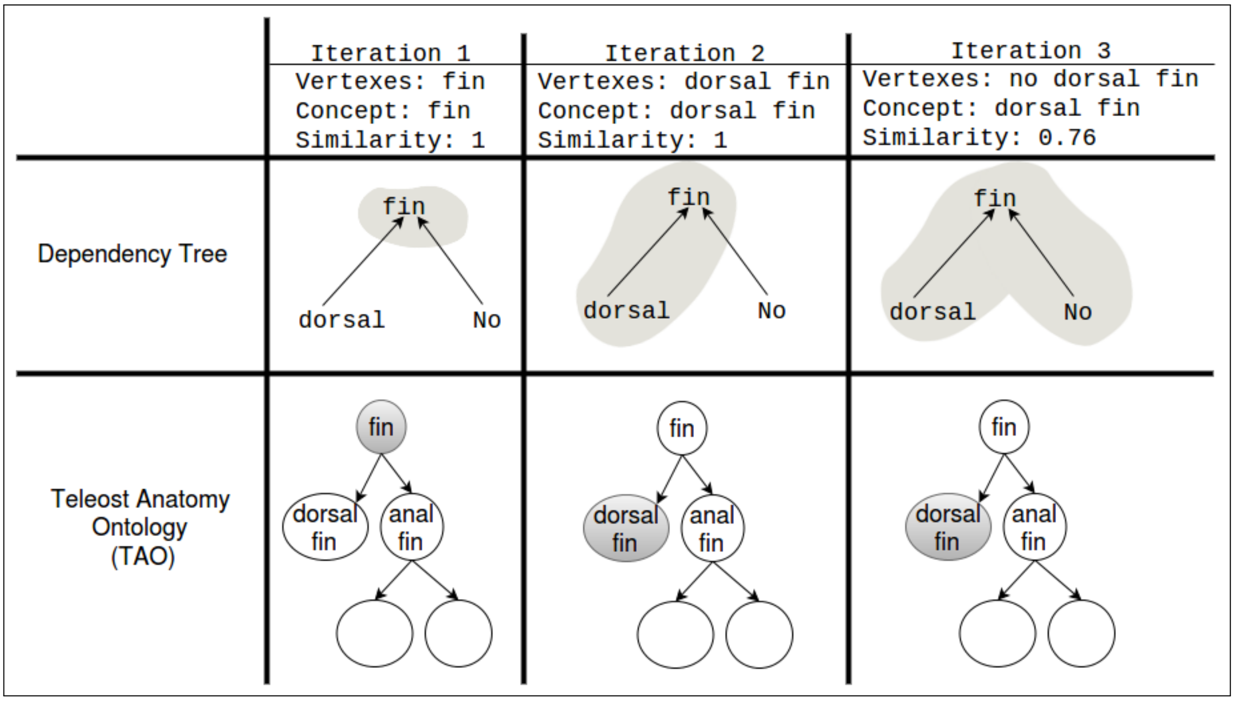


Figure 3.8: Match by expansion algorithm over the sentence $S[\textit{No dorsal fin}]$.

At Line 10 the method retrieves all vertexes connected to the subgraph (neighbors). It recursively tries to match all the combinations of the subgraph with each neighbor (Lines 10, 11, and 12). For example, at iteration 1, the neighbors are: "*dorsal*" and "*no*".

The recursive calls finalize when the dependency tree ends. Then, the method returns to the Algorithm 1 with the most similar concept.

Technically, method *getClosestConcept* (Lines 5 of Algorithm 2) perform the *SPARQL* Query shown in Listing 3.1, over the given ontology. It looks for the term and synonyms. The Levenshtein function [26] – which computes the similarity by edit-distance between a given query term and the concepts inside of the ontology – is used to rank the terms according to their similarity. The query returns only the first one (best match).

```
SELECT DISTINCT ?resource
WHERE {
  {?resource rdfs:label ?label . }
  UNION {?resource obo:hasExactSynonym ?label . }
  UNION {?resource obo:hasRelatedSynonym ?label . }
} ORDER BY DESC(f:LevenshteinFilter(?label, value)) LIMIT 1
```

Listing 3.1: SPARQL query to retrieve the most similar concept.

Figure 3.9 presents two elements recognized by the matching algorithm. The subgraph containing $V[fin,\ dorsal]$ is matched with $E[Dorsal\ fin]$ (a concept of TAO), while the subgraph containing $V[no]$ is matched with $Q[absent]$ (a concept of PATO). In the latter case, although the resource `label` and the vertex have low similarity, the similarity between the vertex and the value of the property `hasExactSynonym` is high, which is considered by the *Union* clauses in Listing 3.1. As the two recognized elements are linked by a relation, they are considered as an EQ pair.

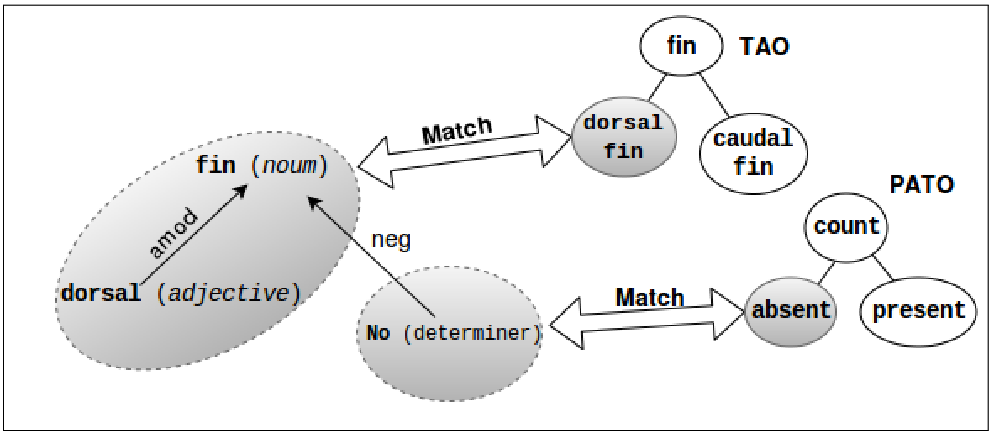


Figure 3.9: EQ pair recognized by Step 1.

Figure 3.10 presents the result of Step 1: a graph where each *Key Question* is connected to the respective recognized *Entities* and *Qualities*, *e.g.*, the nodes $E[dorsal\ fin]$ and $Q[absent]$.

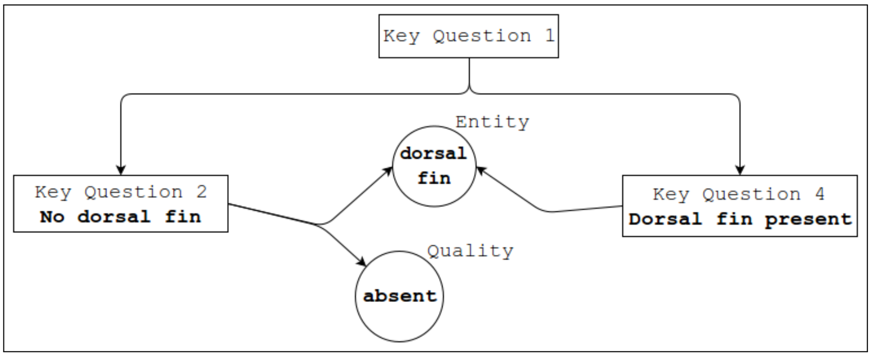


Figure 3.10: Step 1 output.

It is possible to observe in Figure 3.10 that the method fails at recognizing $Q[present]$ in the sentence $S[Dorsal\ fin\ present]$. This failure is due to the violation of the English language rules in the sentence formulation. There are other cases where Step 1 fails, then the following Step 2 aims to treat these cases.

In addition, differences in language usage in distinct domains might significantly impact the performance of NLP tools. Therefore, the application of a generic domain parser over a specific domain sentence can produce erroneous results on morphological descriptions. This is the case here since there is no dependency parser specifically developed to handle morphological description texts. Step 2 of our method aims to treat these type of failures, improving the results.

3.2 Step 2: Exploring the Structure of Identification Keys

This step explores the structure of IKs to enrich the output graph from Step 1. We assume that the correlation between distinct descriptions might be useful in the extraction of additional EQs. Such correlation is an intrinsic characteristic of IKs, as a result of their organizational structure. This step is based on the previously mentioned Assumption 2: **The way in which a set of phenotype descriptions is organized and structured holds implicit relations that can be exploited to improve the extraction of EQ statements**.

We believe that the principles behind this work could be generalized to other fields in the future. An organizational structure, as we exploit in the IKs, could also be the sessions of a technical report, the structure of legal documents with juridical rules, the layout of a Web site, *etc.* Wong *et al.* [45] indicate that such noncontent cues may be used to support information extraction tasks. This perspective opens a future wider application scenario for our technique.

IKs are structured in a tree format, in which the alternatives of a given KQ are its sibling nodes containing complementary alternative sentences. This structure offers clues about its content, from which we consider the following characteristics:

(a) Alternatives of a KQ frequently refer to the same *Entities.* In our previous example, both sibling sentences S_1[*No dorsal fin*] and S_2[*Dorsal fin present*] refer to the same anatomical character E[*dorsal fin*];

(b) Alternatives of a KQ are frequently complementary, in the sense that they assign complementary states to the described *Entity.* In the same previous example, the *Qualities* Q_1[*absent*] and Q_2[*present*], assigned to the *Entity* E[*dorsal fin*], are opposites, encompassing its possible state values.

In summary, we assume that if an EQ pair is identified in a KQ, it is very likely that the sibling KQs must refer to the same *Entity*, but potentially using complementary *Quality* terms to modify the *Entities.* The challenge here is to verify if the sibling nodes hold this property

Therefore, we developed an algorithm that measures the similarity between two sentence pieces. It is based on the general principle of Paraphrase Recognition, which is a process to judge if two different sentences convey the same aspect or the same information. Androutsopoulos and Malakasiotis [3] present a survey regarding Paraphrase Recognition techniques. There are techniques that exploit the dependency tree to measure the similarity between the sentences. In general, they assume that if there is a value above a given threshold, the involved sentences are considered paraphrases.

Usually, Paraphrases Recognition algorithms compare the whole trees [3]. We have adapted the principle of Paraphrases Recognition to the problem of recognizing complementary sentences in an IK.

As an example, Figure 3.11 shows dependency trees of two different sentences that are paraphrases. The tree format enables to analyze the high similarity between the structures despite their differences in the word order.

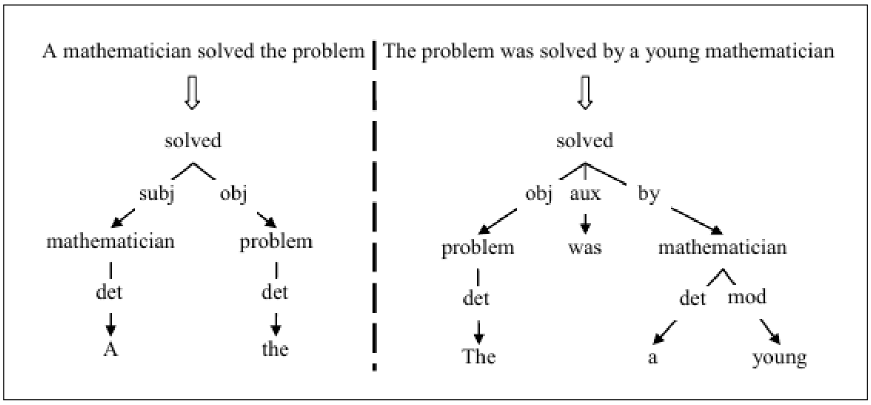


Figure 3.11: An example of paraphrase. Source: Androutsopoulos *et al.* [3]

3.2.1 Algorithm for Detecting Complementary Qualities

Step 2 acts in cases where Step 1 was successful in one sentence, but failed in recognizing EQ statements in its siblings. It determines if these sentences have complementary *Qualities* for the same *Entities*. It measures the similarity between the subtrees comparing each edge inside them. We aim to verify if they refer to the same *Entity* with complementary *Qualities*, based on a settled threshold.

Figure 3.12 illustrates the input elements in the algorithm. The algorithm receives a pair of *KeyQuestions*: KQ_{main} and $KQ_{sibling}$. Inside the KQ nodes, there are the Dependency Trees DT_{main} and $DT_{sibling}$ of the respectively sentences S_{main} and $S_{sibling}$, which were processed in Step 1.

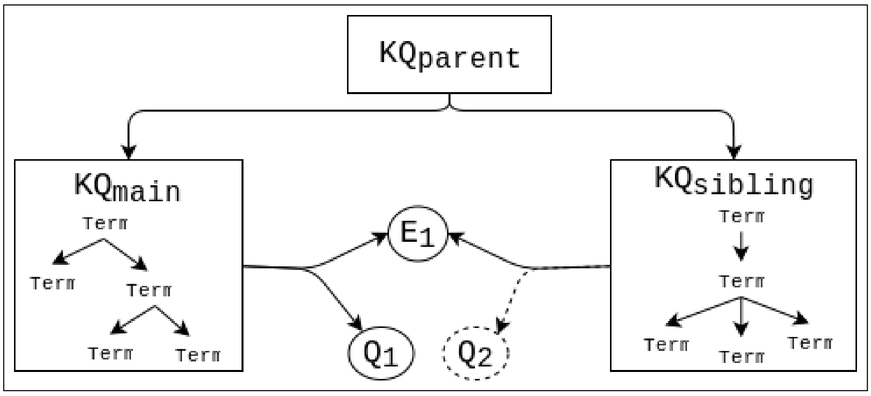


Figure 3.12: A generic example of Step 2 input.

In the example of Figure 3.12, the KQ_{main} has a link to an EQ pair E_1Q_1 and the $KQ_{sibling}$ has a link to the same *Entity* E_1, but it lacks the *Quality* part (*cf.* Q_2 dashed in Figure 3.12). The Q_2 is the *Quality* part to be inferred by the algorithm.

Algorithm 3 iterates over each EQ pair (Lines 2 to 18), linked to the KQ_{main} (E_1 and Q_1 in Figure 3.12). Lines 3 and 6 get the subtrees $Entity_{main\_subtree}$ and $Quality_{main\_subtree}$

Algorithm 3

```
INPUT:  $KQ_{main}$        ▷ Key Question with an EQ pair connected
        $KQ_{sibling}$     ▷ Sibling Key Question

 1: procedure STEP2($KQ_{main}$, $KQ_{sibling}$)
 2:     for each $E_1$ connected to $KQ_{main}$ do
 3:         $Entity_{main\_subtree} \leftarrow$ getSubTree($E_1$)
 4:
 5:         for each $Q_1$ connected to $KQ_{main}$ do
 6:             $Quality_{main\_subtree} \leftarrow$ getSubTree($Q_1$)
 7:             $edge_{main\_eq} \leftarrow$ getEdge($Entity_{main\_subtree}$, $Quality_{main\_subtree}$)
 8:
 9:             $Entity_{sibling\_subtree} \leftarrow$ getSubTree($E_1$)
10:             for each $edge_{sibling\_eq}$ connected to $Entity_{sibling\_subtree}$ do
11:                 $similarSubTrees_{list}$.add(simFunction($edge_{main\_eq}$, $edge_{sibling\_eq}$))
12:
13:             if $similarSubTrees_{list}$.getHigherSimilarity() $>$ $threshold$ then
14:                 $Q_{sibling\_subtree} \leftarrow$ $similarSubTrees_{list}$.getMostSimilarSubTree()
15:                 $Q_2 \leftarrow$ matchByExpansion($Q_{sibling\_subtree}$, $PATO$)
16:
17:                 if $Q_2 \neq Null$ then
18:                     $KQ_{sibling}$.attach($Q_2$)
```

(from the entire tree DT_{main} contained in KQ_{main}). These subtrees contain the terms that are part of the *Entity* E_1 and *Quality* Q_1, respectively. Figure 3.13 exemplifies these subtrees highlighting $Entity_{main\_subtree}$ and $Quality_{main\_subtree}$ inside the DT_{main}.

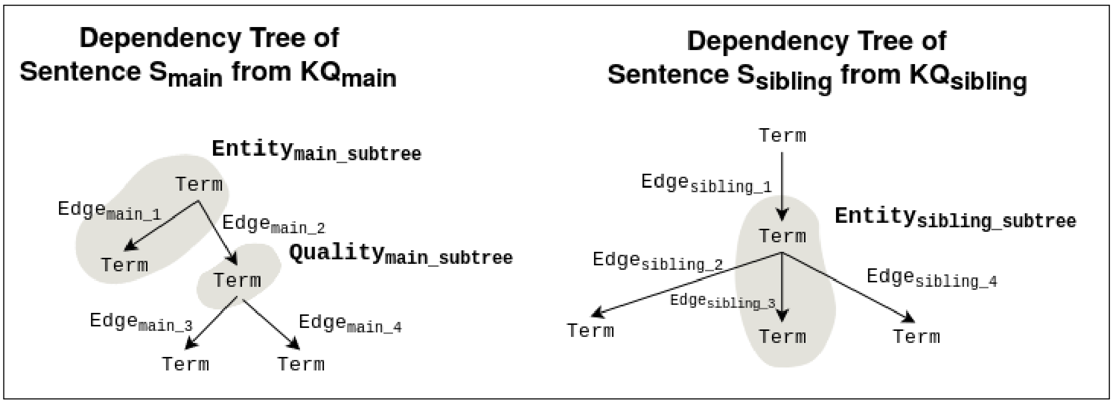


Figure 3.13: Generic example of dependency trees of two sentences. *Entity* and *Qualities* recognized in the previous Step 1 are highlighted.

Line 7 gets the $edge_{main\_eq}$, which links $Entity_{main\_subtree}$ to $Quality_{main\_subtree}$ (in Figure 3.13 is the $Edge_{main\_2}$). The algorithm, via the loop at Line 10, compares this edge with all edges related to the subtree $Entity_{sibling\_subtree}$ in the $DT_{sibling}$ (the edges $\langle Edge_{sibling\_1}$, $Edge_{sibling\_2}$,$Edge_{sibling\_3}$, $Edge_{sibling\_4}\rangle$ in Figure 3.13) to decide which one connects to the complementary Q_2.

The loop at Line 10 iterates over each $edge_{sibling\_eq}$ related to the $Entity_{sibling\_subtree}$

(got at Line 9). At each iteration (Line 11), the algorithm computes the similarity between the current $edge_{sibling\_eq}$ with the $edge_{main\_eq}$ through the method $simFunction()$. The similarity computation between the edges takes into account the following parameters:

(a) Directions of the dependency relations $edge_{sibling\_n}$ and $edge_{main\_2}$;

(b) Grammatical class of Q_1 and Q_2;

(c) Types of the dependency relations of $edge_{sibling\_eq}$ and $edge_{main\_eq}$;

(d) Antonymy between Q_1 and Q_2 (the algorithm explores the *WordNet* lexical database [32] to check if two words are antonyms).

These parameters represent to which extent one edge is similar to another. To calculate the degree of similarity, each parameter contributes with a pre-defined value: $v_a = 0.25$; $v_b = 0.50$; $v_c = 0.75$; $v_d = 1$.

We have chose these parameters and estimated their corresponding values based on empirical observations regarding their relevance in Dependency Tree elements (edges and vertexes) concerning phenotype description sentences. For example, we noted that a pair of edges having the same direction is important, but it is less important than the fact that the *Qualities* have antonyms terms since the algorithm is looking for opposite *Qualities.* These parameters and their values can be adapted to the execution of the algorithm in other scenarios.

The similarity between each pair of edges is calculated through a summation of those parameters. At Line 11, the algorithm adds this similarity to the $similarSubTrees_{list}$. Line 13 tests if the subtree inside $similarSubTrees_{list}$ with the highest similarity value is equal or higher than a determined *threshold.* In the conducted experiments, we assigned the $threshold = 0.75$ to avoid retrieving edges with low similarity values.

The *threshold* value can be modified and it affects the behaviour of the algorithm. A high *threshold* value enables to recognize more *Qualities*, but it can increase the rate of false positives. On the other hand, a low value can decrease the number of recognized *Qualities*, but it increases the rate of correct elements. The values of each parameter and threshold have been empirically determined by experimental analyses.

If the algorithm is able to select an edge with the highest similarity value, then Line 14 attributes this subtree to the $Q_{sibling\_subtree}$. Afterward, Line 15 calls the function $matchByExpansion$ defined in Algorithm 2 to recursively discover the subgraph inside the selected subtree which matches a *Quality* in $PATO$ ontology. If the match is successful (Line 17), the *Quality* is attached to the sibling KQ (Line 18), *i.e.*, the Q_2 illustrated in Figure 3.12.

Figure 3.14 illustrates an example in which Step 2 will act. It goes back to Figure 3.10 showing the output of Step 1 executed over two sibling KQs, in which $Q_1[absent]$ from KQ_2 was recognized, while $Q[present]$ from KQ_4 was not, *i.e.*, the Step 1 failed to recognize it. In the following, we use this example to illustrate the execution of the Algorithm 3.

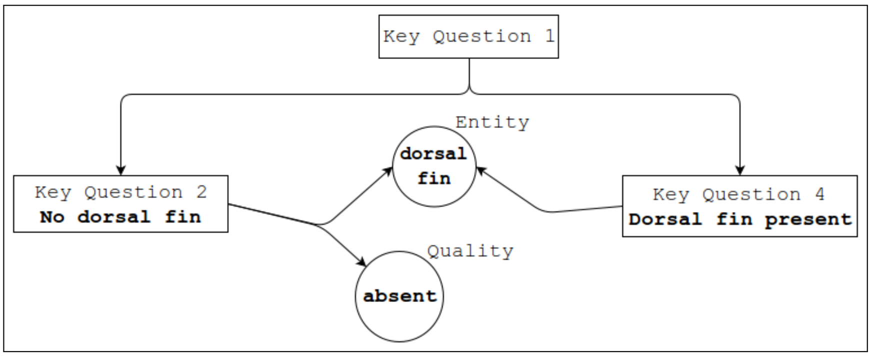


Figure 3.14: Step 2 input.

Figure 3.15.a shows the dependency trees of the sentences $S_{main}[No\ dorsal\ fin]$ and $S_{sibling}[Dorsal\ fin\ present]$ extracted from two input KQs, in the beginning of the execution of Step 1. Figure 3.15.b presents the same dependency trees after Step 1 execution, highlighting the subtrees containing the terms that were recognized as *Entities* and *Qualities*. Step 2 gets the DT_{main} and $DT_{sibling}$ in this format. We further detail the execution of Algorithm 3.

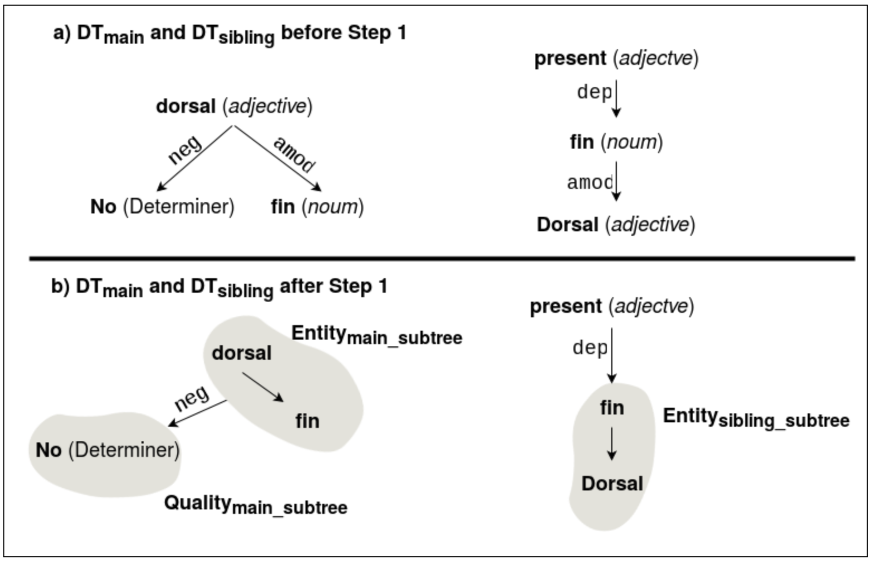


Figure 3.15: Dependency tree regarding two couple KQs.

From DT_{main}, Line 7 gets $edge_{main\_eq}[neg(dorsal, no)]$ connecting the subtrees $Entity_{main\_subtree}[Dorsal\ fin]$ to $Quality_{main\_subtree}[absent]$ (got in Lines 3 and 6, respectively). From $DT_{sibling}$, Line 9 gets $Entity_{sibling\_subtree}[dorsal\ fin]$. Line 10 iterates over each $edge_{sibling\_eq}$ connected to the subtree $Entitty_{sibling\_subtree}$. In this example, $edge_{sibling\_eq}[dep(present,\ fin)]$ is the only edge connected to the subtree. We chose this example to facilitate the visualization of the algorithm execution, but it could be connected to more edges, depending on the size and structure of the sentences.

Line 11 compares the $edge_{main\_eq}[neg(dorsal,\ no)]$ with $edge_{sibling\_eq}[dep(present, fin)]$, through the $simFuntion()$ as follows:

(a) They have different directions: $edge_{main\_eq}[neg]$ outcomes from $Entity_{main\_subtree}$, while $edge_{sibling\_eq}[dep]$ incomes in $Entity_{sibling\_subtree}$;

(b) The $Quality_{main\_subtree}[no]$ and the possible $Quality$ part from $edge_{sibling\_eq}[dep]$ do not have the same grammatical classes, *no* is a *determiner* and *present* is an *adjective*;

(c) $edge_{main\_eq}[neg]$ and $edge_{sibling\_eq}[dep]$ have different types of dependency relation, the first one is a *neg* and the second is a *dep*;

(d) The $Quality_{main\_subtree}[no]$ and the possible $Quality$ part *present* from $edge_{sibling\_eq}[dep]$ are antonyms. Then, it sums 0.75 to the similarity between the edges.

Since the parameter d holds, the result of Equation **??** remains 0.75, which is equal to the fixed $threshold = 0.75$ (Line 13). Therefore, Line 14 gets edge $Q_{sibling\_subtree}[dep(present, fin)]$ as the most similar. Line 15 calls the function $matchByExpansion()$, which returns the $Quality$ $Q_2[present]$ according to the ontology. Finally, Line 18 attaches Q_2 to $KQ_{sibling}$.

At this stage project, our Algorithm 3 is limited in the comparison of edges: $edge_{main\_eq}$ and $edge_{sibling\_eq}$. We plan to expand such technique to compute the comparison between entire sub-trees representing the *Quality* parts.

Figure 3.16 brings the same example of Figure 3.3 showing the Step 2 output to the given example. Note that, compared to Figure 3.14, the $Q[present]$ was inserted as a new node in the graph according to the result of the algorithm.

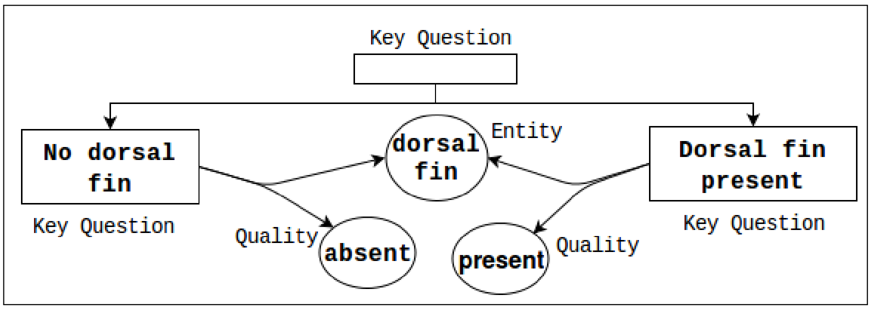


Figure 3.16: Step 2 output.

3.3 Summary

This chapter presented an original method to recognize EQ elements from semi-structured IKs. The proposed methods have two steps which explore different aspects of the IK to refine the extraction process. Step 1 explores the typical writing characteristics of phenotype descriptions and considers a *Typed Dependencies Parser* to detect relations between sentence elements. In our method, the dependency relations returned by the TDP were matched with domain ontologies to obtain a semantic representation of EQs. We have defined algorithms to detect the EQs and to match them with domain ontologies by a recursive expansion process. Step 2 explored the structure of IKs to enrich the graph and to overcome limitations of dependency parsers. The algorithm in Step 2 determined if two given couple sentences have complementary EQ pairs.

This investigation resulted in the development of a software prototype implementing the proposed approach. The next chapter describes experiments to evaluate the quality of the EQ recognition applying our proposal.

Chapter 4

Experiments and Evaluation

This chapter reports experimental results of this investigation. We rely on the *FishBase* database to conduct the proposed assessments. Section 4.1 presents an evaluation to assess the viability of our extraction method. The objective is to investigate the effectiveness of the approach considering a gold standard dataset and traditional metrics.

The initial motivation for this research was to obtain a knowledge network correlating and integrating several elements of phenotype description. To this purpose, Section 4.2 presents a knowledge network evaluation that generated and integrated species data with the recognized EQs from *FishBase.* The objective is to conduct practical applications and analysis over the generated network.

4.1 Empirical Assessments

The first assessment (Section 4.1.1) analyses the results regarding basic metrics, to provide an overview of the recognition performance and its viability. The Gold Standard-based assessments (Section 4.1.2) aim to examine the accuracy of the recognition method against a set of expected outcomes. Section 4.1.3 discusses the obtained findings.

4.1.1 First Approach Assessment

We applied the proposed method to all *Identification Keys* (IKs) of *FishBase* to observe its viability when extracting phenotypes in *Entity-Quality* (EQ) format. This experiment was performed over the total of 1,659 IKs, containing 25,542 Key Questions (KQs). We considered the following metrics:

- **Amount:** calculates how many *Entities* and *Qualities* were recognized;
- **Ratio:** presents the average of extracted EQs from each KQ;
- **Coverage:** shows the rate of KQs containing at least one element extracted, varying from 0 to 1.

These metrics were computed separately for Step 1 and Step 2. Therefore, it is possible to observe the differences and the impact of the assumptions underlying the Observations

1 and 2. Table 4.1 presents the obtained results for extraction of *Entities* (alone) and EQs.

Table 4.1: Results concerning Amount, Ratio, and Coverage.

	Amount		**Ratio**		**Coverage**	
	Step 1	**Step 2**	**Step 1**	**Step 2**	**Step 1**	**Step 2**
Entity	30,747	41,611	1.2	1.62	0.61	0.7
EQ	15,239	17,267	0.6	0.67	0.36	0.4

Results reveal that the method allows extracting the phenotypes. The amount of extracted elements increases with Step 2, as expected. It keeps consistent with the metrics of Ratio and Coverage, with better results in Step 2.

There is still space for further improvements. Several research efforts can be devoted to refine Steps 1 and 2. In Step 1, improvements can involve refining the match algorithm, considering not only the structure of the *Dependency Tree*, but also the semantic of the relations among the terms. It is possible to explore other types of relations returned by the NLP parser. Furthermore, Step 2 can be improved by exploring additional characteristics of the IK's structure. For instance, Algorithm 3 can take advantage of other branches of the tree beyond the siblings.

Although the results of this evaluation show the initial viability of our approach, they do not consider its correctness, *i.e.*, how precise is the method. Such evaluation is showed in the following subsection.

4.1.2 Gold Standard-based Assessment

The quality of the recognition and extraction of elements in natural language texts, *i.e.*, entities or relations, can be evaluated by several mechanisms. The most common considers a standard evaluation set generated by either a group of specialists in the domain, or an organizing committee of a competition. A standard evaluation set contains fragments of texts highlighting the elements that are supposed to be recognized. Such kind of evaluation is suitable when there is a mature developed community acting in the area of interest.

However, there is still no standard evaluation set of morphological descriptions in the context that we are working: *Entity* and *Quality* linked in an EQ pair. Therefore, this investigation involved the creation of an evaluation dataset to assess the performance of our method. This dataset has the original sentence descriptions where the EQ elements are annotated.

Unlike the previous evaluation (4.1.1), this one shall not be performed over the total dataset, since it is necessary to manually annotate the sentences. A set of 100 KQs were manually annotated to act as a Gold Standard, from the total of 25,542 KQs from *FishBase*.

Figure 4.1 shows five examples of sentences in our evaluation dataset. The words in bold compose *Entities*, and words in italic compose *Qualities*, while the boxes represent EQ pairs.

1. **Lips** *not fringed*; **mouth** *horizontal*.
2. *No dark longitudinal* **stripes** on **head** and **body**.
3. The *two* **light organs** near the **tail**; clearly separated from the rest of the **light organs**.
4. Total **vertebrae** *119 to 132*.
5. *Scattered* **breast melanophores** (Fuiman et al., 1983). Pteronotropis hubbsi can also be distinguished from Notropis chalybaeus by the presence of *two* **caudal spots**, *one large* **spot** centered at the base of the **caudal fin** below the *flexed* **notochord** and a *smaller* **spot** located dorsally above it, and by the presence of *9* **dorsal rays** in late metalarvae. Notropis chalybaeus has a *single* **caudal spot** in which no part extends above the **notochord** and *8* **dorsal rays** (Marshall, 1947).

Figure 4.1: Examples of Standard Gold sentences.

Several criteria were explored to create the Gold Standard. First, we considered only Simple EQs, *i.e.,* those composed strictly by one *Entity* and one *Quality*, such as in the second sentence in Figure 4.1: $E[stripes]Q[no]$, $E[stripes]Q[dark]$ and $E[stripes]Q[longitudinal]$. To save space, we group them as follows: $E[stripes]Q[no]Q[dark]Q[longitudinal]$.

We ignored complex EQs, *i.e.,* those composed by complex *Qualities*, which recursively contain *Qualities* linked to other *Entities.* For example, Sentence 4 in Figure 4.1 has a complex EQ formed by $E[spot]$ $Q[centered\ at\ the\ base\ of]$ $E[caudal\ fin]$. This kind of phenotype construction requires further efforts and expertise to produce annotation. In particular, complex EQs are not treated by our approach and to avoid misinterpretations in the numerical evaluation, they are not computed.

We report the results considering exact matches (more precise) (*cf.* Section 4.1.2.1) and also partial matches (*cf.* Section 4.1.2.2).

4.1.2.1 Exact Matches Recognition Analysis

We applied our method to each annotated KQ. We compared the EQs recognized by our method with the annotations of the Gold Standard. The comparison considers four indicators:

- **True Positive (TP):** elements correctly identified. For Example: our method identified in Sentence 1 the following EQs: $E[lips]Q[not\ fringed]$; $E[mouth]Q[horizontal]$. These elements were actually annotated in Sentence 1 of the Gold Standard;

- **False Positive (FP):** An expression recognized by the method as a phenotype, which does not appear as such in the Gold Standard. Example: in Sentence 4, our approach recognized *E*[*vertebrae*]*Q*[*132*], which is a *Quality* that slightly differs from the expected one;

- **False Negative (FN):** those phenotypes annotated in the Gold standard that were not detected by the method. Example: *E*[*breast melanophores*]*Q*[*Scattered*] should be identified in Sentence 5 and we failed in recognizing it.

The computation of TP, FP and FN allows calculating the following traditional measures:

$$\mathit{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

$$\mathit{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

$$\mathit{F\text{-}measure} = \frac{2 * \mathit{Precision} * \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} \tag{4.3}$$

Precision stands for the percentage of elements detected by the algorithms that are correct. Recall refers to the percentage of elements present in the Standard Gold Set found by the algorithms. These measures are complementary indicating false alarms and miss errors, respectively. F-measure refers to a harmonic mean of precision and recall.

Our first analysis considered exact matches in the computation of the measures. In this sense, a EQ recognized by our method should be strictly equal to the correspondent in the Gold Standard Set to be computed as a TP. For instance, *E*[*mouth*]*Q*[*horizontal*] in the example of Sentence 1.

Table 4.2 presents the obtained results of Precision, Recall and F-measure. The column "EQ pair" compute the recognition of complete *Entity-Quality* pairs and the column "Entity" computes the recognition of *Entities* alone without related *Qualities*.

Table 4.2: Results concerning only Perfect Matches.

Elements Measures	**EQ pair**	**Entity**
Recall	0,39	0,69
Precision	0,75	0,85
F-measure	0,51	0,76

Results indicate better performance with the recognition of isolated *Entities*, was expected. While the Precision of the "EQ pair" is slightly worst when compared to "Entity", the Recall further decreases. This result is impacted by EQs containing ranges of values in the *Quality* part. For example, the *S*[*total vertebrae 119 to 132*] has the expected result *EQ*[*E*[*vertebrae*] *Q*[*119 to 132*]] in the Gold Standard. Our method is only able to recognize part of the *Quality*, yielding $Q[132]$. This might not be considered a totally wrong result, but the exact match analysis is unable to consider it. The next section presents a more flexible partial match analysis to take such cases into account.

4.1.2.2 Partial Matches Recognition Analysis

Partial matches occur when a recognized element intersects with an expected element, but their boundaries do not perfectly match. For instance, $E[vertebrae]Q[\mathit{132}]$ is a partial match with $E[vertebrae]Q[\mathit{119\ to\ 132}]$. Partial matches are significant pieces of information, although they are not exactly the expected results. Atdag and Labatut [5] propose a set of additional counts to consider in the measures:

- **Partial Matches (PM):** when the recognized element contains only a part of the expected one;
- **Complete Miss (CM):** expected elements not detected by the phenotype recognition;
- **Wrong Hit (WH)**: recognized elements that do not correspond to any expected element;
- **Full Match (FM):** equivalent to the *True Positive.*

These indicators allow redefining in a smoothly way the *False Negative* as *Partial Match*+ *Complete Miss* and *False Positive* as *Partial Match* + *Wrong Hit.* Atdag and Labatut [5] propose an adaptation of *Precision* and *Recall*:

$$\mathit{Partial\ Precision} = \frac{\mathit{Partial\ Match}}{\mathit{Full\ Match} + \mathit{Partial\ Match} + \mathit{Wrong\ Hit}} \tag{4.4}$$

$$\mathit{Full\ Precision} = \frac{\mathit{Full\ Match}}{\mathit{Full\ Match} + \mathit{Partial\ Match} + \mathit{Wrong\ Hit}} \tag{4.5}$$

$$\mathit{Partial\ Recall} = \frac{PartialMatch}{\mathit{Full\ Match} + \mathit{Partial\ Match} + \mathit{Complete\ Miss}} \tag{4.6}$$

$$\mathit{Full\ Recall} = \frac{FullMatch}{\mathit{Full\ Match} + \mathit{Partial\ Match} + \mathit{Complete\ Miss}} \tag{4.7}$$

Total Precision and Total Recall can be stated as *Partial Precision* + *Full Precision* and *Partial Recall* + *Full Recall*, respectively.

Tables 4.3 and 4.4 present these measures applied to our results. As expected, in an overall analysis, results reached with the partial matches overcome the results of the exact matches recognition analysis. We note that better results yield mostly by the Recall.

Table 4.3: Results concerning Perfect and also Partial Matches.

Elements Measures	**EQ pair**	**Entity**
Partial-Recall	0.05	0.08
Full-Recall	0.39	0.67
Partial-Precision	0.11	0.1
Full-Precision	0.75	0.84

Table 4.4: Total Results.

Elements Measures	**EQ pair**	**Entity**
Total Recall	0,45	0,76
Total Precision	0,87	0,94
Total F-measure	0,59	0,84

4.1.3 Discussion

The obtained findings indicate the consistence of the defined techniques. Nevertheless, our approach demands further refinements to identify more EQ elements. Among the already mentioned details required to improve the results, the key findings about the approach rely on the necessity of extending the EQ formalism to handle complex EQs with compound *Entities* and *Qualities*.

By the conducted evaluation, we observe an improvement of obtained results when comparing the analysis considering only the perfect matches. Nevertheless, we can refine our proposal in some directions. For example, to embed a method to perform a Entity Linking task to handle complex cases, *e.g.*, *S*[*first four dorsal spines prolonged, the second and third longest*]. This sentence requires identifying that the words *second* and *third* implicitly mention the *Entity E*[*dorsal spine*].

The obtained results are affected by the coverage of explored ontologies. The Recall related to *Entities* is high which indicates that the TAO ontology is relatively complete. Whereas the Recall of *Qualities* remains relatively low. It is close related to the low coverage of the PATO ontology. This is due to the fact that the universe of *Quality* terms is more vast than those available in PATO. Moreover, PATO is also a generic ontology, supporting a wide range of organisms, unlike the specialized anatomical ontologies. Our approach can serve as a tool to enrich the ontology, suggesting terms to be added.

A study comparing our results with related work is hampered by the unavailability of a Gold Standard Set used by them. However, it is possible to compare the proposals conceptually. Among the existing approaches, the most related is the *CharaParser* – part of the *Phenoscape* project – which has a good acceptance in the community. Our work presents, a more independence of human work in identifying the EQ elements, since *CharaParser* requires some steps of validation, by the user, over the information extracted, to feed the next steps.

4.2 Application Experiments

In this section, we present practical applications, which are possible due to the extraction of phenotypes. The objective is demonstrating the usefulness of explicitly recognizing EQs.

Section 4.2.1 describes possible relevant analyses on a knowledge network generated from *FishBase* data, exploring the recognized EQs. The knowledge network was created by correlating the detected EQs with other information elements available in *FishBase*. In particular, we correlated EQ pairs with data concerning taxonomic groups of fishes. Afterwards, we generated different information visualizations/perspectives to evaluate the obtained correlations. We selected specific cases to highlight the relevance of considering EQ statements.

Section 4.2.2 presents relevant features that can improve the system usage thanks to the recognized EQs.

4.2.1 Knowledge Network Analysis

From the analysis of the network graph generated in this evaluation, we highlight possible scientific studies to understand facts about living beings. The key subject investigated in this subsection concerns the ability of changing the focus of the analysis when the *Entities* that describe organisms are unified. It enables to focus in the descriptive characteristics of the organisms and analyse/compare them departing from such characteristics.

Phenotypes distinguishing taxa: Figure 4.2.a shows a fragment of the obtained knowledge network highlighting 3 classes of fishes and the EQ elements concerning the *tooth* structure. As can be seen, our approach enabled to unify the *Entities* and it is possible to verify that all 3 classes share the same EQ elements. However, if we drill down to the level of family, it is possible to verify which EQ elements distinguish the two families *Aulopiformes* and *Cetomimiformes* – the size of the tooth: the first one (Figure 4.2.b) is large and the second (Figure 4.2.c) is small.

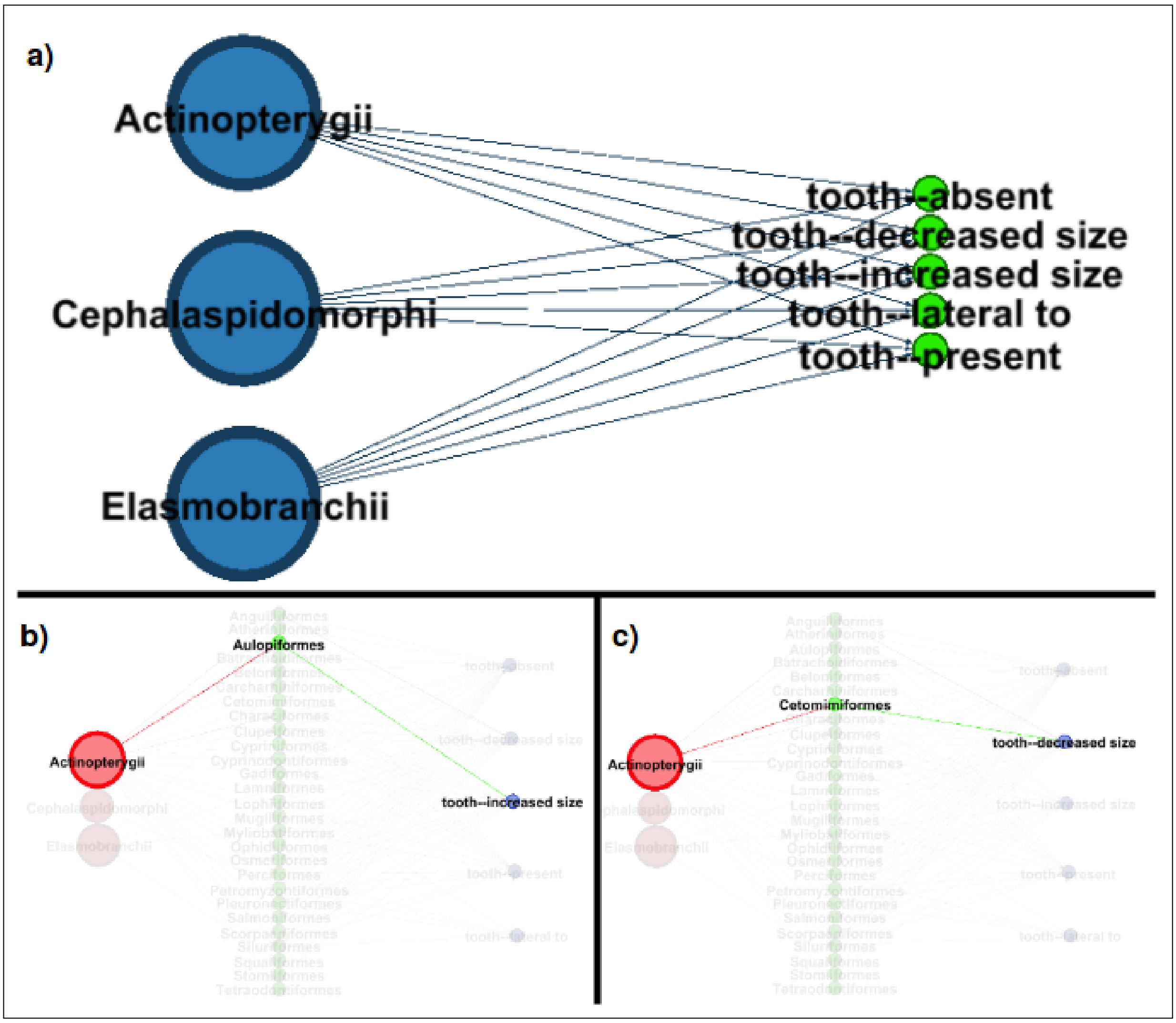


Figure 4.2: (a) Relation between a set of EQ elements and classes. (b) EQ element determiner of Aulopiformes Family. (c) EQ element determiner of Cetomimiformes family.

EQ sharing through taxa: We built a bipartite network consisting of two different types of nodes: species and EQ statements. In this network, each EQ element is linked to the species that has it.

Figure 4.3 shows a small portion of this network, in a synthetic view. Since several EQ elements are shared by a large number of species, the resulting bipartide network is too dense for direct visualization. While 29 species are on the left side, 6 EQ pair elements are on the right side. This network enables visualizing which EQ pairs are the most shared by the species.

In the visualization aspect, the size of the EQ nodes indicates the amount of linked species, *e.g.*, the $E[melanophore\ spot]Q[low\ brightness]$ is the biggest node, which means that it is an EQ pair present in many species.

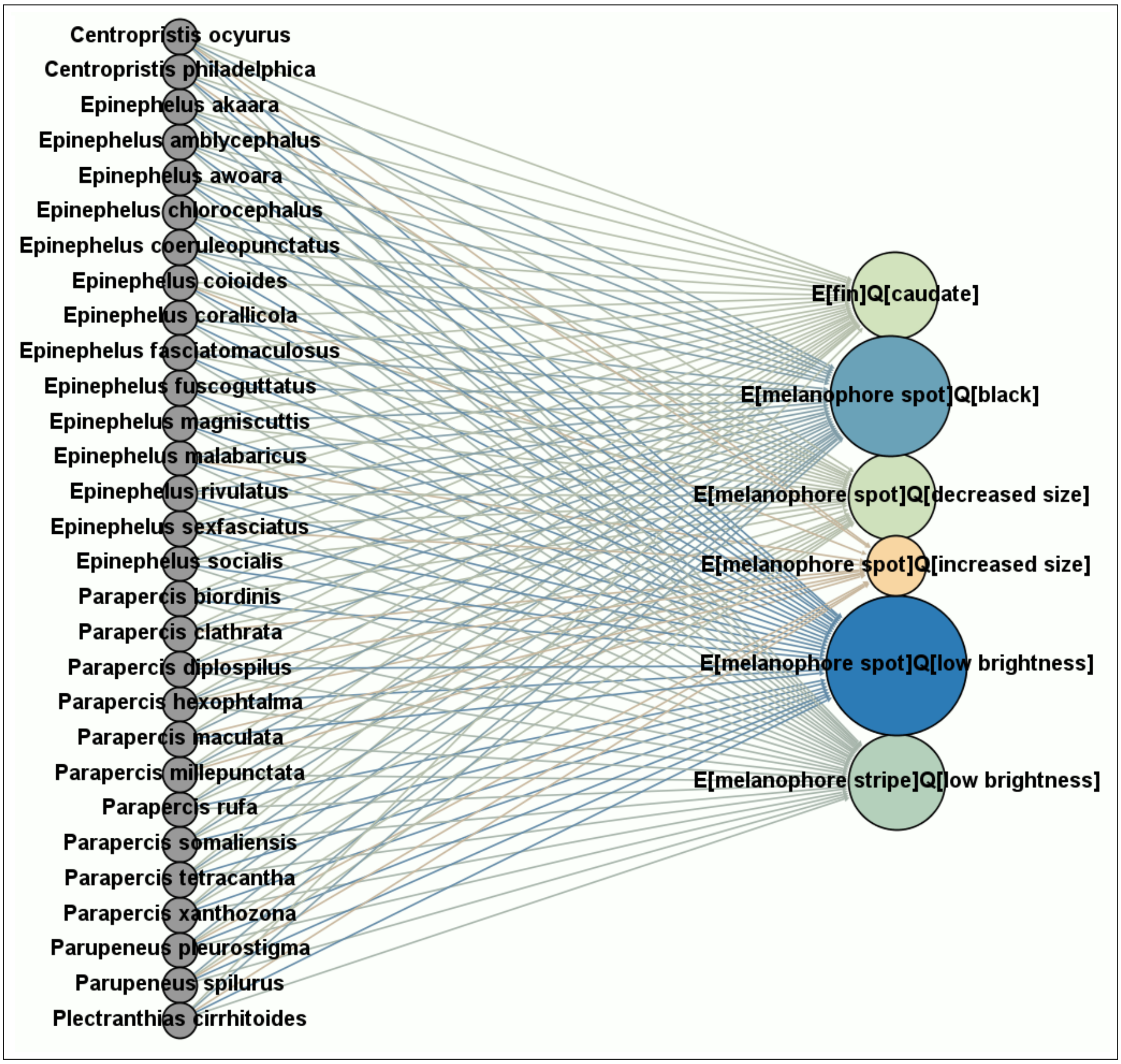


Figure 4.3: Bipartite network of Species and EQs elements, showing some of the most present EQ in the species.

Figure 4.4 shows a projection of the bipartite network. In this visualization, the nodes are EQs and they are connected if they are present together at least one species. The link width is proportional to the amount of shares. The size of the nodes indicates the prevalence of the EQ elements in species.

This visualization allows one to study which EQ elements frequently occur together. For example, the link width between the nodes $E[melanophore\ spot]Q[low\ brightness]$ and $E[melanophore\ spot]Q[decreased\ size]$ indicates that they are EQ elements present together in many species.

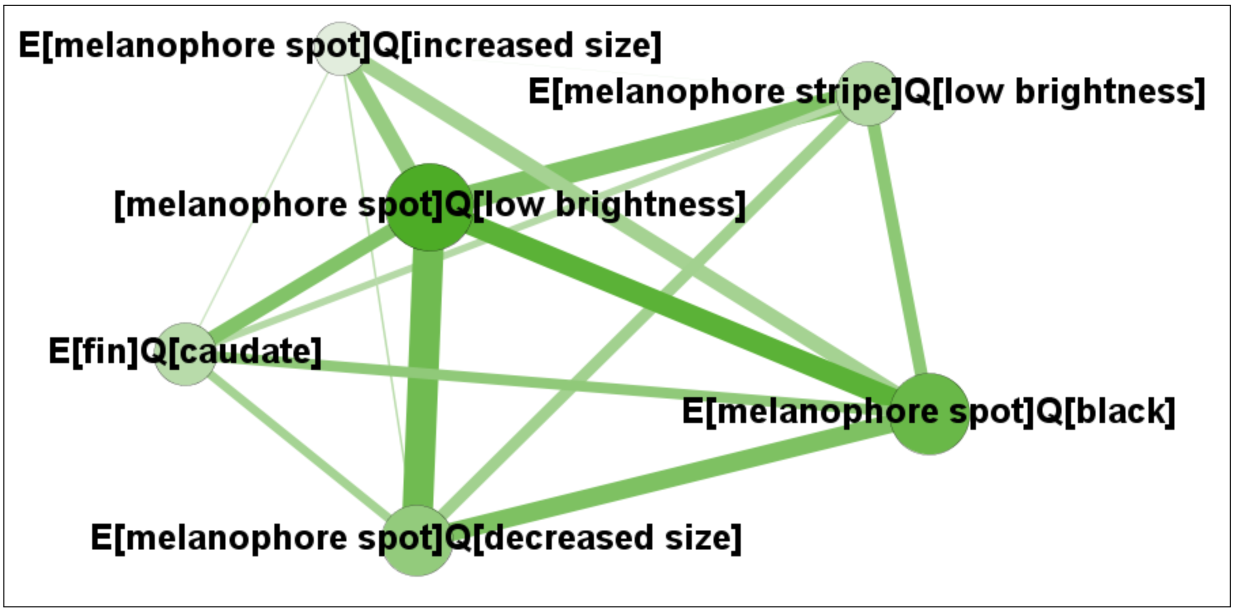

E[melanophore spot]Q[increased size]
E[melanophore stripe]Q[low brightness]
[melanophore spot]Q[low brightness]
E[fin]Q[caudate]
E[melanophore spot]Q[black]
E[melanophore spot]Q[decreased size]

Figure 4.4: Projection of the Bipartite network showing the most shared EQ elements by species.

4.2.2 System Usage Improvements

In addition to the the creation of a knowledge networks that benefits several types of biological analyses, the following applications are examples of how the general usage of a biological Information system, *e.g.*, *FishBase*, can be improved by the phenotype descriptions based on EQ elements. We further summarize them and Appendix A provides additional details.

Relation of taxa and IKs: This work allows unifying concepts before spread in many independent *Identification Keys* creating a network connecting *Entities*, *Qualities*, *Identification Keys* and taxa. In the network, it is possible perform relevant analyses including: (i) what keys share characteristics of a given taxon; (ii) what keys have complementary information about a taxon.

Knowledge to start the identification process: In *FishBase*, the *Identification Keys* are classified based on the taxon that they identify. Therefore, the user must have some previous knowledge to start the identification process, *e.g.*, the user must know that the observed fish belongs to the family of Teleostean to pick an IK that identify this group of fishes.

Using the new structure achieved with this proposal, it is possible to start the identification process without picking an *Identification Key*, *i.e.*, we can start the process from the observable characteristics of the specimen.

Searching through incremental filtering: In the current hierarchical structure of *Identification Keys* from *FishBase* it is not possible to conclude the process if the user does not know about a characteristic stated in an intermediate Key Question (a Key

Question that is not a leaf on the tree representing the *Identification Key*).

With the generated graph, it is possible to search for specific taxa by applying an incremental filtering process. The user can thus perform the process of identification based on the characteristics that (s)he knows, instead of following a flow created by another people.

4.2.3 Discussion

The results obtained by the analyses conducted over the data resulting of our approach show the potential of this work. Information Systems about living beings, like *FishBase*, can be empowered handling phenotypes in EQ format, generating a vast amount of system usage possibilities. Moreover, the knowledge networks provide useful information to several biology studies about species evolution.

The analyses performed over the network are examples of possible studies. However, due to the relatively low recall of our extraction approach, they cannot be taken as facts. Future work aims to improve the results of the extraction method, which can allow to perform refined analyses over the network with a high rate of confidence.

4.3 Summary

This chapter presented an empirical validation of our proposal by conducting quantitative and qualitative evaluations. The numeric evaluation allowed a initial assessment to understand the rate and coverage of the EQ recognition from the IKs of *FishBase*. To examine the correctness of the results, a gold standard was constructed and both exact and partial matches recognition analysis were performed. To comprehend the general benefits of the proposed recognition method for phenotype integration, we departed from our the knowledge network and conducted a qualitative analysis correlating KQs and species from *FishBase*.

The validation showed an overall efficiency of the proposal in the different numeric evaluations. Results indicated a good accuracy for the technique. This chapter also demonstrated that our approach remains applicable and useful to network-driven analysis. Results revealed several advantages of the method to improve reuse, identification facilities and refined data connections in graph-based analyses. Furthermore, we discussed the limitations of the investigation and highlighted potential improvements. The next chapter closes this thesis with the major conclusions and description of future work.

Chapter 5

Conclusion and Future Work

Phenotype descriptions play a key role in biological knowledge bases, but most of the descriptions remain in a free-textual format, which affects machine interpretation and their applicability in network-driven analyses.

This thesis proposed an original approach to recognize *Entities* and *Qualities* connecting them to concepts in ontologies to make their representation semantically interpretable by machines. Our key point, not addressed by related work found in literature, consists in exploring clues of non-textual information: from the writing characteristics of phenotype descriptions to their organizational structure.

The experimental evaluations revealed encouraging results regarding the assessment against a gold standard set. The experiments point out the contributions of each step to improve the results of the recognition process.

The experiments using the EQ elements, extracted from free-text sentences applying our proposal, showed the advantages of bringing these descriptions to a common and formal language. It enables machines better consuming and interpreting the available descriptions.

Phenotype descriptions in EQ format are more suitable to be reused by different systems and researchers. The demonstrated applications are relevant examples of how the extracted data can be used in scientific research. Future work involves to validate the technique with biology researchers. Other kinds of studies could be performed, such as analyses involving graph theory, complex networks, link prediction, and so on. Even though the proposed method has been developed and experimented inside the *FishBase* context, it was designed to be generalized to a wider spectrum of biological information systems.

In spite of the relevance of the achieved contributions in this work, the proposed approach still requires further evaluation in terms of comparison with other investigations present in literature. Future work involves conducting such additional evaluations to measure and compare the efficiency with respect to other approaches.

Although precision and recall metrics achieved lower indexes than in approaches found in literature, our usage of non-content information shows clear improvements in information extraction tasks, as an alternative to the strong need of training sets containing a previously annotated corpus.

Future work aims at addressing issues concerning the limitations of the recognition of

EQ elements. In the Match Expansion Algorithm used by both Step 1 and Step 2, we plan to create alternative flows to handle special relations beyond *number* and *conj*. To create such alternative flows it is necessary a thoroughly study about such special cases and how they reflect the writing characteristics of phenotype descriptions.

Improvements on Step 1 demand to consider complex EQs, *i.e.*, *Qualities* comparing an *Entity* to another *Entities*, *e.g.*, $E[Mouth]Q[extending\ to\ level\ of]E[Eye]$ in the sentence $S[Mouth\ extending\ to\ eye\ level]$. To handle such kind of phenotype descriptions it is necessary to extend the EQ formalism to consider these cases, since their configuration is not strictly a pair of EQ elements.

Improvements in Step 2 involve to expand the comparison to be performed between the subtrees representing the *Quality* parts, instead of only the edges connecting the subtrees. At this stage, Step 2 is limited to recognize *Qualities* linked to *Entities* already recognized in Step 1. Thus, we plan to refine this step to recognize complete new EQ pairs, *i.e.*, *Entities* and *Qualities* not recognized previously in Step 1.

In our method, the algorithms detected some candidate *Entities* and *Qualities* that do not belong to the employed ontologies, so they are not confirmed as new vertices representing *Entities* and *Qualities.* Such cases negatively affected the obtained results concerning the Recall metric. Therefore, the our findings are limited by the coverage of the ontologies. To address this limitation, an extension of our investigation is in the area of Ontology Engineering, supporting the coverage improvement of these ontologies, providing suggested concepts to enrich them.

The research developed in this thesis resulted in three scientific papers: the first, entitled “Semantic Interpretation of Biological Identification Keys” [37], was presented at the XXX Simpósio Brasileiro de Banco de Dados (SBBD - October 2015, Petrópolis-RJ). The second, entitled “Knowledge Network Generation from Phenotypic Descriptions” was accepted in the IEEE 12th International Conference on e-Science, to be presented at Baltimore, Maryland (EUA) in October 2016. The third, entitled “Progressive Data Integration and Semantic Enrichment Based on LinkedScales and Trails” has been submitted and it is waiting for the notification of acceptance to 9th International SWAT4LS Conference - Semantc Web appplications and tools for life sciences.

Bibliography

[1] Noha Alnazzawi, Paul Thompson, Riza Batista-Navarro, and Sophia Ananiadou. Using text mining techniques to extract phenotypic information from the phenochf corpus. *BMC Medical Informatics and Decision Making*, 15(Suppl 2):S3, 2015.

[2] Sophia Ananiadou, Douglas B Kell, and Jun-ichi Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579, 2006.

[3] Ion Androutsopoulos and Prodromos Malakasiotis. A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research*, 38:135–187, 2010.

[4] Alan R Aronson and François-Michel Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[5] Samet AtdaÇ§ and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. *2013 2nd International Conference on Systems and Computer Science, ICSCS 2013*, pages 228–233, 2013.

[6] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[7] Patrícia Cavoto, Victor Cardoso, Regine Vignes Lebbe, and André Santanchè. Fishgraph: A network-driven data analysis. In *e-Science (e-Science), 2015 IEEE 11th International Conference on*, pages 177–186. IEEE, 2015.

[8] B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, January 1999.

[9] Wendy W Chapman and K Bretonnel Cohen. Current issues in biomedical text mining and natural language processing. *Journal of Biomedical Informatics*, 42(5):757–759, 2009.

[10] Massimiliano Ciaramita, Aldo Gangemi, Esther Ratsch, Jasmin Saric, and Isabel Rojas. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 659–664, 2005.

[11] Andrew B Clegg and Adrian J Shepherd. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24, 2007.

[12] Hong Cui. Charaparser for fine-grained semantic annotation of organism morphological descriptions. *Journal of the American Society for Information Science and Technology*, 63(4):738–754, 2012.

[13] Wasila Dahdul, T Alexander Dececchi, Nizar Ibrahim, Hilmar Lapp, and Paula Mabee. Moving the mountain: analysis of the effort required to transform comparative anatomy into computable anatomy. *Journal of Biological Databases and Curation*, 2015, 2015.

[14] Wasila M Dahdul, John G Lundberg, Peter E Midford, James P Balhoff, Hilmar Lapp, Todd J Vision, Melissa A Haendel, Monte Westerfield, and Paula M Mabee. The teleost anatomy ontology: anatomical representation for the genomics age. *Systematic Biology*, 59(4):369–383, 2010.

[15] Mike J Dallwitz, TA Paine, and EJ Zurcher. Principles of interactive keys. Web-based document http://biodiversity.uno.edu/delta, 2000. Accessed: 2015-09-13.

[16] Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.

[17] Elizabeth J Farnsworth, Miyoko Chu, W John Kress, Amanda K Neill, Jason H Best, John Pickering, Robert D Stevenson, Gregory W Courtney, John K VanDyk, and Aaron M Ellison. Next-generation field guides. *BioScience*, 63(11):891–899, 2013.

[18] Miriam Fernández, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. Semantically enhanced information retrieval: an ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434–452, 2011.

[19] Rainer Froese and Daniel Pauly. *FishBase 2000: concepts, design and data sources.* WorldFish, 2000.

[20] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex–relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[21] Anais Grand, Regine Vignes Lebbe, and Andre Santanche. From phenotypes to trees of life: A metamodel-driven approach for the integration of taxonomy models. In *IEEE 10th International Conference on e-Science*, volume 1, pages 65–72, 2014.

[22] David Grosser, Noël Conruyt, and Henri Ralambondrainy. Identification with iterative nearest neighbors using domain knowledge. *Proceedings of the International Congress of Tools for Identifying Biodiversity: Progress and Problems*, 2010.

[23] Baohua Gu. Recognizing nested named entities in genia corpus. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 112–113. Association for Computational Linguistics, 2006.

[24] Lynette Hirschman, Alexander A Morgan, and Alexander S Yeh. Rutabaga by any other name: extracting biological names. *Journal of Biomedical Informatics*, 35(4):247–259, 2002.

[25] Chung-Chi Huang and Zhiyong Lu. Community challenges in biomedical text mining over 10 years: success, failure and the future. *Briefings in bioinformatics*, 17(1):132–144, 2016.

[26] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[27] Lucid. Lucid phoenix, 2016.

[28] Y Lussier and C Friedman. Biomedlee: a natural-language processor for extracting and representing phenotypes, underlying molecular mechanisms and their relationships. *International Conference on Intelligent Systems for Molecular Biology*, 2007.

[29] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[30] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489, 2013.

[31] S Martellos and PL Nimis. Keytonature: teaching and learning biodiversity. dryades, the italian experience. In *Proceedings of the IASK International Conference Teaching and Learning*, pages 863–868, 2008.

[32] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[33] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[34] Meenakshi Narayanaswamy, KE Ravikumar, K Vijay-Shanker, and K Vij Ay-shanker. A biological named entity recognizer. In *Pac Symp Biocomput*, page 427, 2003.

[35] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the second International Conference on Human Language Technology Research*, pages 82–86. Morgan Kaufmann Publishers Inc., 2002.

[36] Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

[37] Fagner Leal Pantoja, Júlio César dos Reis, and André Santanchè. Semantic interpretation of biological identification keys. In *XXX Simpósio Brasileiro de Banco de Dados - Short Papers, Petrópolis, Rio de Janeiro, Brasil, October 13-16, 2015.*, pages 99–104, 2015.

[38] Sampo Pyysalo and Sophia Ananiadou. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875, 2014.

[39] Cartic Ramakrishnan, Pablo N Mendes, Shaojun Wang, and Amit P Sheth. Unsupervised discovery of compound entities for relationship extraction. In *Knowledge Engineering: Practice and Patterns*, pages 146–155. Springer, 2008.

[40] Matthew S Simpson and Dina Demner-Fushman. Biomedical text mining: A survey of recent progress. In *Mining Text Data*, pages 465–517. Springer, 2012.

[41] Cynthia L Smith and Janan T Eppig. The mammalian phenotype ontology: enabling robust annotation and comparative analysis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(3):390–399, 2009.

[42] Min Song, Hwanjo Yu, and Wook-Shin Han. Developing a hybrid dictionary-based bio-entity recognition technique. *BMC Medical Informatics and Decision Making*, 15(Suppl 1):S9, 2015.

[43] Anne E Thessen, Hong Cui, and Dmitry Mozzherin. Applications of natural language processing in biodiversity science. *Advances in bioinformatics*, 2012, jan 2012.

[44] Visotheary Ung, Guillaume Dubus, René Zaragüeta-Bagils, and Régine Vignes-Lebbe. Xper2: introducing e-taxonomy. *Bioinformatics*, 26(5):703–704, 2010.

[45] Tak-Lam Wong, Wai Lam, and Tik-Shun Wong. An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *Proceedings of the 31st annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42. ACM, 2008.

[46] Pierre Zweigenbaum and Dina Demner-Fushman. Advanced literature-mining tools. In *Bioinformatics*, pages 347–380. Springer, 2009.

Appendix A

Application Experiments

This appendix details the application experiments showed in Section 4.2. We have conducted these experiments together with Msc. Patrícia Cavoto.

A.1 Improving the System Usage

Figure A.1 shows a graph model derived from *FishBase* (*FishGraph*), created by a previous work of Cavoto *et al.* [7]. It highlights the node types (*class, order, family, species, genus, country, key, and ecosystem*) and relationships among them. We have added new nodes to FishGraph – *keyQuestion*, *EQ*, *Entity*, and *Quality* – and linked them to the existing ones.

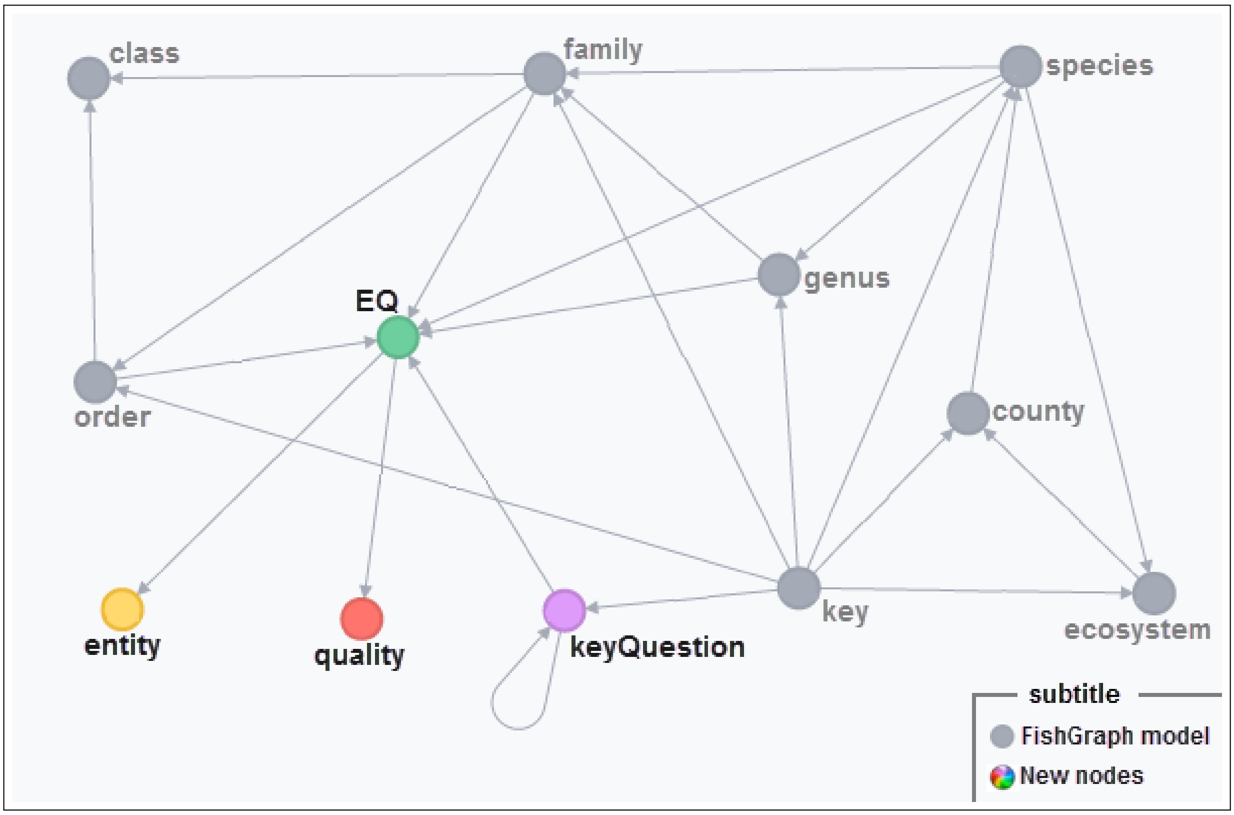


Figure A.1: New nodes added in the *FishGraph* database.

This updated *FishGraph* version models our knowledge network. It allows scientists to perform analyses making use of the new extracted information. We further present examples of possible applications to improve the system usage by the user and analyses to understand facts about living beings.

A.1.0.1 No need of previous knowledge

Currently, each *IK* is represented in *FishBase* as an independent tree, which hampers their usage, as one needs to know beforehand the main taxon (the root of the IK tree) to start the identification process.

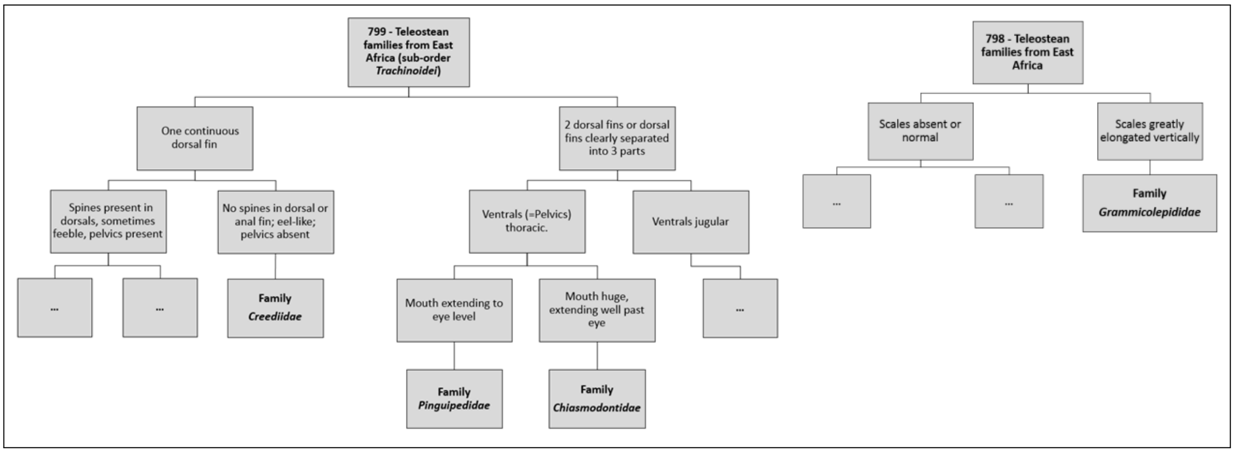

799 - Teleostean families from East Africa (sub-order *Trachinoidei*)
One continuous dorsal fin
2 dorsal fins or dorsal fins clearly separated into 3 parts
Spines present in dorsals, sometimes feeble, pelvics present
No spines in dorsal or anal fin; eel-like; pelvics absent
Ventrals (=Pelvics) thoracic.
Ventrals jugular
...
...
Family *Creediidae*
Mouth extending to eye level
Mouth huge, extending well past eye
...
Family *Pinguipedidae*
Family *Chiasmodontidae*
798 - Teleostean families from East Africa
Scales absent or normal
Scales greatly elongated vertically
...
...
Family *Grammicolepididae*

Figure A.2: Part of *Identification Keys*: 799 of Teleostean families from East Africa (sub-order *Trachinoidei*) and 798 of the Teleostean families from East Africa.

As an example, consider the IK "799 – Teleostean families from East Africa (sub-order Trachinoidei)" (*cf.* Figure A.2). The identification process using this key requires the following knowledge: the specimen belongs to the teleostean family, sub-order Trachinoidei and it is found in East Africa. The identification process is hampered if the user knows only part of the root – suppose family and geographic location – since FishBase has another 6 IKs of the teleostean family distinguished mainly by the sub-order, *e.g.*, IK 798 also in Figure A.2. Even with all the required knowledge, it is necessary to follow the proposed path in the IK tree. All these particularities make the identification process only possible to specialists.

The new structure allows starting the identification process from any known characteristic. For instance, we can start the identification process using a known characteristic like *dorsal fin soft*, independently of any IK or other characteristic.

A.1.0.2 Searching through incremental filtering

The generated knowledge graph allows searching for specific taxa by applying an incremental filtering process.

Figure A.3 shows an example of this incremental filter using *Entities* and *Qualities*, which leads to a family with three specific characteristics. Figure A.3.a shows the initial filtered graph with 27 families of species that have the $E[dorsal\ fin]Q[soft]$ (*Entities* and

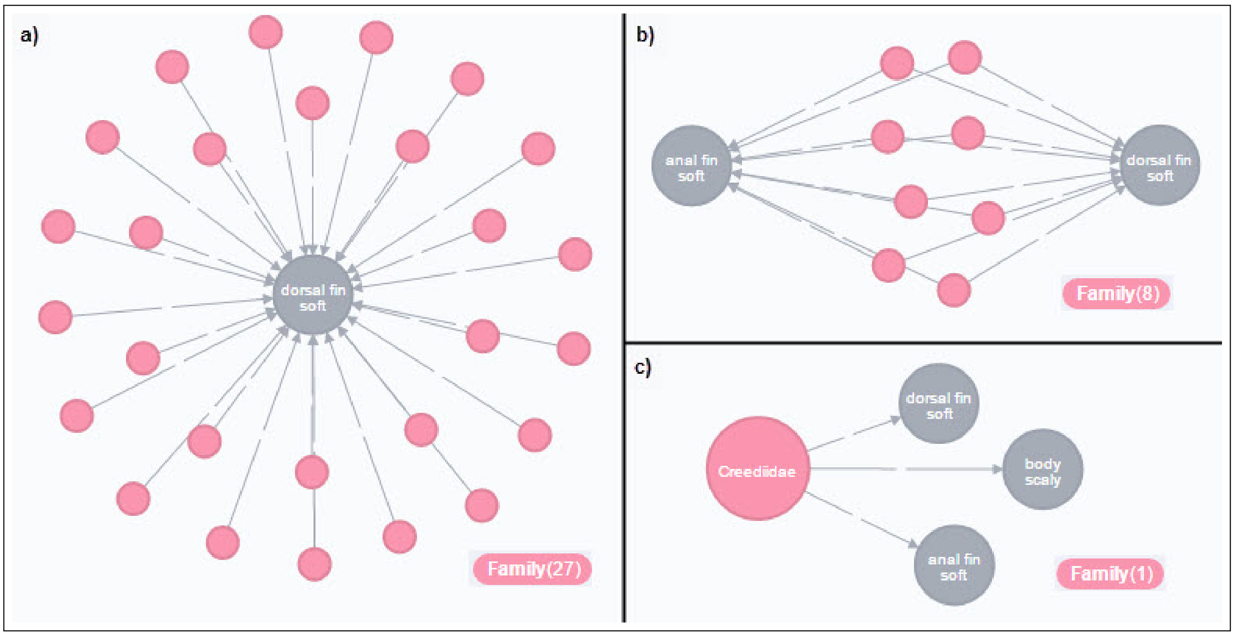


Figure A.3: Filtering families of species by EQ: a) dorsal fin soft; b) dorsal fin soft and anal fin soft; and c) dorsal fin soft, anal fin soft, and body scale.

Qualities are collapsed in a single node, in order to simplify the view). Adding a second filter of the $E[anal\ fin]Q[soft]$ (Figure A.3.b) means to select those species with edges to both EQs. The number of families with both characteristics decreases to 8. A third filter of the $E[body\ scale]$, results in only 1 family that has the 3 characteristics: *Creediidae* (Figure A.3.c).

A.1.0.3 Relation of taxa and IKs

One taxon is referred in many IKs in *FishBase* but, since they are independent, each IK has its own set of characteristics. When we analyse IKs referring to the same taxon, there are two possible cases: (i) keys share partially or totally the characteristics of a given taxon; (ii) keys that have complementary information about the taxon.

Our unified graph structure links distinct characteristics of the same taxonomic group, coming from many independent IKs, enriching and facilitating the identification process. Returning to the previous experiment, the $E[body\ scale]$ is a characteristic that belongs to IK 324 but it does not belong to IK 799. Since they refer to the same taxonomic group, it is possible to combine them to achieve a more complete description of the taxa.