ONTOLOGY BASED INFORMATION EXTRACTION ON FREE TEXT RADIOLOGICAL REPORTS USING NATURAL LANGUAGE PROCESSING APPROACH

A THESIS SUBMITTED TO
THE INSTITUTE OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERGIN SOYSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE INSTITUTE OF INFORMATICS

SEPTEMBER, 2010

Approval of the Institute of Informatics

_____

Prof.Dr.Nazife Baykal

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

_____

Prof.Dr.Nazife Baykal

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

_____	_____
Assoc.Prof.Dr.İlyas Çiçekli	Prof.Dr.Nazife Baykal
Co-Supervisor	Supervisor

Examining Committee Members

Prof.Dr.Osman Saka	(Akdeniz U)	_____
Prof.Dr.Nazife Baykal	(METU, II)	_____
Dr.Ali Arifoğlu	(METU, II)	_____
Assoc.Prof.Dr.Erkan Mumcuoğlu	(METU, II)	_____
Prof.Dr.Mustafa Özmen	(Hacettepe U)	_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name : **Ergin Soysal**

Signature : _____

ABSTRACT

ONTOLOGY BASED INFORMATION EXTRACTION ON FREE TEXT RADIOLOGICAL REPORTS USING NATURAL LANGUAGE PROCESSING APPROACH

Soysal, Ergin

MD, PhD

Supervisor: Prof. Dr. Nazife Baykal

Co-Supervisor: Assoc. Prof. Dr. Ilyas Cicekli

September 2010, 110 Pages

This thesis describes an information extraction system that is designed to process free text Turkish radiology reports in order to extract and convert the available information into a structured information model. The system uses natural language processing techniques together with domain ontology in order to transform the verbal descriptions into a target information model, so that they can be used for computational purposes. The developed domain ontology is effectively used in entity recognition and relation extraction phases of the

information extraction task. The ontology provides the flexibility in the design of extraction rules, and the structure of the ontology also determines the information model that describes the structure of the extracted semantic information. In addition, some of the missing terms in the sentences are identified with the help of the ontology. One of the main contributions of this thesis is the usage of ontology in information extraction that increases the expressive power of extraction rules and helps to determine missing items in the sentences. The system is the first information extraction system for Turkish texts. Since Turkish is a morphologically rich language, the system uses a morphological analyzer and the extraction rules are also based on the morphological features. TRIES achieved 93% recall and 98% precision results in the performance evaluations.

Keywords: Radiological Reports, Information Extraction, Ontology, Natural Language Processing, Turkish

ÖZ

SERBEST METIN RADYOLOJİ RAPORLARINDAN, DOĞAL DİL İŞLEME YAKLAŞIMLARI KULLANARAK ONTOLOJİ TEMELLİ ENFORMASYON ÇIKARIMI

Soysal, Ergin

MD, PhD

Danışman: Prof. Dr. Nazife Baykal

Y. Danışman: Doç. Dr. Ilyas Cicekli

Eylül 2010, 110 Sayfa

Bu tez, serbest metin Türkçe radyoloji raporlarını işleyerek, var olan bilgiyi çıkartıp, yapılandırılmış bilgi modeline dönüştüren bir bilgi çıkarım sistemini tanımlar. Sistem, doğal dil işleme tekniklerini, bir alan ontolojisi ile birlikte kullanarak, sözel olarak yapılmış tanımlamaları hedef bilgi modeline çevirir ve böylece bilgi, bilgisayar tarafından işlenebilir hale getirilmiş olur. Geliştirilen alan ontolojisi, bilgi çıkarımı sırasında, varlık tanımlama ve ilişki çıkarılması aşamalarında etkili olarak kullanılmıştır. Ontoloji, çıkarım kurallarının tasarımında esneklik sağlar. Ontolojinin yapısı, çıkarılan semantik bilginin yapısın da tanımlayan bilgi modelini belirler. Ek olarak, ontoloji cümlelerde yer alması gereken, ancak, bilindiği var sayıldığı için

ifade edilmeyen varlıkların saptanmasını sağlar. Bu tezin temel katkılarından biri, ontolojinin bilgi çıkarım kuralları içerisinde kullanılarak ifade gücünü artırması ve cümlelerdeki kayıp varlıkların saptanmasını sağlamasıdır. Sistem Türkçe metinler için geliştirilen ilk bilgi çıkarım sistemidir. Türkçe, morfolojik olarak zengin bir dil olduğu için, sistem bir morfolojik analizör kullanır ve çıkarım kuralları da bu morfolojik özelliklerden faydalanır. Sistem, performans değerlendirmesinde, %93 geri çağırma ve %98 duyarlık değerlerine ulaşmıştır.

Anahtar kelimeler: Radyoloji Raporları, Enformasyon Çıkarımı, Ontoloji, Doğal Dil İşleme, Türkçe

ACKNOWLEDGMENTS

I would like to thank to my supervisors Assoc. Prof. Ilyas Cicekli and Prof. Dr. Nazife Baykal for their support throughout the conduction of this thesis, and to Prof. Dr. Osman Saka for his contribution in evaluation of the system user interface and support throughout the program.

I also would like to thank to Prof. Dr. Serdar Akyar, Prof. Dr. Mustafa Özmen and Prof. Dr. Utku Şenol for their support, assistance and their allowance to access to de-identified radiology reports during our research.

TABLE OF CONTENTS

LIST OF FIGURES

Number *Page*

LIST OF TABLES

Number *Page*

PREFACE

Medical narratives still constitutes the majority of clinical information and records collected by health information systems. This is an unchanged fact, even if the electronic records are getting widely used since 1980s. This arises from difficulties of data acquisition because of the diversity of information structures in healthcare domains.

Objective of this thesis is to generate a system that will transform free text radiological reports in Turkish into computationally reusable structured information. This transformation requires overcoming of several challenges. Initially, Turkish as an agglutinative language requires an extensive morphological analysis to access to proper semantics. On the other hand proper semantics also requires domain knowledge to establish proper relations among the concepts embedded into the natural language. Finally, system requires a meaningful target information model, which will allow to maximum re-use of the information output without loss of semantics.

This thesis covers the aspects of this information extraction system that successfully achieves this goal. After introductory and background chapters (chapters 1 and 2), third chapter overviews the components of Turkish Radiological Information Extraction System (TRIES) and interactions among them. Chapter 3, 4 and 5 explain the Turkish morphological analysis, information modeling and information extraction tasks for TRIES. In chapter 7, an application named TRIES user interface (TRIES UI) is introduced, which to allow clinicians to query TRIES extracted data repository. In chapter 8, evaluation results are given for both TRIES and TRIES UI.

ABBREVIATIONS

PL : Plural noun form

ACC : Accusative noun form

DAT : Dative (to …) noun form

LOC : Locative (at, on or in …) noun form

ABL: Ablative (from …) noun form

GEN : Genitive (of …) noun form

INS : instrumental (with …)

NEG: Negated form

COP: Copula

PERS1SG: First person singular.

PERS2SG: Second person singular.

PERS3SG: Third person singular.

PERS1PL: First person plural.

PERS2PL: Second person plural.

PERS3PL: Third person plural.

POSS1SG: First person singular possessive.

POSS2SG: Second person singular possessive.

POSS3SG: Third person singular possessive.

POSS1PL: First person plural possessive.

POSS2PL: Second person plural possessive.

POSS3PL: Third person plural possessive.

PAST: Past tense

PRESCONT: Present Continuous Tense

FUTURE: Future tense

INVL: Affix adding the meaning of "involved in" to the root word

SNOMED: Systematized Nomenclature of Medicine

ICD-9: International Classification of Diseases, 9th Revision

ICD-10: International Classification of Diseases, 10th Revision

UMLS: Unified Medical Language System

Chapter 1

INTRODUCTION

Health information systems and electronic health records are expected to lower costs and improve health care quality through improved access to information [1]. Free unstructured text is still the most common information source in medical records. Many medical disciplines such as radiology, pathology, and nuclear medicine almost completely rely on unstructured free text as the route of dissemination for information. This format is widely used for both storage and exchange of information about an individual patient, and the file of an individual patient usually contains several different free text reports such as clinical notes, patient history, or discharge summaries. Information covered in these reports is a valuable data resource for management, research, or educational purposes. Medical applications such as clinical decision support systems require utilizing this information. Nevertheless, this form of information is not as useful as structured and coded data for neither decision making nor knowledge discovery

Since the free text information constitutes the majority of the clinical data produced by healthcare professionals, natural language processing becomes even more critical in medical domain. Based on the processing requirements for these data to use for clinical research or decision support, there are several early attempts to process medical narratives for research starting from 1970s [2, 3]. Initial works conducted on the medical language processing (e.g. Hirschman or Sager) was focusing on retrieval of facts from medical records.

As more and more text becomes available electronically, there is a growing need for systems that extract information automatically from narrative data. Manual extraction of this information is quite costly and time consuming process. As the text source grows, machine

evaluation becomes mandatory to be able to use this huge amount of text. Information extraction (IE) and natural language processing (NLP) techniques are required to extract the useful information from these free texts.

Information extraction which is a subdiscipline of NLP focuses on the identification of the specific facts and relations within unstructured texts, the extraction of the relevant values, and their transformation into standardized codes and/or structured information. An information extraction task takes two inputs, namely a free text document that is the source of information and predefined templates, and fills these templates with suitable information extracted from the given document. The filled templates are the structured representation of the information available in the given document.

Radiology as a medical science generates many narrative documents very rich in information content, so called *radiology reports*. Radiology reports are generated as the results of different types of radiodiagnostic examinations such as computerized tomography (CT), ultrasonography (US), magnetic resonance imaging (MRI) or plain x-ray films. These examinations intent to collect information from patients about specific conditions, those may be useful during the diagnostic processes of patients. Depending on the clinical requirements, one or more part of the body of the patient is examined by one or more of the radiological techniques mentioned above, and reports that are documenting the findings are created in the form of free texts. Most of the general hospitals already store these reports in electronic media. These reports contain very important clinical information that would be useful for improvement of quality of care for the individual patient or expand the research capabilities as aggregated data. As narratives, although the required information to answer many medical questions is stored electronically, we cannot answer precisely many questions like "What is the rate of non pathological renal cysts in patients without renal complaints?", "What are the average sizes of left and right kidneys in our population?", and "How is renal parenchymal echogenic structure changing over the time, before a renal cancer is diagnosed?" since the required information is not available computationally. Transforming this narrative into a

structured format would offer us ability to use this information for many advanced purposes both in clinical practice and research.

Although there are some attempts to process radiological reports for different purposes, there is not any solution covering Turkish. As an agglutinative language, it is relatively more difficult to handle Turkish language with usual solutions. On the other hand, a complete transformation of a report into a structured form without loss or change of semantics of information is another challenge for such a quest.

1.1 Aims of the Thesis

In this thesis, we present a prototype IE system for Turkish radiology reports. The system addresses following research topics:

- How to transfer the domain knowledge implicitly used by radiologist into information extraction process?
    - Is it possible to represent the domain knowledge with ontology? How?
- How to integrate/utilize morphological analysis into different steps of information extraction process, which is a required process for agglutinative languages like Turkish?
- What should be the target information model, which will cover the complete report without loss or change in semantics of stored information?

As a result, our IE system is designed to convert a complete radiology report into a target relational information model. The prototype presented here tested against reports obtained from abdominal ultrasonography examinations. The Turkish radiological information extraction system (TRIES) uses rules as grammatical knowledge and ontology as both domain knowledge for named entity recognition and semantic analysis. The usage of effective hand-

coded rules is still one of the best approaches in order to get a medical information extraction system with high precision and recall values.

1.2 Contributions

One of the main contributions of this thesis is the usage of ontology in information extraction that increases the expressive power of extraction rules and helps to determine missing items in the sentences. The usage of the domain ontology provides flexibility in the design of rule templates in information extraction systems. The domain ontology can determine the information model that describes the structure of the extracted semantic information in information extraction systems.

Our system is the first information extraction system for Turkish texts. Since Turkish is a morphologically rich language, we use a morphological analyzer and our extraction rules are also based on the morphological features. The morphological processing is important in information extraction systems in agglutinative languages such as Turkish. The morphological analysis increases the flexibility of entity recognition and relation extraction in those kinds of information extraction systems.

1.3 Organization

The rest of the thesis is organized as follows. Chapter 2 overviews the background and the related work in medical information extraction systems and ontology-based information extraction systems. Chapter 3 covers a general overview of our work Turkish Radiological Information Extraction System (TRIES). In Chapter 4, we present the details of morphologic analyzer with morphological problems and solutions specific to Turkish language. Chapter 5 explains both information models represented in the reports, and model extracted by the system as the final output to be consumed for further use. Rule based extraction process of TRIES is discussed in Chapter 6. In Chapter 7, the user interface is described, which was built

to serve to let end user to query and utilize TRIES extracted data. The performance results of our information extraction system and evaluation results for TRIES user interface are given in Chapter 8. We give the concluding remarks in Chapter 9.

Chapter 2

BACKGROUND AND RELATED WORK

In the absence of structured and standardized methods of storage and sharing of information, the free text is the most important tool that is being used to maintain information. On the other hand, difficulties in aggregation and re-use of this information become an important challenge. Information extraction (IE) as a field, aims to transform unstructured natural language into a structured form, which can be processed computationally. It is a gripping field since mid-80s, which was promoted by US government since late 1980s.

Information extraction is an emerging subdiscipline of natural language processing (NLP), and frequently borrows methods from NLP. Since the main objective of IE is to achieve extraction of embedded *entities* and *relations among them* from the written language, handling of the rules affecting the *semantics* become an important factor in the success of IE. In order to overcome many problems arising from the language itself such as ambiguity, linguistic analysis at varying levels is required. NLP especially becomes mandatory in agglutinative languages like Turkish since affixes heavily contributes to *the meaning*.

2.1 Natural Language Processing

Language is an organized set of signals that provide communications between human beings. It's not only a way of communication, but, it is also used over historical periods for the preservation of information of any kind. Regarding its information content, demands on analysis, understanding or generation of spoken or written language by computers yielded the field of natural language processing.

Natural language processing offers different tools to process language depending on the requirements. These applications show a great variation starting from just identification or synthesis of sounds (i.e. speech analysis and generation) to understanding and representing the meaning in a different manner for a given text (i.e. natural language understanding).

Free text processing applications require different set of tools. These analysis tools can roughly be divided into three categories based on the target processing level on the language. Some applications require processing of components of each individual word. This is achieved by morphological analysis tools. Syntax analyzers work on sequences of words on the sentence level. They try to take the advantage of rules in the sentence formation and syntactic categories (count nouns, verbs with tenses, etc.) [4]. On the semantic level, semantic analyses try to determine the meaning of a given sentence.

Different applications of text processing such as information retrieval, information extraction or text mining often rely on one or more of these three categories at varying degrees.

2.1.1 Morphological Analysis

Morphology is the study of *morphemes*. Morphemes are the smallest meaningful units of grammar, forming words by joining together. E.g. in the word "dogs", there are two morphemes: the morpheme "dog" which stands for the name of an animal, and the morpheme "s" which mark the word as plural.

A morpheme can occur stand alone fashion. E.g. the word "dog" in above example. These are called *free morphemes* or *root words*. Some morphemes can occur only as affixes. E.g. prefixes like un-, dis- and suffixes like -ly, or -ness. These are called *bound morphemes*. Turkish suffixes -lük (gözlük - eyeglasses), or -cu (yolcu - passenger) are other examples of bound morphemes.

A morpheme may have alternate forms. E.g. negation morpheme may change depending on the like in-capable, il-legal, ir-regular, im-mobile. Similarly, Turkish morphemes frequently

change with harmonization rules like consonant or vowel harmony. E.g. past tense suffix -di may be transformed to -di (geldi), -dı (aldı), -du (oldu), -dü (gördü), -ti (içti), -tı (açtı), -tu (koptu), -tü (göçtü). These various forms of the same morpheme are known as *allomorphs*.

From the functionality point of view, morphemes can be studied in 2 distinct categories. *Derivational morphemes* form new words pointing to new concepts. E.g. the word *baker* is derived from the word *bake* by suffix of –er. Similarly in Turkish, the word *fırıncı* (fırın-cı – owen +AGT (baker)) is derived from the word *fırın* (fırın – owen) which forms a different word with a different meaning. On the other hand, *inflectional morphemes* do not change the meaning of the word. Plural suffix (Eng: -s, or Tur: –lAr) or case suffixes for nouns (e.g. dative, accusative, locative suffixes) are some examples of inflectional suffixes. The concept referred by the word remains the same. These morphemes are generally used to point out the role of word within the sentence. Tense suffixes of verbs are other examples for inflectional morphemes. A past tense suffix (Eng: -ed, Tur: -dı) adds a time information to the action indicated by the verb itself, without changing the meaning of the verb.

Morpheme structures are frequently affected by preceding sounds. So a morphological parser should handle these phonetic variations of affixation. Phonology studies the speech sounds and their patterns. Although human beings are capable of producing infinite number of sounds, a small portion of this set is used in languages. Some distinct phonological units are combined sequentially to form words. These basic building blocks of speech are called *phonemes*. In written text, these sounds are represented in the form of letters. These phonological units are roughly categorized into vowels (e.g. a, e, i, o) and consonants (e.g. b, c, d, f, g, …).

Since alphabetical letters are not interpreted in the same manner, frequently lexical representation (phonetic or function within a morpheme) of a phoneme may differ from its surface representation (expression by letters as a word).

Morphological analysis aims to detect and relate the structure of word forms i.e. morphemes, and derive some featural information about the form such as person, gender or count [5]. Morphological analysis also identifies some syntactic elements contributing to the meaning of the sentence. There a number of techniques to implement morphological analysis. Finite state transducers [6, 7] and regular expressions [8] are widely being used for morphological analysis.

Turkish as an agglutinative language is very rich in both inflectional and derivational suffixes. So, morphological analysis becomes one of the most important components in any attempt to process Turkish as a natural language.

2.1.2 Syntax and Semantic Analysis

Syntax studies grammatical structures and the order of the words with in a sentence. The meaning will be produced by combinations of words in a particular order.

Syntax analysis primarily endeavors identifying verbs, object and subject(s) of these verbs and the verb modifiers within the given sentence. This is usually achieved by parsing the sentence into a tree structure. Then, in the next step it regularizes this tree by omitting, merging or summarizing the items within the tree [4]. Syntactic analysis may be coupled to a semantic analysis, as a preparatory phase. .Syntactic analyzers also require a well defined lexicon for the particular free text.

Individual elements of a sentence are studied by lexicology. So it is the study of words, meaning and grammatical features of words for a given language or subset of a language. Words are the isolated building blocks of the language, and made up of one or more morphemes attached to each other either in the form of prefixes or suffixes. They play different functional roles in syntax to construct phrases and sentences.

Semantics studies the meaning of a word or sequence of words. Overall meaning for a word sequence may be unpredictable by examining individual words. E.g.

```
Sanki gök delinmişti.
(As if the sky was got a hole = there was a heavy rain)
```

Semantic analysis aims to determine what a sentence or sequence of words means. This problem is commonly handled as mapping ambiguous natural language with complex rules for interpretation into an unambiguous formal language, with simple interpretation and inference rules [9].

Traditionally, information extraction systems do not require morphologic analysis [10] and a deep semantic analysis [11].

2.2 Information Extraction

Information extraction (IE) is a subdiscipline of natural language processing, aiming to extract some required information from unstructured free texts, and store it in a structural way, so that the information becomes machine interpretable. Since the human communication and interaction primarily rely on natural language and free texts, area becomes very attractive for organizations interested in that embedded information.

IE has become a popular research topic since late eighties by the promotion of Message Understanding Conferences (MUCs) sponsored by Defense Advanced Research Projects Agency (DARPA) [12]. The MUCs have a great impact on the research on information extraction. Many new IE problems have been identified, and the algorithms are developed to solve these problems. The MUCs have helped the development of the evaluation metrics that are used in the comparisons of the information extraction systems participated in the competitions.

2.2.1 Message Understanding Conferences

Message Understanding Conferences (MUCs) have an important role in the development of information extraction as a field. These conferences were organized with the support of the Defense Advanced Research Projects Agency (DARPA) of United States, aiming to promote information retrieval and information extraction technologies [12]. The MUCs deeply influenced and shaped the methodologies and research directions in this field, and as a result, basic tasks in an information extraction system were defined in accordance with evaluations on given real world problems such as extraction of information from newspaper articles on a given topic like terrorist activities. After seven message understanding conferences between 1987 and 1998, tasks for an information extraction system were determined as: [13]

- Named Entity Recognition
  Process of identification of objects within the target text, such as organs, devices, particular tissues.
- Template Element Task
  Templates for identified objects. These templates contain different slots for objects of different classes. Identified objects fill in proper slots these templates.
- Template Relation Task
  Represents a special type slots. Usually, two slots for pointers to named entities identified and extracted in prior tasks. This task extracts the relations between the objects. E.g. kidney object and owned stones, or kidney object and masses identified in template element task, will have *ownership relation*.
- Co-Reference Resolution Task
  Identification of terms referring the same "named entities" or relations. When an object is pointed in multiple sentences, usually pronouns,

variants of names or abbreviations will be used to refer this particular object. Task of resolution of these references is called as co-reference resolution.

- Scenario Template Task
  This final task is the filling the slots of a structured target template with all named entities and relations extracted from the unstructured text.

These steps are closely related to understanding natural language itself. Traditionally, IE systems do not try a deep semantic analysis of all aspects of a text. They generally use pattern matching techniques such as finite state methods or regular expressions [14].

So, a typical information extraction system may have two main subtasks: *entity recognition* and *relation extraction*. Entity recognition tries to identify the boundaries of the text segments representing entities in natural language texts. For example, protein name extraction is an entity recognition task that tries to identify text segments representing protein names in medical texts. Relation extraction tries to identify the relations between entities in order to fill predefined templates. For example, the extraction of interaction relations among proteins is a relation extraction task. Both of these tasks use pattern matching techniques in order to extract the required information. The extraction rules that are generally regular expressions are applied to a given document in order to extract entities or relations.

A successful IE system at least relies to some degree on *domain knowledge* [15] and some level of *grammatical information*. All the facts, relations and implicit assumptions of the domain, which are required to identify semantic entities and extract the information within the text properly, must be conveyed to the IE system. The success of a system closely correlates with the coverage of the required domain knowledge which is made available to the system as data sources. The domain knowledge is very complex and covers all of our world knowledge for general natural language texts, and the complexity of the required grammatical information

for general natural language texts are complex as the whole grammar of that natural language. On the other hand, medical narratives are relatively easier to process from grammatical point of view because of their nature. Like many other technical subjects, medical texts also use a narrower subset of the language with limited number of information types [16], relatively unambiguous terminology [17] and predictable presentation patterns [18]. In other words, an information extraction system targeting a specific field such as medical texts which use a specific domain knowledge and sublanguage can be more successful than a general information extraction system because of the less ambiguity problem in those texts. Our information extraction system concentrates only on Turkish abdominal radiology ultrasonography reports that have less ambiguity problem, and its required domain knowledge is limited.

2.2.2 Supervised and Unsupervised Systems

There are two basic approaches for information extraction: a supervised methodology, also known as *Knowledge Engineering Approach*, and an unsupervised (or semi-supervised) methodology referred as Automatic Training Approach [11].

In the supervised approach, extraction rules are manually developed by a domain expert or a knowledge engineer in consultation with a domain expert. The system performance is affected by the performance of the knowledge engineer and/or the domain expert. The main disadvantages of these systems are difficulties in the adaptation to another domain, and the requirement of a domain expert for the domain knowledge. On the other hand, it is expected to have a higher performance in comparison to automatic training approach, as a consequence of human intelligence in the construction of the system parameters. The information extraction system described in this paper uses a supervised methodology, and its extraction rules and ontology are developed by a domain expert.

In the unsupervised approach, IE system is trained by means of an annotated training set data using statistical approaches. For example, after manual annotation of entity names, the text can be used to train the system on named entity recognition. During the training period, the system may interact with a user to test whether the extracted data is correct or not, so that it can fix its rules accordingly [11]. One of the major obstacles in IE is the manual adaptation of an IE system to a newer domain since the manual adaptation is a costly process. The manual adaptation requires recreation of rule-sets and templates on the basics of the new domain. The difficulty of the domain knowledge creation for a new domain is another limitation for the performance. As a consequence of these problems, machine learning techniques for information extraction are viable alternatives, and they are discussed as a research topic for information extraction [19].

2.2.3 Ontology-based Systems

As a term, ontology represents a specification of conceptualization, which is an abstraction of given universe for a particular purpose [20-22]. Originally, the term *Ontology* belongs to philosophy, denoting a systematic account of "Existence" [22]. As a computational method, an ontology represents entities for a given domain (or universe), and ontologies are frequently implemented in knowledge-based systems.

Ontologies are getting more and more popular to model knowledge in medical domain. OpenGalen is an initiative to create open source resources, which includes an ontology development environment and a large open source description logic-based ontology for the medical domain [23]. Several communities try to model radiological knowledge in the form of ontologies such as RadLEX [24, 25] and RadiO [26]. Witte et al also published an ontology for biomedical texts on the web [27].

In IE systems, it is claimed that the use of a formal ontology as one of the system's resources improves the performance of entity recognition and semantic annotation tasks [28]. There are

some published systems that use ontology during the information extraction task [29-34]. Ontology is utilized in several different tasks of these IE systems, such as semantic tagging at the named entity identification task [29-31, 34], and extracted data as final outcome [30, 32, 33].

An ontology may also be designed for a domain as a sharable knowledge component across different systems [20-22].

2.3 Related Work

Related works for TRIES constitute two major groups. One group is related to information extraction from medical texts and radiology reports in particular. Another group is the information extraction systems, attempting to use of ontology in one or more tasks of IE.

The *recall* and *precision* values are frequently used in evaluation of performances of information extraction systems. The precision is calculated as the ratio of the relevant findings in all findings of the system, whereas, the recall is the ratio of relevant findings within the total numbers of all expected findings.

2.3.1 Medical Information Extraction

After the initial introduction of information extraction approaches, the medical domain has become a popular application field for these systems. Many different research groups have emerged, mainly focusing on indexing reports as a free medical text search facility, automatic term coding such as diseases or physical findings, and detection of abnormal conditions such as disease findings. Recently, many medical IE extraction systems have been developed using different approaches, and some of them are discussed in this section.

Linguistic String Project (LSP) [16, 35] is one of the earliest supervised rule based systems aiming to extract data from medical narratives to populate predetermined template slots, aiming to improve search on these texts. The project is based on a subset of natural language so called *sublanguage*. Since medical narratives only use a subset of natural language, LSP aims to recognize the texts in this sublanguage and uses the patterns that are specific to the sublanguage to achieve information extraction without a complete language processing. Additionally, LSP tries to code entities using Systematized Nomenclature of Medicine (SNOMED). For their evaluation test set, LSP showed a performance of a recall value of 82.1% and a precision value of 82.5%. Our IE system also uses a sublanguage that covers the sentence structures used in Turkish radiology reports.

Haug describes Special Purpose Radiology Understanding System (SPRUS) for the extraction of coded findings from free-text radiologic reports, and the evaluation results for the prototype system are reported as 87% recall and 95% precision [36]. The system mainly relies on semantic approach rather than syntactic methods. SymText is developed on top of SPRUS, extending its functionality to syntactic analysis of the text with different statistical methods [37, 38]. SymText is evaluated with the reports of acute pulmonary embolism patients. 92% recall and 88% precision values are achieved for making a diagnosis in chest radiography reports [39]. Our IE system also targets the radiology reports in Turkish.

Medical Language Extraction and Encoding System (MedLEE) [40, 41] has been developed to extract clinical information from clinical texts by Freidman. Its initial application domain was radiology reports. The system used a controlled vocabulary to code entries. The initial evaluation of this rule based system resulted in 85% recall and 87% precision results. Later Hripcsak evaluated MedLEE to use the coded data for automated decision-support [42]. The system was tested for identification of six medical conditions from radiology reports. Recall and precision were found to be 81% and 98% respectively. Freidman et al also adapted the system to biomedical texts with the name of GENIES, aiming to extract molecular pathways from journal articles [43].

MENELAS is a multilingual medical language system (French, Dutch, and English), primarily focusing on discharge summaries and coding diseases using International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM) [44, 45]. It's rule based system utilizing ICD concept tree with morphological analysis. The overall recall and precision results are measured at 48% and 63% on the coding task, and 66% and 77% on the questionnaire task, respectively.

MedSyndikate is developed to extract medical information automatically from findings reports in German language [46]. It uses a semi-automatic tool to acquire the domain knowledge. Its recall and precision values are found to be 93%. Recently, Mykowiecka et al. have developed a rule based IE system for medical narratives in Polish [47]. The system uses a syntactic parser and relies on ontology for named entity recognition. Its recall and precision values are over 80%.

Berrut reported an information retrieval system named RIME for indexing medical reports in French [48]. At the indexing step, system translates each sentence into a tree of concepts by using a formal grammar that is expressing hierarchical definition of the domain knowledge. RIME also proposed a morphological analysis coupled to syntactic analysis.

Bekhouche et al presented another architecture for an IE System for medical texts in French [49]. System mainly focused on annotation of symptoms, diagnoses, procedures or other items from a given terminology such as International classification of the diseases 10 and Common Classification of the Medical Acts (CCMA). The system implemented as a rule based manner without a morphological analysis.

RADA was another supervised rule based IE System for radiology reports in English, which was developed by Johnson et al [50]. Lexical analyzer of RADA was not differentiating the morphological variations of the words. RADA used glossaries from two main sources, the Unified Medical Language System (UMLS) and a specialized thoracic glossary. Authors

reported evaluation results on different entities with a recall range of 37-87% and a precision range of 65-100%.

Amaral and Satomura published an approach to transform medical texts into structured information in physical databases [51]. Although they did not call their system as an IE system, they created some conceptual semantic patterns that might occur in medical texts, and recognized portions carrying semantic information. They used different axes (diseases, findings, …) of SNOMED as a source for concept definitions, and extracted those information into database tables. They established following results: 73.41% of the sentences were formatted; 81.05% of the analyzed words were identified; and 95.33% of the medical terms were indexed.

2.3.2 Ontology based Information Extraction

Textpresso is an ontology based system, mainly aiming to index biomedical papers for better information retrieval from literature [29]. Ontology is used for term tagging and clarifying the underlying semantics – terms and relations among them – for the domain of interest. It has an overall performance of 94.7% and 30.4% for recall and precision in keyword search in full texts, whereas the same values are 44.6% and 52.3% in abstract search.

Embley et al. developed an ontology based system to extract information from unstructured data rich documents [30]. Ontology is both used in entity recognition task of information extraction process, as well as the final structure of the extracted data. System built as a supervised rule based system. However, there was no support for semantic functionality of morphological variances. Authors reported recall and precision ratios of 0.98 and 0.995 respectively for extraction of model names from car ads and to 0.99 and 0.99 respectively for job names from computer job listings in their tuning sets.

Vulcain is another ontology based IE system published by Todirascu et al [31]. System was designed to filter domain specific messages. An ontology that was developed by a domain expert was used in the validation of indentified concepts at entity recognition task. The performance of this rule based system was not evaluated.

Buitelaar et al. describe an ontology based system named as SOBA, focusing on extraction of sports events from soccer web sites [32]. The system transforms linguistic annotations into an ontology based representation, so that resources crawled from different web sites can be integrated to form a knowledge base. A morphological analysis was performed during the syntactic analysis. The performance of this rule based system was not evaluated, either.

Wood et al published MultiFlora [33], which was a rule based unsupervised IE system, extracting information from plant related texts in the domain of botany. Extracted information was populated to an ontology. Authors reported a 50% false positive rate.

OntoSyphon was an IE system aiming to identify possible semantic instances, relations, and taxonomic information from the web pages by use of a given ontology by an ontology-driven manner [34]. System parses web content to extend its ontology in an unsupervised manner. An experimental evaluation by authors yielded a 50% recall.

Maedche et al proposed a bootstrap of a rule based IE system using ontology for domain knowledge [52]. But, there was no further information about the implementation or evaluation of the system.

Erozel published a system to query natural language video databases [53]. If an exact match was not found, WordNet ontology was utilized to measure similarity of the concepts by means of the distance between them.

RadLEX is a general purpose radiological ontology aiming to model radiological anatomy [24, 25]. Although conceptually, RadLEX has overlapping parts with TRIES ontology, it does not

cover every aspect of a single examination. Primary intention of TRIES is to model the information content of a complete study in a compact ontology.

RadiO was developed as prototype application ontology to close the gap between radiology reports and RadLEX [26]. But, the main purpose of RadiO was structured entry of radiological reports.

Chapter 3

TURKISH RADIOLOGICAL INFORMATION EXTRACTION SYSTEM (TRIES)

TRIES is an information extraction system aiming to parse free text Turkish radiological reports into computationally usable structured information. Although the system was designed to process all kinds reports from different types of radiological examinations such as magnetic resonance imaging, computerized tomography and plain X-Rays, prototype presented here uses ultrasonographic reports as input. The major components of TRIES are given in Fig. 1. All the words in a given report are analyzed by a Turkish morphological analyzer. Each word is converted into a sequence consisting of a root word followed by possible morphemes. Morphological analyzer uses a lexicon, which is the source of lexical information for a set of Turkish root words. All the possible root words that can be seen in radiology reports are available in the lexicon The words in the lexicon are grouped according to their functional properties of words such as verbs, nouns, adjectives, as well as abbreviations (e.g. units – mm, cm, ml, cc, mgr). In case of any failure during morphological analysis of a word in the report, the spell-corrector is invoked in order to fix a possible typing error. The fixed word returns back to morphological analyzer.

After the morphological analysis, a sentence can be seen as a sequence of root words and morphemes. Then, the entity recognition module recognizes some substrings of the sentence as terms, and marks them as a named entity term such as an ontological concept, an attribute, or an attribute value (Fig. 2).

TRIES ontology is designed at the conceptual level. The verbal representation of each ontological concept is maintained as a terminology attachment to conceptual ontology. These

terms are commonly represented by morphological structures to let term analyzer to distinguish the morpheme belongs to term itself and the morphemes related to syntactic structures. In a sentence like

```
Safra kesesinde 3 mm taş izlendi.
Gall bladder   3 mm stone observed
(A stone of 3 mm was observed in gall bladder.)
```

Morphological analysis of *kesesinde* will yield *kese* +POSS3SG +LOC (bladder +POSS3SG +LOC). During term analysis, the terminology part of ontology provides the Turkish term safra kese+POSS3SG as a representation of *GallBladder* entity. The remaining morphemes are attached to the newly formed term to be processed further during rule extraction such as *GallBlader* +LOC for the above example. So, the morphemes taking place in the formation of a named entity term are merged, and they are treated as a single unit after the entity recognition phase. The remaining morphemes are kept as modifiers. Turkish strings that can be named entity terms are determined with the help of the knowledge stored in the terminology part of TRIES ontology.

In the next step, a sentence is processed by the relation extractor to match against TRIES rule templates, and the semantic information in the sentence is extracted as a set of relations (Fig. 3). In the definition of the rule templates, the entity terms appearing in ontology are used in order to have more flexible rules. Rule templates may also utilize morphological elements to capture semantics gained by natural language grammar. So a typical rule template is made up of ontological concept elements and syntactic elements that are bound by regular expression elements.

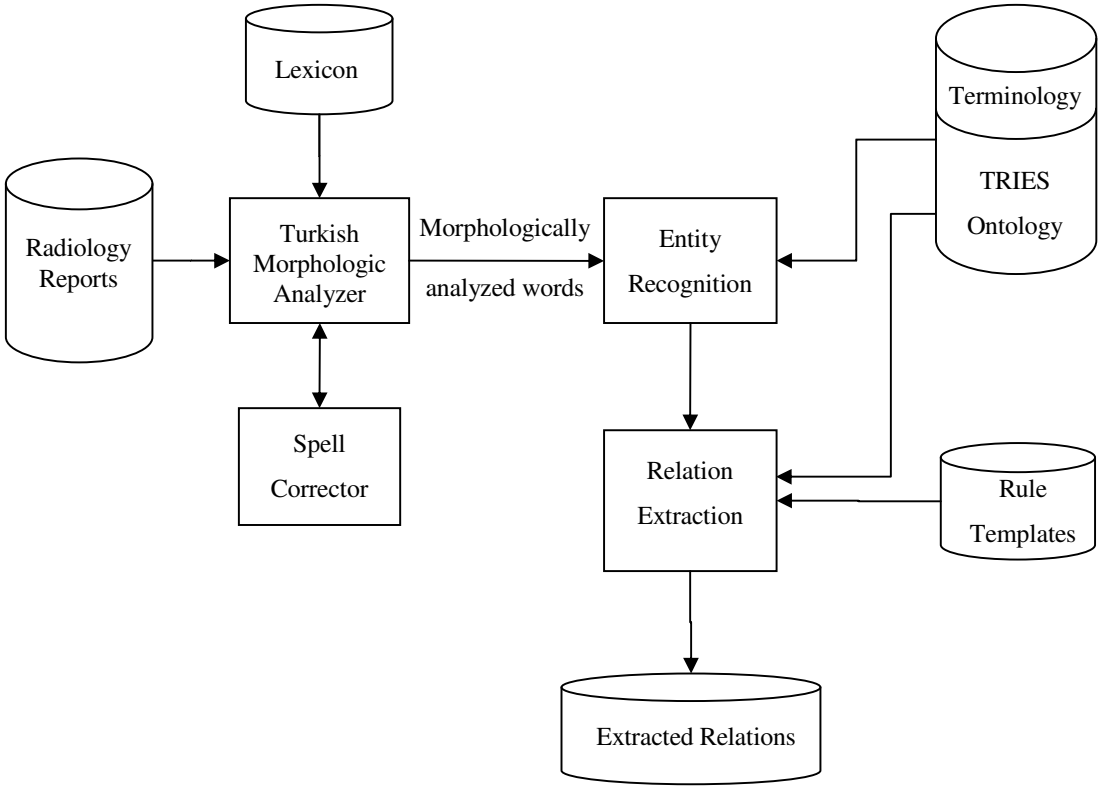

Fig. 1. Components of Turkish Radiological Information Extraction System (TRIES)

Table 1 gives the steps of TRIES which is applied to a sample sentence. After morphological analysis, the sample sentence can be seen as a sequence of root words and morphemes. The TRIES entity recognition module recognizes the root word "*karaciğer*" (liver) as the ontological concept "*Liver*", the morpheme sequence "*vertikal uzun +NESS*" (height) as the attribute "*height*", and the sequence *"14 cm"* as an attribute value of "*NUMERIC*" type.

Table 1. Application of TRIES to a sample sentence. (POSS3SG: Possessive suffix for 3rd singular person, NESS: -ness suffix, COP: copula)

Text
Karaciğer vertikal uzunluğu 14 cm'dir. The height of liver is 14 cm.
Morphological Analysis
Karaciğer vertikal uzun+NESS+POSS3SG 14 cm+COP Liver vertical tall+NESS+POSS3SG 14 cm+COP
Named Entity Recognition
[Karaciğer] [vertikal uzun+NESS] +POSS3SG [14 cm] +COP [entity:Liver] [attribute:height] +POSS3SG [value:NUMERIC: 14 cm] +COP
Relation Extraction – rule to be matched, and rule constraints to be satisfied:
<VisibleStructure O> <O:Attribute A> +POSS3SG <O:A:Value V> +COP obj_has_attribute(Object, Attribute) – (Liver, height) obj_attribute_accept_value(Object, Attribute, Value) – (Liver, height, 14 cm)
Extracted Relation
Liver.height = 14 cm

After entity recognition, if the sentence matches a rule template and satisfies its rule constraints, a set of relations is extracted from that sentence. In our example, the entity "*Liver*" matches the entity "*VisibleStructure*" in the rule template since "*Liver*" is a sibling of "*VisibleStucture*" according to TRIES ontology. The attribute "*height*" matches the attribute field in the template, and it satisfies the rule constraint since "*height*" is an attribute of "*Liver*" according to our ontology. Similarly, the string "*14 cm*" matches the value field in the template, and it satisfies the rule constraint since the "*height*" attribute of the "*Liver*" entity accepts a numeric value as its attribute value. The relation *"Liver.height = 14 cm"* is extracted

ontology also determines the structure of its information model. The usage of ontological concepts in the extraction rules increased their flexibility. TRIES ontology is also used in the reference resolution problem in order to determine missing entities and attributes in sentences. To the best of our knowledge, TRIES is the first Turkish medical IE system. TRIES achieved 93% recall, and 98% precision results.

Chapter 4

TRIES: TURKISH MORPHOLOGIC ANALYSIS

Turkish language is an agglutinative language, and it has very rich morphological structures. Many grammatical functions are represented by affixes in Turkish [54]. Since English language does not have such complex morphological structures, many NLP systems do not use morphologic analysis. On the other hand, the usage of the morphological analysis in Turkish systems increases their flexibility. In our IE system, recognizing morphemes enables it to handle words much more flexibly [10]. For example, the place of a single accusative morpheme determines the whole meaning of the sentence in the following sentences.

Doktor hastayı muayene etti (The doctor examined the patient)
Doktor hasta+ACC muayene et+PAST (Doctor patient+ACC examine+PAST)

Doktoru hasta muayene etti (Patient examined the doctor)
Doktor+ACC hasta muayene et+PAST (Doctor+ACC patient examine+PAST)

TRIES has a Turkish morphological analyzer that looks like a PC-Kimmo [55] based morphological analyzer. As an initial preparatory step, morphological analyzer tokenizes the sentence into tokens. At this step, words, symbols, numeric expressions and punctuation marks are identified and marked by means of regular expressions. Then the words are taken into the analyzer. The morphological analyzer uses finite state methods (FSM) and its own restricted lexicon that is generated from the ultrasonography reports repository. We explicitly used a restricted lexicon for the morphological analyzer in order to reduce the amount of ambiguity. This analyzer parses a given word into possible morpheme combinations using its

own lexicon. The lexicon provides the word roots together with their part of speeches such as noun, adjective, verbs, abbreviations, units, etc. The morphological parser can handle Turkish specific phonological rules such as vowel harmony, consonant softening and consonant doubling, and it uses a PC-Kimmo compatible phonological rules that are compiled by KGEN component of PC-Kimmo. It can also identify the different Turkish specific suffixes and use morphotactic rules in order to determine the morpheme sequence, based on the functional role of the word obtained from the lexicon.

4.1 Turkish morphological rules

Many affixes are used, like possessive suffix, locative suffix, tense suffixes etc. Suffixes may have different allomorph, and the type of the allomorph used is affected by preceding phonemes. These rules include vowel harmony, consonant dropping, buffer deletion, dropped vowel. For instance, allomorphs for *locative suffix* include "de", "da", "te", "ta". Selection of the allomorph to be used will be influenced by prior vowels and consonants: "ev-de", "okul-da", "iş-te", "sokak-ta" [56, 57].

TRIES lexicon uses *two level phonology* paradigm to handle this problem [55]. Lexical representation of a phoneme may be different than its surface representation, frequently. These transformations are coded in the form of a rule-set. The lexicon of TRIES contains this information for any individual word. In TRIES morphological analysis, letter A represents an unround-open vowel (i.e. either "a" or "e") depending on the vowel harmony. E.g. surface representation of a plural suffix may be either *-ler* (pencereler –windows–) or *-lar* (kapılar –doors–) influenced by prior phonemes. In TRIES lexicon, plural suffix is represented as "-lAr" and functional letter capital A is handled by following rules:

```
RULE A:e      [:Ve | L: ] :SCONS * _
RULE A:a      :Va :SCONS * _
```

where *Ve* is equal to a front vowel (e, i, ö, ü), *Va* is any of the back vowels (a, ı, o, u), SCONS is a surface consonant, and L is the lexical representation of soft (palatalized post-alveolar) *l*. Based on the above two level rules, word *pencere +lAr* is processed as follows:

```
Lexial representation : p e n c e r e + l A r
                        ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation: p e n c e r e 0 l e r
```

Full set of these two level transformation rules can be found on Appendix C.

There are several rules affecting successive suffixation namely vowel harmony, consonant harmony, consonant softening and buffer characters. Additionally, some suffixes may change the root of the word like vowel deletion. Other rules changing the root of the word are consonant softening, consonant doubling rules. Morphology analyzer component of TRIES can handle these variations.

4.1.1 Vowel harmony

Aim of vowel harmony is to reduce the muscular effort required for the generation of sounds by tongue and lips, which can be summarized as follows: [56, 57]

- The first back or front vowel of a word is followed by the same type of vowels (*major vowel harmony*).
- An unround vowel is followed by unround vowels. But, round ones are followed either by round-closed (u, ü) or unround-open vowels (a, e) (*minor vowel harmony*).

In Turkish, vowel harmony is very important from morphological aspects, since it primarily affects formation of the proper allomorph. Even if the two words are suffixed by the same suffixes, they may sound different based on vowel harmony. E.g. *ev-ler-in-de* (house +PL +PERS3PL + LOC, in their houses), or *okul-lar-ın-da* (school +PL +PERS3PL + LOC, in their schools). Similarly, *gel-di-ler* (come +PAST +PERS3PL), *gör-dü-ler* (see +PAST +PERS3PL), *al-dı-lar* (take +PAST +PERS3PL).

Table 2. Classifications of Turkish vowels based on different functional criteria

Round / Unround	o, ö, u, ü / a, e, ı, i
Back / Front	a, ı , o, u / e, i, ö, ü
Open/Closed	a, e, o, ö / ı, i, u, ü

Turkish alphabet has 8 vowel letters. These letters are frequently classified based on their *roundness*, *frontness* and *openness* (Table 2).

In TRIES lexicon, capital letter A represents a lexical representation, which will turn to proper vowel based on former phonemes. Another example, letter H represents closed vowels u, ü, ı or i depending on vowel harmony. In addition to surface transformation of A:a and A:e, letter H will be transformed to a closed vowel (ü, u, i, ı) in concordance with vowel harmony. Since H is both affected by major and minor vowel harmonies, if the word ends with a round vowel, Corresponding rules are expressed as follows:

```
RULE H:u    [ Vbkrd:0 | :Vbkrd ]  :SCONS * _
RULE H:ü    [ Vrd:0 | :Vrd | L: ] :SCONS * _
RULE H:ı    [ Vbk:0 | :Vbk  ]    :SCONS * _
RULE H:i    [ Vi:0 | :Vi  | E: | L: ] :SCONS * _
```

Vowel harmony is one of the most important rules effecting selection of allomorphs.

Table 3. Some examples for different phonemes represented with the same surface representation.

Surface Representation	Phoneme source
K	**k**ilit(lock), **k**apak(cover)
H	i**h**mal(ignore), ka**h**ve(cofee), **h**ayat(life)
E	**e**rgin(mature), **e**lma(apple)

Turkish alphabet is known as a phonetic alphabet composed of 29 letters, thus, it's read as written. But reality, although Turkish alphabet is alphabet composed of 29 letters, there are more than 35 phonemes. So, some letters represent more than one phoneme, i.e. different phonemes within different words (Table 3). Although this difference may not be so important from linguistic aspects, in majority of cases this helped us to formulize some morphological irregularities, so called *irregular words*.

Table 4. Sample irregular words. These do not obey classical vowel harmony.

Use in speech	Expected
medialinde	Medialında
's' harfi, harfler	s harfı, harflar
saatlik	saatlık
alkollü	alkollu
goller	gollar

Many words that do not obey the rules like vowel harmony are accepted as irregular words (Table 4), especially for the words with foreign origin. Whatever the surface representation of the phoneme is, these letters are phonetically different. For instance, one can easily recognize the spelling difference between *l* (velarized dental) and *soft l* (palatalized post-alveolar) for the words "banal" and "banâl", which arises from the different location and position of the tongue even if they are represented by the same *letter l*. Furthermore, these different phonemes participate to vowel harmony, even if they are not vowels: "banaldı" and "banâldi". Since the suffixes are affected by vowel harmony, these phonemes must be distinguished during a successive morphological analysis. So, TRIES had to analyze the words at the phonological level. These phonemes were also represented in TRIES lexicon such as "mediaL" where capital letter represents soft L in lexicon and converted to "l" during surface representation of the word (i.e. L:l). So that, morphological analyzer of TRIES knows that, word "medialde" will be split into morphemes of root word "medial" and locative suffix "DA" and, phoneme L is taken into account during the selection of proper allomorph. For example, the word "medialde" will be handled as follows:

```
Lexial Representation :  m e d i a L + D A
                         ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:  m e d i a l 0 d e
```

Following rule will be executed since morphological analyzer first meet "L" as lexical representation (i.e. L:) within the word mediaL:

```
RULE A:e      [:Ve | L: ] :SCONS * _
```

As soon as morphological analyzer meets letter L, this rule satisfies and lexical A is transformed to letter e at the surface.

Table 5. Turkish consonants

Voiceless	p, t, s, ş, k, f, ç, h
Voiced	b, d, z, j, g, v, c, l, ğ, m, n, r, y

4.1.2 Consonant Harmony

Twenty one of 29 letters of Turkish alphabet are consonants. These consonants can be classified based on voicing properties as *voiceless* and *voiced* (Table 5).

Allomorphs in Turkish language are frequently affected by voicing properties of the root. If the last consonant of the root is voiced, consonants of following suffixes are transformed into voiced counterparts. If the last consonant is voiceless, then, suffixes are transformed to use voiceless consonants. E.g.

```
işçi = iş + INVL (work + INVL = worker)
oduncu (odun + INVL, wood + INVL = lumberjack)
```

Similarly,

```
okulda = okul + LOC (school + LOC)
işte = iş + LOC (work + LOC)
```

In above example, following 2 rules determines how the lexical element D in locative suffix +DA will be represented in its surface form:

```
RULE D:t [ :h | :ç | :s | :ş | :k | :p | :t | :f ] +:0  _
RULE D:d [ :b | :c | :d | :g | :ğ | :j | :l | :m | :n | :r | :v
| :y | :z | :SVOWEL] +:0 _
```

So based on these 2 rules and A:e transformation mentioned above, the word formation okul+DA (okul+LOC, school+LOC) will be processed as

```
Lexial Representation :   o k u l + D A
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   o k u l 0 d a
```

and the word formation iş+DA (iş+LOC, work+LOC) will be processed as

```
Lexial Representation :   i ş + D A
                          ↓ ↓ ↓ ↓ ↓
Surface representation:   i ş 0 t e
```

4.1.3 Buffer characters

In Turkish morphology, joining of suffixes beginning with a vowel to a root ending with a vowel requires a buffer consonant. Letters y, n and s are used for this purpose:

kapı (door) + ACC = kapı + YH = kapıyı

Capital Y triggers one of these two rules

```
RULE Y:y    VOWEL:SVOWEL +:0 _
RULE Y:0    CONS:SCONS  +:0  _
```

And, following word formation takes place on kapı + YH

```
Lexial Representation :   k a p ı + Y H
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   k a p ı 0 y ı
```

On the other hand, top + YH (ball + ACC) will be processed as

```
Lexial Representation :   t o p + Y H
                          ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   t o p 0 0 u
```

Y buffer is used in widely in many suffixes such as DAT (+YA), FUTURE (+YAcAk), and INS (+YlA).

Capital N is a similar buffer character, which is interpreted by following rules:

```
RULE N:0    :SCONS +:0 _
RULE N:n    :SVOWEL +:0 _
```

E.g. in genitive suffix as +NHn:

```
kapı-nın rengi (door+GEN color + POSS3SG)
ev-in rengi (house+GEN color + POSS3SG)
```

Capital Z another is buffer consonant, which is managed by this ruleset:

```
RULE Z:s    VOWEL:SVOWEL +:0 _
RULE Z:0    CONS:SCONS +:0 _
```

E.g. in suffix POSS3SG as +ZH:

```
kapı-sı (door + POSS3SG)
ev-i (house + POSS3SG)
```

4.1.4 Vowel Deletion

In Turkish, suffixations may change the root of the word. In vowel deletion, for some suffixes, if the root is ending with a vowel and the suffix starts with a vowel, vowel of the suffix may replace the end vowel of the root. As an example, present continuous tense suffix +Hyor is an example for this rule:

```
kapa - kapıyor (close - close+PRESCONT+PERS3SG)
ye - yiyor (eat - eat+PRESCONT+PERS3SG)
de - diyor (say - say+PRESCONT+PERS3SG)
```

This rule is expressed in TRIES morphology ruleset as:

```
RULE SVOWEL:0  _ +:0 H y o r
```

Word koş(run)+PRESCONT is processed as follows:

```
Lexial Representation :   k o ş + H y o r
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   k o ş 0 u y o r
```

On the other hand, word kapa +PRESCONT will be processed as:

```
Lexial Representation :   k a p a + H y o r
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   k a p 0 0 ı y o r
```

since the pattern satisfies the rule vowel deletion above.

4.1.5 Consonant Voicing

When the words ending with a voiceless consonant (p, t, ç, k) are suffixed with a suffix starting with a vowel, consonant will be transformed to voiced counterparts (b, d, c, and g or ğ) correspondingly.

TRIES morphological analyzer applies following rules for these cases.

```
RULE {K, T, P, Q, J}:{ğ, d, b, g, c}    _ +:0 :SVOWEL
RULE {K, T, P, Q, J}:{k, t, p, k, ç}    _ [ # | :SCONS ]
```

E.g. böbrek (kidney) + ACC

```
Lexial Representation :   b ö b r e k + Y H
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   b ö b r e ğ 0 0 i
```

E.g. böbrek (kidney) + ABL

```
Lexial Representation :   b ö b r e k + D A n
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   b ö b r e k 0 t e n
```

kitap (book) + DAT = kitap + YA

```
Lexial Representation :   k i t a p + Y A
                          ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Surface representation:   k i t a b 0 0 a
```

4.1.6 Consonant Doubling

Adding a suffix beginning with a vowel to some words from Arabic origin like *hat* (line), *hac* (pilgrimage), *hak* (right), the final consonants will be duplicated: hattı (hat+ACC), hacca (hac+DAT), hakkı (hak+ACC), hakka (hak +DAT). In TRIES lexicon, these words are marked with a caret sign at the end like hak^ and hat^. Following morphological rulesets are applied to these word roots:

```
RULE ^:0      _ #
RULE ^:0      _ +:0 @:SCONS
RULE ^:r      r _ @:SVOWEL
RULE ^:t      t _ @:SVOWEL
RULE ^:k      k _ @:SVOWEL
RULE ^:f      f _ @:SVOWEL
```

4.2 Turkish Morphotactic Rules

For a meaningful suffixation, suffixes in Turkish must follow a particular order, depending on the type of the root word like noun, adjective and verb.

```
böbrektekiler ( böbrek (noun) +locative(case) +relative +plural)
the ones located inside the kidney
böbreklerdeki ( böbrek (noun) +plural +locative(case) +relative)
the one located inside the kidneys
böbreklerkide ( böbrek (noun) +plural +relative +locative(case))
meaningless
```

Some suffixes may transform the type of the word they attached, and, derived word will follow the path of new word type. These morphotactic rules for nouns, verbs, and adjectives are summarized in Fig. 4, Fig. 5, and Fig. 6, correspondingly.

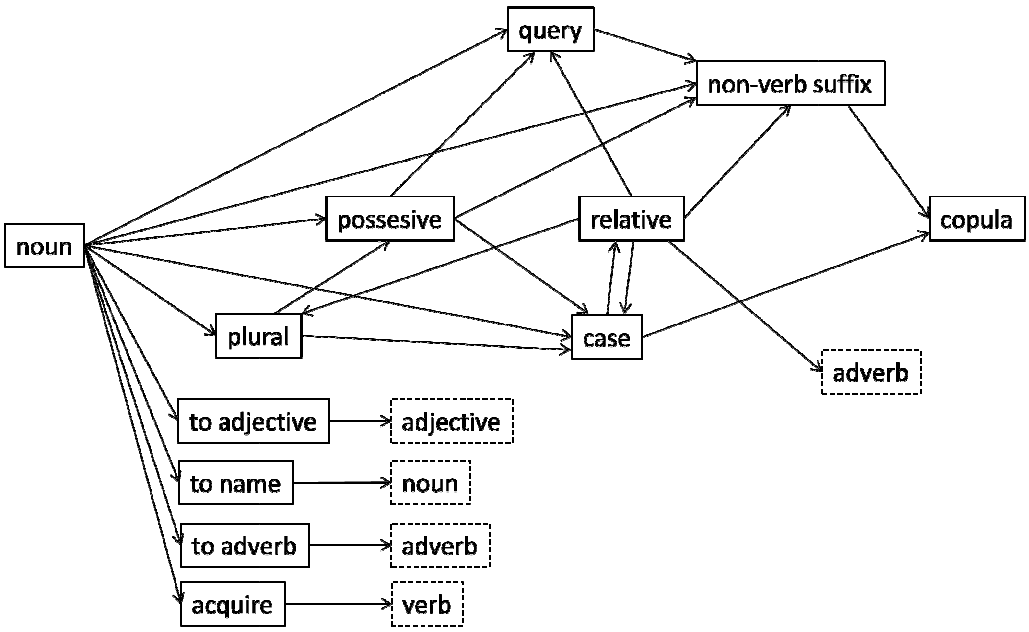

Fig. 4. Morphotactic rules for nouns

A noun can derive to yield other word types (Fig. 4) like adjectives (tuz (noun, salt), tuz+WITH (adjective, salty)) or verb (taş (noun, stone), taşlamak (verb, to throw stone)). Derived word behaves obeying the morphotactic rules of its new type. However, inflectional suffixes do not change the nature of the word. In this case, suffixes must follow a certain order. It's possible to produce meaningful words as long as the path given is followed:

```
Böbrek+ler+im+de+ki+ler+den+mi+dir+ler+ki
(Kidney+PL+POS1SG+LOC+REL+ABL+QUEST+COP+PL+REL)
Are those the ones of which the ones in my kidneys?
```

Suffixation which does not follow this pathway will be meaningless.

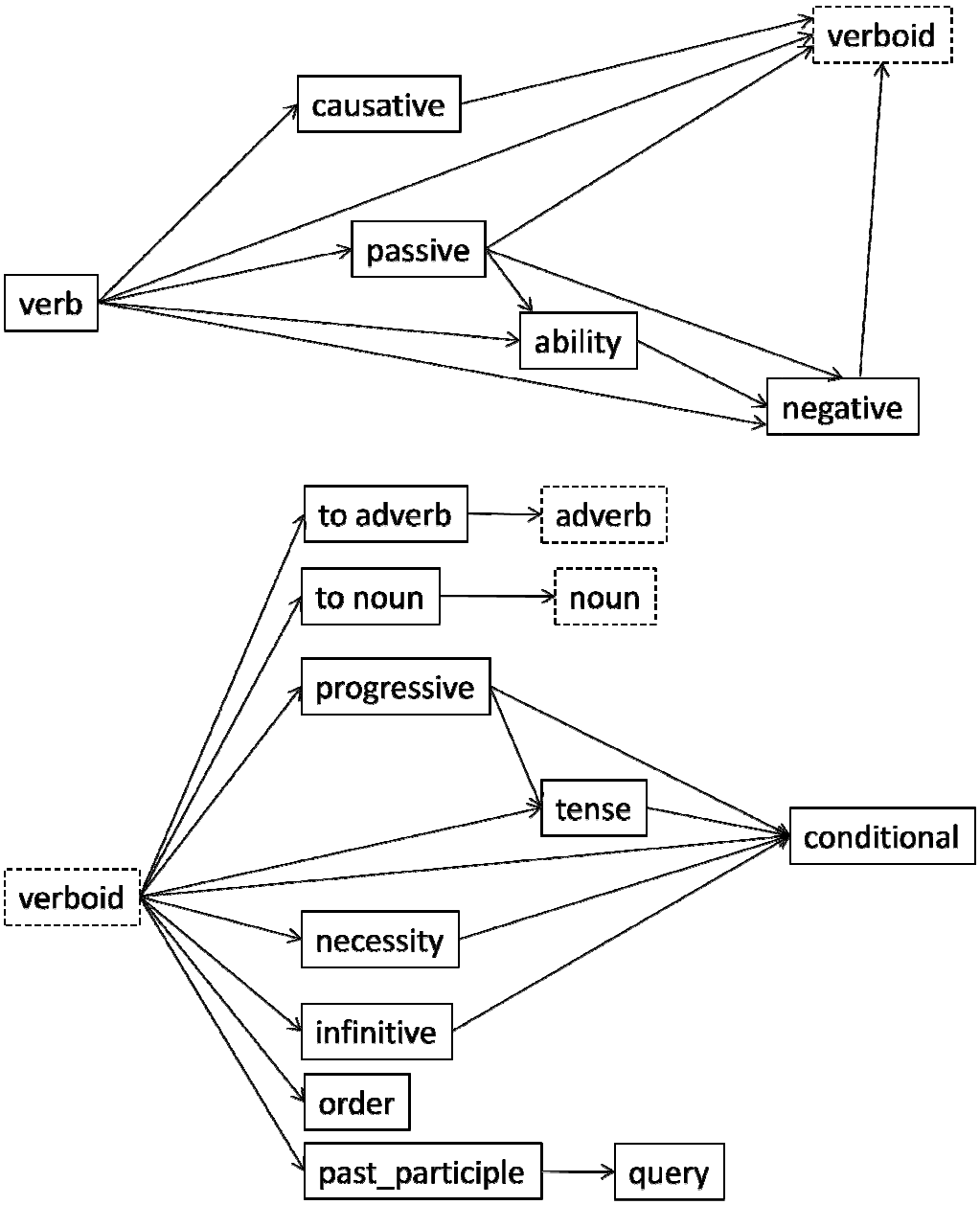

Fig. 5. Morphotactic rules for verbs.

Verbs have the most complicated morphotactic rules among other word classes. Tense suffixes, personal suffixes, question/passive/request forms all managed by suffixes attached to verbs. Almost all other classes of words can derive to yield a verb. Similarly other verb classes can be derived from verbs. Morphotactic rules for verbs are given in Fig. 5.

Morphotactic rules for adjectives are relatively simpler (Fig. 6). These words can derive other word classes. These derivations sometimes implicitly occurs. E.g.

```
Sarılar (sarı+ '' +lAr)  = yellow ones
(Yellow + TO_NOUN + PL)
```

Adjective "yellow" functionally transformed into a name and then, the word can exactly be used as a noun, following the morphotactic path of nouns given above.

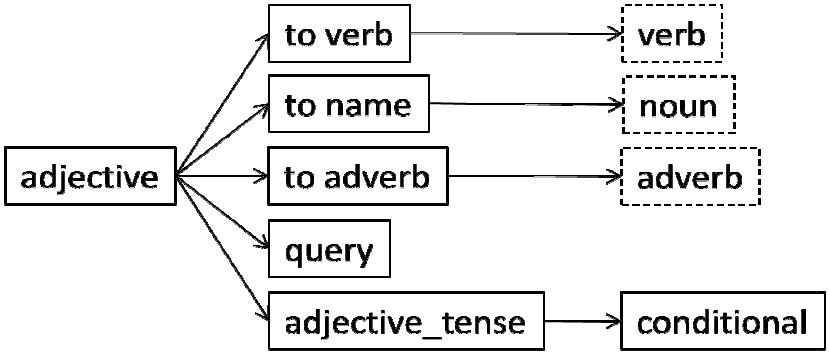

Fig. 6. Morphotactic rules for adjectives

Proper suffixation with morphotactic rules generates endless combinations, frequently deriving different word classes from the words:

```
Sarılaşanlı = the one with the one becoming yellow
Sarı   + laş    + an      + lı
Yellow + BECOME + TO_NOUN + INS
ADJ       VERB    NOUN      ADJ
```

4.3 Error handling

The morphological analyzer is tightly coupled to a spell-corrector, so that it can fix some simple typing errors such as a missing letter, an extra letter, or two transposed letters. This integrated spell-corrector algorithm is developed to overcome typing errors that can break the pattern recognition tasks that are used during entity recognition or relation extraction. This integrated spell-corrector helped to improve the performance of our IE system.

In Turkish, the average number of morphological parses for a given word is 2.5. As a side effect, the morphological analysis introduces ambiguities [58]. The usage of the restricted lexicon in our morphological analyzer reduces the ambiguity problem for our system. Although we have a reduced ambiguity problem, still there are morphologically ambiguous words in our sentences. A separate sentence is created for each of the morphological parse combinations of the words, and they are processed by the other steps of TRIES in order to extract templates.

Chapter 5

TRIES: INFORMATION MODELLING

5.1 Ontology

Domain knowledge is one of the most important components of information extraction systems. Free text information frequently relies on many concepts as well as explicit or implicit relations among them. Concepts and concept attributes correspond to named entities within the text. It is usually easier to incorporate these named entities into the system to utilize during processing of the text, by means of vocabularies or lexicons. On the other hand, relations are usually difficult to be modeled and often implicit. Highly technical documents such as medical documents are rich in implied knowledge. Writers of such texts usually assume that the reader already has some obvious domain knowledge. These hidden relations and knowledge content frequently affect the meanings of free texts. If these are ignored by information extraction systems, then the output of the system will be crippled.

The ontology is a formal specification of a shared understanding of the domain of interest [20], and it is getting more popular to share knowledge across the systems [21, 22].

In terms of ontological entities, a reasoning process is conducted through relationships:

```
Kidney IS_A SolidOrgan
SolidOrgan HAS_A Parenchyma
 → Kidney HAS_A Parenchyma
```

TRIES ontology is created by examining 756 abdominal ultrasonography reports consisting of 11780 sentences in order to model the abdominal region organs that appear in the reports. The ontology currently contains 135 hierarchical entities with possible 70 attributes. In addition to entities and attributes, the ontology contains the terms that can be possible values for attributes. In TRIES ontology, currently there are 740 terms, and these terms are associated with a set of Turkish strings to indicate their representations in Turkish. In order to achieve this, a list of terms is maintained as an appendix to ontology in the form of a terminology server. Terms denoting concepts of the ontology (e.g. entities and attributes) including the synonyms of concepts are maintained in this terminology.

TRIES ontology entities implemented two types of relations. The former one, "Is a" relation creates the skeleton of TRIES ontology (Fig. 8), which is closely correlated to target information model for the extracted information. On the other hand, the next relation type is a family of relationship that helps to create parent-child relationships. The parts of the entities and other owned entities are linked to parent entity by means of a corresponding relationship specialized to for the target entity such as has_lobe, has_cyst or has_mass. By definition, these relationships may require varying instances for that particular entity class (e.g. one to one or zero to many). This approach simplifies the relationship of ontology and information model, and the semantics of represented information. Furthermore, it plays an important role in the validation process of rule constraints.

TRIES ontology is created using Protégé ontology creation tool (Fig. 7) [59]. Entities inherit particular attributes in an *is_a* hierarchy. Entity-entity relationships other than *is_a*, are maintained by slots. For example, *Kidney* has several attributes inherited from its parent entities, and, it also defines its own specific attributes. The *parenchyma* and *cyst* attributes of *Kidney* can be seen as the examples of specialized *part_of* relations. *Kidney* can have a single instance of *Parenchyma* (1 to 1), and, it can also have multiple instances of *Cyst* (0 to many). These slots host proper instances of these entities at during rule extraction, satisfying the rule constraint conditions.

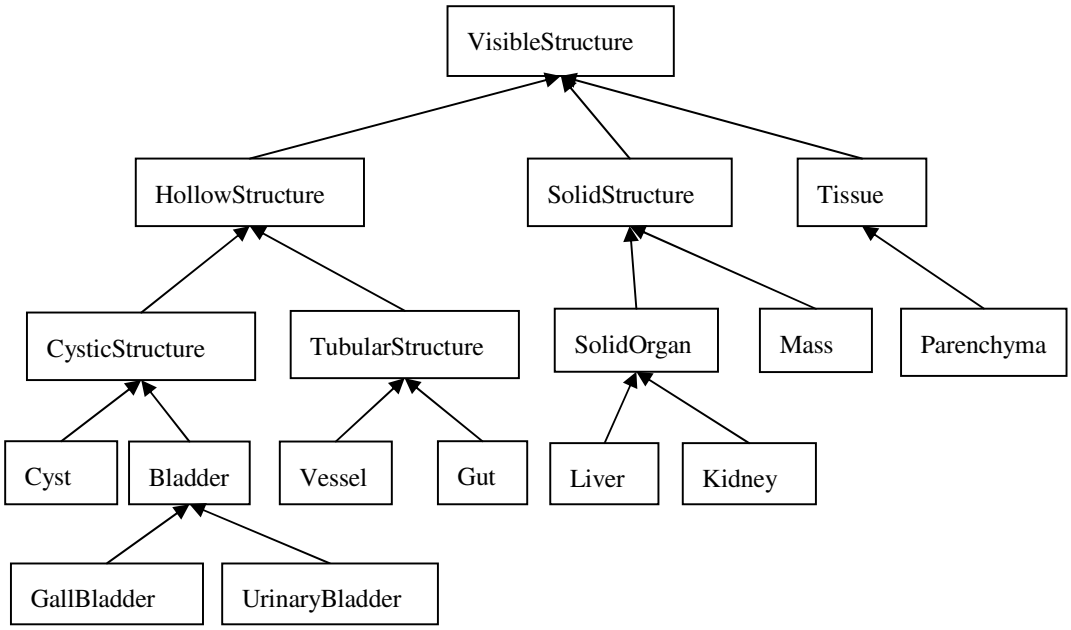

Fig. 8**. An excerpt from TRIES Ontology that was designed using Protégé: VisibleStructure is the parent for all other entities.**

Attributes of entities correspond to information slots in the extracted relations, and they may have strict or loose type checking to allow or disallow the assignment of an attribute value. This means that each attribute is associated with a set of constraints to limit the type of attribute values that it can take. The type of an attribute is one of the constraints, and it may be a simple type such as number, date, enumeration and string. An attribute type may also be some other entity name, or a collection of entity names defined within the ontology. So, the ontology also plays the role of controlled vocabulary for types. For example, if the type of an attribute is the simple type *NUMERIC*, it means that it can only be instantiated with a numeric attribute value. On the other hand, since the *parenchyma* attribute is typed as *Parenchyma* entity in TRIES ontology, the *parenchyma* attribute of *Liver* entity can only be bound to an instance of *Parenchyma* entity with its own instantiated attributes.

When some of the attributes of an entity are associated with values, it is called as an instantiated entity. An instantiated entity may define a non-empty set of relations in the extracted information. Although the instances of some entities can directly appear in the extracted information, the instances of some other entities cannot be directly seen, and their instances must be the attribute values of other instantiated entities. We refer the first group entities as normal entities and the latter as *sub-entities* because their instances can only be attribute values. For example, *Liver* entity is a normal entity, and its instances can directly appear as a set of extracted relations. On the other hand, *Parenchyma* is marked as a sub-entity in TRIES ontology because its instance can be a value of the *parenchyma* attribute of an instantiated *Liver* entity.

TRIES ontology requires modeling a collection of items such as the *cyst* attribute of *Liver* entity. An attribute value can be a collection of instantiated instances of sub-entities. For example, the *cyst* attribute of an instantiated *Liver* entity is a collection of instantiated instances of *Cyst* sub-entities. Table 6 gives some attributes of *Liver* entity together with their types and sources.

Table 6. Some attributes of Liver class with attribute types and sources.

Attribute	Type	Attribute Source
Border	ENUM	VisibleStructure
Height	NUMERIC	SolidStructure
Width	NUMERIC	SolidStructure
Parenchyma	Parenchyma	SolidOrgan
Cyst	Collection	SolidOrgan

Entities in TRIES ontology are also categorized as *instantiable* and *abstract* entities depending on whether their instances can be creatable or not. The instances of *instantiable* entities can be creatable, and they are further categorized as *standalone* entities and *sub-entities*. The instances of standalone entities are directly represented as a set of relations in the TRIES information model. On the other hand, the instances of sub-entities can only be attribute values of other instantiable entities. The usage of sub-entities makes it easy to model the relations in the form of

Entity.entity2.attribute2 = value2

where *Entity* is the name of an instantiated standalone entity with the attribute *entity2*. The value of the attribute *entity2* is an instance of a sub-entity *Entity2*, and that instance contains an attribute named as *attribute2* with a value named as *value2*. The approach that we use for sub-entities is similar to the model defined by Archbold and Evans [18].

The instances of *abstract* entities cannot be created. They help to organize TRIES ontology, and their siblings inherit the attributes that are defined for them. Of course, each abstract entity must have at least one *instantiable* entity as its sibling. In fact, all inner nodes in TRIES ontology are abstract entities and all leaves are instantiable entities.

The strings representing abstract entities often appear in radiology reports, and they cause ambiguity. Let us consider the following example.

Safra kesesi normal boyuttadır. (The size of **gall bladder** is normal.)
Kese içinde taş ya da kitle izlenmedi. (Stone or mass is not observed inside the **bladder**.)

The expression "*bladder*" may be used as a shorthand for either "*gall bladder*" or "*urinary bladder*". This ambiguity must be resolved before the semantic information is extracted from these sentences. TRIES handles this ambiguity problem through abstract entities. At the entity recognition level, these terms are recognized as abstract entities. For example, TRIES

entity recognition module recognizes the Turkish string "*safra kesesi*" as the entity *GallBladder* which is an instantiable entity, and the string "*kese*" as the entity *Bladder* which is an abstract entity. During the relation extraction, an abstract entity is replaced by one of its proper instantiable offspring entities using the context information. In our example, *Bladder* abstract entity is replaced with *GallBladder* instantiable entity because *GallBlader* is an offspring of *Bladder*, and it appears in the previous sentence.

Another kind of ambiguity that is caused by a string representing an abstract entity is that the string can refer to all instantiable siblings of that abstract entity. In order to solve this problem, the abstract entities whose usages in the reports refer to all of its possible instantiable siblings are marked as *propagable* entities. Although an instance of a propagable abstract entity is not created, any value assigned to the attributes of this entity is propagated to siblings. In other words, the instances of its instantiable siblings are created, and all assigned values are copied into these instances. For example, the abstract entity *Kidney* is *propagable*, and all assigned values are copied into the instances of its instantiable siblings *LeftKidney* and *RightKidney*. When TRIES considers the following sentence, the Turkish string "*böbreklerin*" is recognized as the entity *Kidney* by the entity recognition. All extracted attribute values from this sentence are copied into the instances of *LeftKidney* and *RightKidney* entities, and the following relations are extracted.

Böbreklerin büyüklükleri, şekilleri ve yerleri normaldir.
Kidneys are normal in sizes, shapes and locations.

Extracted Relations

```
LeftKidney.size = normal
LeftKidney.shape = normal
LeftKidney.location = normal
RightKidney.size = normal
RightKidney.shape = normal
RightKidney.location = normal
```

5.2 Information Model

One of the main problems for IE systems in medical domain is the proper computational usability of the extracted information. An information model for TRIES is created based on domain expert opinions (Radiologist and Clinician) and guidelines of Turkish Ultrasonography Association. This is a key challenge for the usability of the extracted data for decision making and knowledge discovery. The solution to this problem is achieved by means of domain experts. TRIES ontology is heavily influenced by the target knowledge structures. The complete information model is integrated into the ontology as entities and attributes. So, the ontology also hosts the information model for TRIES. The information extracted from a sentence is populated from the instances of entities of TRIES ontology.

The extracted information is represented as a set of relations. Each relation represents an attribute with its value. Of course, the entity that owns the attribute also appears in the relation. A relation is in the following form:

$Entity.attribute_1. \ldots .attribute_n.simpleattribute = simplevalue$

where $attribute_1. \ldots .attribute_n$ is optional, *Entity* is an instantiable entity, *simpleattribute* is an attribute whose value cannot be an entity instance, and *simplevalue* is its value. If $attribute_1. \ldots .attribute_n$ are present, all of them are attributes whose values can be the entity instances, $attribute_1$ is an attribute of *Entity*, each $attribute_{i+1}$ is an attribute of $attribute_i$, and *simpleattribute* is an attribute of $attribute_n$.

5.3 Entity-Attribute-Value Model (EAV)

Relational data models are based on entities, attributes and relations among those entities. A running application using a relational data model requires exact determination of a fixed data model. However, diversity of medical data models commonly meets to these limitations of relational data model of modern databases that becomes a major obstacle for medical data repositories. In order to overcome this challenge, another model EAV has been proposed

and successfully applied in clinical data as early as late 1970s [60]. Several other attempts are followed by other applications of EAV on clinical data [61-65].

Entity attribute value model borrows concepts from association lists. Attribute values are stored in the form of attribute value pairs, and the object that this attribute belongs to [66]. In EAV, each row is designed to store a single value for a particular attribute, unlike a conventional table that stores one entity per row with a set of attributes belonging to that entity.

Since radiology reports have an arbitrary number of entities having attributes in an arbitrary number, EAV was the model of choice for physical storage of extracted rules. A sample report and data obtained from this particular report are given in Appendixes A and B. EAV provides required flexibility for this diverse data obtained from free texts. So, TRIES uses entity attribute value model to achieve physical storage of extracted rules.

Table 7 Sample data extracted by TRIES. Values for attribute_id and value columns are mapped to SNOMED codes when possible.

row_no	object_instance_id	attribute_id	Value
1	751	report_type	abd_usg
2	751	patient_id	10201023
3	751	RightKidney	1001
4	1001	height	121mm
5	1001	Cyst	1011
6	1011	size	12mm

Table 7 gives a sample data stored in TRIES table. The row #1 bears a data for the report entity with instance_id 751. The value for *report_type* attribute of this particular report is *abd_usg* (i.e. abdominal ultrasonography). On the row #3, instance of a RightKidney is assigned to the report 751. Instance id for this RightKidney object is 1001. On the row #4, this object (1001) has height attribute with value 121mm. The object 1001 also has a *Cyst* attribute with an instance id of 1011. On the row 6, a size attribute was given with a value of 12mm.

In order to make use of this data with different applications for different purposes such as data mining and decision making, extracted data should be in a known standard. In extracted data, TRIES maps its concepts to corresponding SNOMED codes when available. Attribute ids are all mapped to SNOMED codes. Some of extracted values are also mapped to SNOMED codes. But some values such as numeric ones cannot be mapped. So this data will be available to any application that can use SNOMED coded data.

Original EAV model is designed to store entities of the same class. However, TRIES requires storing entities of different classes. In order to achieve this, we created a utility table to manage entity types. (Table 8) This utility table maintains the instance ids and object types of particular which is required to trace entities and entity types.

Table 8 A sample utility table for entities of EAV table given in Table 7

instance_id	object_type_id
…	…
751	Report
…	…
1001	RightKidney
…	…
1011	Cyst

Chapter 6

TRIES: INFORMATION EXTRACTION

6.1 Named Entity Recognition

After all words in a sentence are broken into their morphemes, the sentence is passed to the entity recognizer. The entity recognizer identifies phrases as named entities together with their named entity type. TRIES supports five types of named entities:

Entity – Strings representing ontology entries such as organs and major vessels are recognized as named entities of type *Entity*. In fact, any entity that is not a sub-entity in TRIES ontology is recognized as *Entity*.

Sub-Entity – A string representing an entity that is marked as a sub-entity in TRIES ontology is recognized as a named entity of type *Sub-Entity*.

Attribute – Strings representing the defined attributes in the ontology are recognized as named entities of type *Attribute*.

Value – The possible attribute values are recognized as named entities of type *Value*, and the types of value strings are also determined.

Location – Strings representing topographic locations are recognized as named entities of type *Location*, and they are also used as attribute values.

Initial identifications of words are achieved by means of regular expressions. Before the morphological analyzer, sentences are tokenized into components Pure alphabetical characters ([a-zA-Zğüşıöçğüşiöç]+), numbers ([0-9]+([\.,][0-9]+)*), and non-alphanumeric

characters (e.g. punctuation marks) are classified as a preparatory process to morphological analysis.

The strings that are recognized as named entities are packed as a single unit, and replaced with appropriate named entities. The information about all strings that represent named entities is stored in TRIES ontology, and entity recognition module uses this information together with simple regular expressions to determine the named entity strings. Some of the ambiguity introduced at the morphological analysis level is eliminated by the help of this process.

Attribute values on the other hand, require some further methods. During term analysis, after initial ontological term recognition phase, TRIES identifies and handles some other patterns based on the token classes. For example, a numeric value followed by a token with class of unit (e.g. cm, mm, cc, …) will be sticked together as a term. Or, some attribute values modified by adjectives such as "highly", "much", "very" are combined together to form terms. Similarly range values like 12-14 or multiple numeric values e.g. denoting dimensions of an entity 12x6x3 cm are the examples of these methods.

6.2 Relation Extraction and Rule Templates

The set of rule templates is a classical component of an information extraction system. TRIES uses a set of rule templates that are manually extracted by means of a domain expert. Each rule template is combined with a set of constraints to further eliminate ambiguities. Rule templates in our system correspond to grammar rules. These rule templates are also tightly integrated with TRIES ontology. Ontology entities are used in both expressions and constraints of the rule templates. Each rule template may have additional constraints such as "may this object have this attribute?" or "may this attribute of this object have this value?". A rule template is a regular expression that consists of entities from TRIES ontology. For example, the following is a simple rule template.

```
<VisibleStructure O> <O:Attribute A> +POSS3SG <O:A:Value V> +COP
```

This rule template matches sentences that start with a *VisibleStructure* entity O (i.e. any entity in TRIES ontology since *VisibleStructure* is the root of the ontology tree), and continues with an attribute A that can be an attribute of the entity O and the morpheme *"+POSS3SG"*. The sentence must finish with an attribute value V that can be taken by the attribute A, and the morpheme "*+COP*". There is also an implicit constraint, and it says that O must be an instantible entity. If this rule matches a sentence, the relation "O.A = V" is extracted.

Some words or punctuations usually denote a set of similar grammatical functions. For example, the comma and the Turkish conjunction word "ve" (and) play similar grammatical roles in Turkish sentences. TRIES rules also support macros, which are used for some sort of shorthand, and expand to full instructions. For example, a list of similar items can be expressed as a macro. A rule template using macros is given in Table 9. The first row gives the defined macros, the second row gives the rule template, the third row gives some sample sentences that can match this rule template, and the last row gives the extracted relations from these sentences. In the third row, the sentences are given together with their forms after the entity recognition (the morphologically analyzed Turkish words are not given for simplicity reasons). This rule template can match a sentence, if and only if the matched entity must accept all the attributes in the list item, and all the attributes in the list item must accept the matched value in the sentence.

Table 9. A sample rule and real life sentences matching this rule (LOC: locative suffix, COP: copula). <O:A:Value V> term will be assigned to the list of given list of attributes to an ontology entity derived from VisibleObject, if entity O possesses these attributes.

Macros:

```
CONJ = { ",", "and"}
LIST(X) = X [<CONJ> X]*
```

Rule Template:

```
[ <VisibleStructure O> ]? <O:A:Value V> LIST(<O:Attribute A>) +LOC +COP
```

Sentences:

```
Abdominal aorta normal görünümdedir (Abdominal aorta is in a normal appearance)
[AbdominalAorta] [normal] [appearance] +LOC +COP

Böbrekler normal boyuttadır (Kidneys are normal in dimension)
[Kidney] [normal] [dimension] +LOC +COP

Dalak 10.5x2.5 cm boyuttadır (Spleen is 10.5x2.5 cm in dimension)
[Spleen] [10.5x2.5 cm] [dimension] +LOC +COP

Karaciğer normal şekil ve boyutlardadır (Liver is normal in shape and dimension)
[Liver] [normal] [shape] <CONJ> [dimensions] +LOC +COP
```

Extracted Relations:

```
AbdominalAorta.appearance = normal
LeftKidney.dimension = normal
RightKidney.dimension = normal
Spleen.dimension = 10.5x2.5 cm
Liver.shape = normal
Liver.dimension = normal
```

6.2.1 Reference Resolution

The reference resolution is one of the most important problems in the relation extraction. TRIES uses a context mechanism integrated into the relation extractor in order to solve the reference resolution problem. This context mechanism keeps track of the ontology entities appearing in the sentences in a stack, and tries to estimate the missing (omitted) terms along the sentences using this stack. Whenever the relation extractor faces a missing entity, the context is taken into account in "last in first out" fashion. The extractor tries to estimate the missing entity by referencing ontological properties of entities within the context. In some cases, TRIES ontology is used alone to solve some of the reference resolution problems. The reference resolution is an important utility to further overcome ambiguity.

In some cases, the well known entity attributes can be omitted. For example, although the entity *LeftKidney* and the attribute value *smaller_than_normal* are available, the *size* attribute is missing in the following sentence.

```
Sol böbrek normalden küçüktür. (Left kidney is smaller than
                                                normal.)
[entity:LeftKidney] [value:string:smaller_than_normal] +COP
```

Although this sentence is grammatically and semantically a normal sentence, the extracted attribute value must be assigned to the attribute *size* according to the information model, and the relation "*LeftKidney.size* = *smaller_than_normal*" must be extracted. But this attribute is not present in the sentence, because it is very-well known by a human reader. In order to determine the missing attribute, TRIES ontology is used to find an attribute of *LeftKidney* such that the found attribute accepts *smaller_than_normal* as its value.

In some cases, entities themselves are missing in the sentences. An instantiable entity does not appear in the last two of the following three sentences, and it must be found using the context information.

```
Karaciğer sağ lob vertikal uzunluğu 17 cm'dir. (Liver right lobe vertical length is 17 cm.)
[entity:Liver] [subentity:RightLobe] [attribute:height] +POSS3SG [value:string:17 cm] +COP

Parankim ekosu steatozla uyumlu olarak artmıştır.(Parenchymal echo is increased in
                                                  accordance with steatosis.)
[subentity:Parenchyma] [attribute:echogenity] +POSS3SG  [value:string:steatosis] +LOC
     uyumlu olarak [value:string:increased] +COP

Kitle içermemektedir. (It does not contain a mass.)
 [subentity:Mass] [value:string:not_exist] +COP
```

The instantiable entity *Liver* is mentioned in the first sentence, but it is not mentioned in the next two sentences. Thus, the missing instantiable entity *Liver* in the last two sentences is deduced with the help of the context mechanism. The second sentence contains two attribute values, but it contains only one attribute. This means that one attribute is missing. Since the attribute *echogenity* can get the attribute value *increased* in that sentence, it is associated with that value. In order to find out the missing attribute, a *Parenchyma* attribute that can accept the attribute value *steatosis* is searched among *Parenchyma* attributes using the knowledge available in TRIES ontology. Since the *impression* attribute satisfies this constraint, it is identified as the missing attribute. The third sentence has also a missing attribute. That missing attribute is similarly found, and it is identified as the *appearance* attribute of *Mass* sub-entity. After all reference resolutions are determined, the following relations are extracted from the three sentences given above.

```
Liver.rightlobe.height = 17 cm
Liver.parenchyma. echogenity = increased
Liver.parenchyma.impression = steatosis
Liver.mass.appearance = not_exist
```

The resolution problem will be even worse if we append the following sentence to the sentences above.

```
Parenkim homojendir.  (Its parencyma is homogeneous.)
[subentity:Parenchyma] [value:string:homogenous] +COP
```

In this sentence, there is a sub-entity, namely *Parenchyma*, but there is not any main entity or attribute. The main entity will be found with the help of context information, and the missing attribute will be found with the help of ontology. According to ontology and context information, this sentence must be presented as "*Liver* parenchymal *echogenic structure* is homogenous". In other words, the missing entity is *Liver*, and the missing attribute is *echogenic structure***.**

The relation extractor refers to the ontology as a source of domain knowledge for the resolution of some more issues like disparities of verbal expressions and the information model. In the following two sentences, there are such disparities.

```
Barsak duvarlarında aşikar duvar kalınlığı izlenmedi. (A prominent thickening was
                                                    not observed in the intestinal wall.)
[entity:Intestine] [subentity:wall] [attribute:thickeness] [value:string:not_exist]

Karaciğer parenkim görünümü homojendir.  (Liver parenchymal appearance is homogeneous.)
[entity:Liver] [subentity:Parenchyma] [attribute:appearance] [value:string:homogeneous]
```

In the first sentence, the attribute *thickness* does not accept the attribute value *not_exist*. The acceptable values of the attribute *thickness* are searched in order to determine whether one of them has similar meaning with that value in this context, or not. An acceptable value *normal* for the attribute *thickness* is spotted, and the attribute value *not_exist* is replaced with this new found value. The second sentence has also a similar problem. Here, the attribute value *homogeneous* is not an acceptable value for the attribute *appearance*, and the *Parenchyma* sub-entity does not have the *appearance* attribute. In this case, the attributes of the *Parenchyma* sub-entity are searched to find an attribute that has a similar semantic meaning with the attribute *appearance* in this context, and accepts the attribute value *homogeneous*. Thus, the attribute

echogenic_structure is identified, and it replaces the attribute *appearance* in the second sentence. After all reference resolutions are resolved, the following relations are extracted from the sentences above.

```
Intestine.wall.thickness = normal
Liver.parenchyma. echogenic_structure = homogeneous
```

Sometimes, entities or attributes are expressed as if owned by other entities. In the following sentence, although *diverticulum* attribute belongs to the *wall* sub-entity of urinary bladder, it is referred as an attribute of bladder itself.

```
Mesanede 2 cm çaplı divertikül izlenmiştir. (In urinary bladder, a diverticulum in
                                              2cm diameter was observed.)
[entity:UrinaryBladder] +LOC [value:numeric:2 cm] çaplı [attribute:diverticulum] izlenmiştir
```

It looks like the sentence contains all the required named entities. The relation extractor can determine that there is a missing sub-entity attribute by observing that *UrinaryBladder* cannot have the attribute *diverticulum* but its sub-entity *Wall* can have it. With the help of the ontology, the relation extractor can model the information in this sentence as the following relation:

```
UrinaryBladder.wall.diverticulum = 2 cm
```

Since the extracted data may be required in different formats for different purposes, some attributes may require multiple entries for a single value. For example, *size* is a common attribute frequently used for entities derived from *SolidStructure* either with qualitative values such as "decreased", "slightly increased", etc or quantitative values at one to three dimensions

such as *10.5x2.5 cm*. These multidimensional values represent length, width and depth for the given entity. *SolidStructure* also have separate attributes for *length*, *width* and *depth*. For the consistency of extracted data, this multidimensional *size* must be separated into corresponding dimension attributes. TRIES completes this by an optional post-processing. Although this obviously results in redundancy of data, this is a required step for data consistency.

As a rule based system, semantics are fixed by the rules. The negative meanings in Turkish are expressed using negation morpheme attached to verbs. The rule templates containing the negation morpheme are used to recognize negative information in clinical reports. For example,

```
Karaciğer kitle içermemektedir  (Liver does not contain a mass)
 Liver    mass  içer+NEG+PRESENT+COP
            ("içer" means contain in English)
```

The negation morpheme attached to the verb "içer" indicates the negative information. This negative information is represented with "not_exist" attribute value, and the extracted information from this sentence will be as follows.

```
   Liver.mass = not_exist
```

Chapter 7

TRIES USER INTERFACE

TRIES provides end user tool to provide a query interface for extracted data. Purpose of this tool is to let physicians to execute detailed queries on TRIES extracted data repository, and obtain the list of reports, which are matching to a given set of criteria. Through this list of matching reports, TRIES UI will also allow physicians to access to the details of each report and details of analysis and data extracted from this report.

The query builder tool forms the hearth of the TRIES UI. It acquires each criterion for the query one by one. Criteria consist of an entity attribute, a manually entered value and an operator for the relation of this attribute to the given value such as "greater than" or "contains". This tool allows physician to combine multiple criteria by boolean operators "AND" and "OR". So that, it becomes possible to create more complicated and detailed queries against TRIES data repository.

After the physician is completed the list of criteria for the reports that the physician is looking for, and press to the query button, TRIES UI finds the reports that are matching this list of criteria within its data repository. Then, TRIES UI lists the reports satisfying the given criteria, and let physician to access to the required data quickly.

Lastly, a value is entered to value field, as a parameter to this operator. So that a sample criterion may be formed as:

```
Liver.Parenchyma.echogenicity CONTAINS "artmış"
```

Or another example;

```
Kidney.Stone.dimension BETWEEN 10mm AND 50mm
```

On completion of each criteria, physician must select a proper logical operator to mark the relation of this particular criteria to the previous criterion as one of 'AND' or 'OR' operator. Then the "Add" button must be pressed.

Physician can add as many criteria as required for his/her query. After, all set of criteria are entered, physician must press an execute button to run the query and get the list of the reports satisfying the listed criteria.

7.2 Preparation of SQL statement:

Propagable entities -> propagated to derived entities

```
Kidney.Stone.dimension LESSER_THAN 10mm ➔
( LeftKidney.Stone.dimension LESSER_THAN 10mm OR
RightKidney.Stone.dimension LESSER_THAN 10mm )
```

Since, TRIES stores the extracted rules in the form of SNOMED codes, entities, subobjects and attributes must be mapped from TRIES ontology concepts to SNOMED concept ids. Attribute value fields on the other hand, show great variations. Some values may be transformed into SNOMED concept ids. Controlled vocabularies or simple string fields require little preprocessing. But, numeric values such as diameter or dimension require some manipulations for unification for a proper SQL statement creation.

Some values are given in the form of multiple measurements like height, width and depth e.g. (a dimension of 12x6x3). In that case, value is separated into its parts and the criterion is applied to each of these multiple values separately. Sometimes numeric values are given in the form of intervals like "10-12". For such values, depending on the criteria operator, for less than operator, compared to lower limit of the interval, for the greater than operator, upper limit. Furthermore, numeric values may be expressed verbally, such as "a stone with a milimetric size". In any case, numeric values must be translated to proper metric and volume units to obtain meaningful results.

Then, tries UI adds additional relational criteria which are derived from TRIES ontology and corresponding information model. Finally generated query statements by means of this list of criteria are translated into SQL syntax with regard to EAV model, and executed against TRIES repository.

7.3 Results

Records matching to this query statements are listed in a result table (Fig. 12). Each report is listed with the report identifier, and a clickable link to report itself. Each result will also include the field information that caused to be a match for the given search condition. E.g. when the physician searched for a kidney cyst with a diameter greater than 12mm, results list will include the cyst diameter for the records matching the query (Fig. 9).

Clicking on the report number on the list of results will open a new window having the unprocessed report itself, and the complete analysis of this particular report (Fig. 12). In this page, sentence of the report is listed. Completely processed sentences are colored in green. It is colored in red if the sentence cannot be processed completely. Right after each sentence, TIRES extracted data are also listed. This screen allow physician to further examine the search results in detail, as well as examines the performance of TRIES itself.

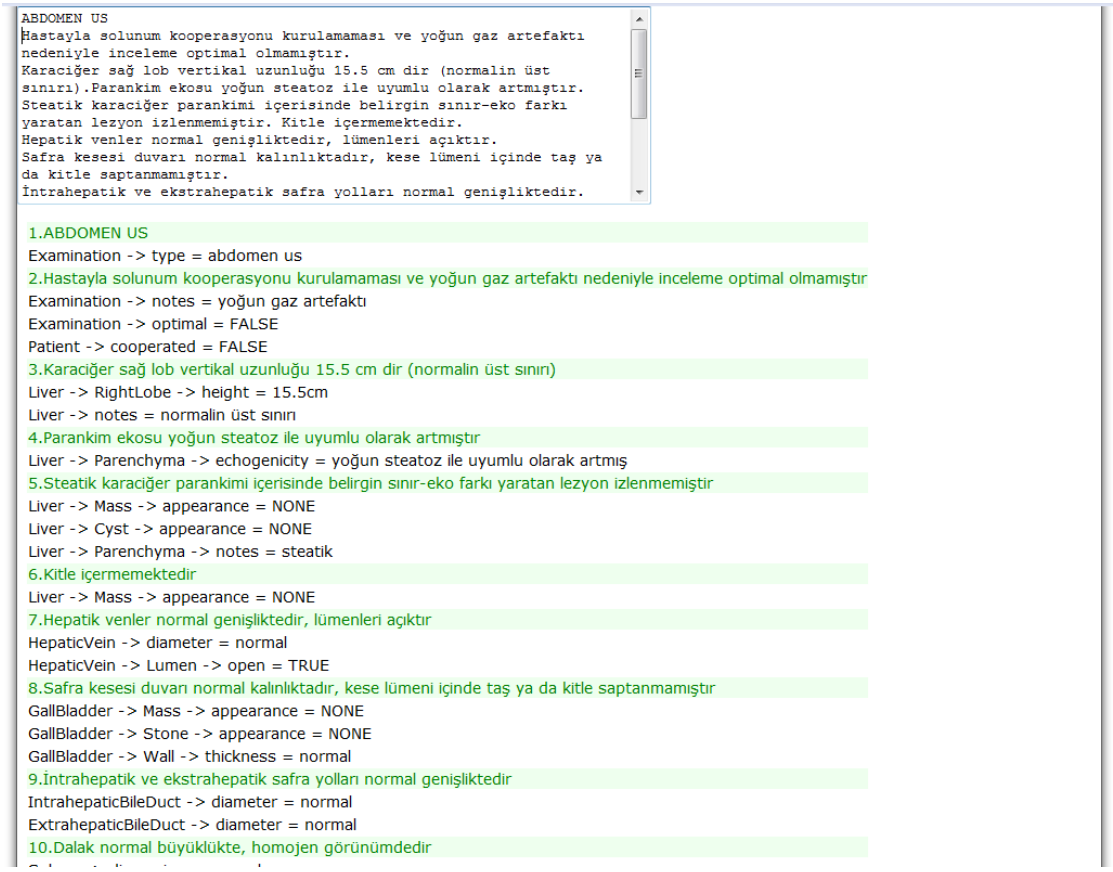

Fig. 12. Details of a report, accessed from search results

Although, this search application allow clinicians to access and utilize TRIES data repository, it does not exploit the complete opportunities provided by TRIES extracted data. These data have a potential to be used many different types of applications such as information summarization (summarize important parts of reports), report visualization or validation during report entry, follow up of lesions over a period of time, alert systems on an abnormal finding, comparison and/or merging of information from different types of reports like Computerized Tomography and Magnetic Resonance Imaging, and many more.

Chapter 8

EVALUATION OF TRIES

For the performance evaluation of TRIES, 100 radiology reports are randomly selected as unseen data. On the average, each report is composed of 14.34 sentences and 105.43 words. The configuration of the system was frozen prior to analyzing the test set. A human domain expert is considered as the gold standard, and the domain expert extracted the relations from these 100 reports. Then, the relations extracted by TRIES are compared against the relations extracted by the domain expert. Table 10 summarizes how the extracted relations are classified. A relation that is extracted by both the domain expert and TRIES is classified as TP (true positive), and a relation that is extracted by TRIES but not extracted by the domain expert is classified as FP (false positive). A relation that is extracted by the domain expert but not TRIES is categorized as FN (false negative).

Table 10. Evaluation table (DE: Domain expert, TP: True positive, FP: False positive, FN: False negative, TN: True negative)

		Extracted by DE	
		Yes	No
Extracted by TRIES	Yes	**TP**	**FP**
	No	**FN**	**TN**

For the evaluation of IE systems, recall and precision values are frequently used [67]. The *recall* of an information extraction system can be defined as the ratio of the number of

relevant findings returned to the total number of findings that are present. The *precision* is the ratio of the number of relevant findings returned to the total numbers of all findings returned. The recall and precision can be formulated in terms of TP, FP, and FN as follows.

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP}$$

Table 11 gives the evaluation results of TRIES. For the evaluation set, the average number of extracted relations for each report is 51.7. For all extracted relations, the overall recall value is 93% and the precision value is 98%. This means that only 2% of the extracted relations are incorrect, and only 7% of the available information is not extracted.

In addition to the general performance of TRIES, its performances in specific cases are also measured and they are given in the rows 2-5 of Table 11. The average number of relations extracted from the sentences containing non-propagable abstract entities is 0.9 per report. In this group of extracted relations, a recall of 92% and precision of 98% have been achieved. Although some sentences contain both an attribute and an attribute value, the appearing value may not be the proper value for the attribute. In those sentences, the attribute value is assigned to another attribute that is found with the help ontology (e.g. parenchymal appearance is homogeneous; appearance mapped to echo structure). For those sentences, the average number of extracted relations is 2.5 per report, the recall and precision values are 91% and 97%, respectively. The average number of extracted relations from the sentences containing missing entities or attributes is 8.1 per report. For this group of sentences, the recall and precision values are 92% and 98%, respectively. Finally, for the group of sentences where attribute values are given by means of a general parent class (e.g. Kidneys are normal in size, instead of declaring left and right kidneys separately), the average number of extracted relations are 21.6 per report, and recall value is 93% and precision value is 98%. These numbers indicate that the performances of our system in special cases are very similar to its overall performance.

Table 11. Average numbers of attributes per report, recall and precision values.

	n per report	*Recall*	*Precision*
Total Extracted Relations	51.7	93%	98%
Relations Extracted from Sentences Containing Non-Propagable Abstract Entities	0.9	92%	98%
Relations Extracted from Sentences Containing Attribute Value Mapped to Another Attribute	2.5	91%	97%
Relations Extracted from Sentences Containing Missing Entity or Attribute	8.1	92%	98%
Relations Extracted from Sentences Containing Propagable Abstract Entities	21.6	93%	98%

SpellCorrector has a prominent contribution to the success of information extraction. Many typing errors that might break the patterns are automatically fixed at the rate of 91% of all misspelled words. The detected errors contain only one error belonging to one of the following cases: a missing letter (25%), an extra letter (39% - frequently doubling of the same letter), a wrong letter (17% - including Turkish letter) and finally two adjacent letters interchanged (9%).

8.1 Errors

For a sentence sequence like following:

```
(1) Safra Kesesi görünümü normaldir. (Gall Bladder appearance
was normal.)
(2) Dalak parankim görünümü normaldi. (Appearance of parencyma
of spleen was normal.)
(3) Taş, kitle, hidronefroz izlenmedi. (Stone, mass,
hydronephrosis were not observed.)
(4) Abdominal büyük damarlar normal genişliktedir. (Greater
vessels of abdomen were normal in width.)
```

TRIES extracts following attributes from the first sentence

```
GallBladder.appearence = normal
```

using the rule

```
<VisibleObject O> <O:Attribute A> <NORMAL> +COP
```

Meanwhile the object *GalBladder* is pushed into the *Context*. Then the next sentence is processed using the rule

```
<VisibleObject O> <O:Subobject S> <S:Attribute A> <NORMAL> +COP
```

and TRIES will extract the following rule:

```
Spleen.Parenchyme.appearence = normal
```

The sentence #3 misses its object erroneously. This may be a copy/paste error, or sometimes sentences may be disorganized. Since, the object was not given, and *Stone*, *Mass* and *Hydronephrosis* are not standalone objects, TRIES attempts to find out the proper object from the Context, which those subobjects belong to:

- Gall Bladder
  - appearence
- Spleen
  - Parencyme
    - appearence

For the first subobject ”Stone”, the last mensioned object was Spleen. Neither the spleen nor any attributes of spleen do not satisfy the **object_has_attribute** constraint. So, the prior object *GallBladder* is peeked. *GallBladder* will satisfy the **object_has_attribute** constraint, since *GallBladder* may have Stones according to ontology. So the rule is extracted, erroneously:

```
GallBladder.Stone.appearence = NONE
```

Similarly, *Mass* subobject is evaluated, and this time the Spleen will satisfy the constraints. And following rule will be extracted:

```
Spleen.Mass.appearence = NONE
```

For the “hydronephrosis”, none of the objects in the context will satisfy the **object_has_attribute** constraint. And this information will not be extracted, and missed.

On the other hand, a human reader will interpret the sentence #3 as a whole and, will notice that this information is about the Kidney objects, even if it is missing.

Any sentence that cannot be matched to any TRIES rule will result in information loss. This may extend to following sentences, if they rely on the object or subobject of this lost sentence.

Similarly, *a template unmatch* because of an unknown pattern or an unresolved term will cause similar information miss within the sentence. Since, this information miss will cause a context

item loss, it will affect the sequential sentences rely on the context and a former object was taken as the owner of the attribute and its value.

The same problem will arise in case of unresolved typing errors. These errors will end up with missing the context entity. At this point, if this erroneous statement is followed by a sentence without entity, the result will be unpredictable. Based on the current state of the context stack, one of the former objects will be taken as the referred entity, having an assignable attribute for the current value.

8.2 Evaluation of TRIES UI

TRIES UI was evaluated in a test environment, which was specifically set up for evaluation purposes. After a brief acknowledgement and a short training, ten radiologists were asked to use the system on their own. Afterwards, they were asked to fill out an evaluation survey consisting of questions rated on a 5-point Likert scale (from 1: strongly disagree to 5: Strongly Agree) and some open questions to let them to express their opinions and comments on the system. The evaluation survey form can be found in Appendix E.

Evaluation results are summarized in Table 12. TRIES UI was found to be easy to learn to use (4.5), easy to use (avg.4.1) and practical to be incorporated into daily practice (avg.4.2). Interaction with the system was clear and comprehensible (avg.4.4).

This application was evaluated as a very useful tool especially for the research projects conducted on radiology reports. Although supporting the daily work (avg. 3.7) and allowing physicians to utilize free text reports in clinical applications (avg. 4.0) did not have so high points, improving clinical researches were found to be the most important advantages (avg. 4.9).

Table 12. Evaluation results for TRIES UI. Each question was rated on a 5-points Likert scale (from 1: strongly disagree to 5: Strongly Agree)

	Average (min-max)
Ease of Use	**4.2(3-5)**
It was easy to use for me.	4.1(3-5)
System is practical to use in daily practice.	4.2(3-5)
Learning to use the system was easy.	4.5(4-5)
Interaction with the system was clear and comprehensible.	4.4(4-5)
System eases my work.	3.7(3-5)
Usability	**4.7(3-5)**
Allows physicians to utilize free text reports in clinical applications.	4.0(3-5)
Improves processing of free texts used in clinical researches.	4.9(4-5)
Tools provided in user interface are sufficient to construct required flexible queries.	4.8(4-5)
Query screen is comprehensible and easy to use.	4.7(4-5)
Organ and organ attributes are sufficient to query target reports in query screen	4.8(4-5)
Query speed and performance are adequate	4.9(4-5)

Radiologists stated that, query screen was comprehensible and easy to use. (avg. 4.7) Tools provided by TRIES user interface were sufficient to construct required flexible queries. (avg. 4.8) For the contents made available in the query screen, organ and organ attributes were sufficient to query the processed reports (avg. 4.8). Finally, system performance and speed were adequate (avg. 4.9)

Open questions yielded several suggestions and requests for improvements, and other uses TRIES system:

- TRIES UI can be used for comparison of different examination of the same patient. This may also be very helpful to merge and compare findings obtained from different observations to help differential diagnosis.

- Another suggestion, follow up of a patient along his/her treatment, and compare examinations of the same type and compare the changes during a given interval. This analysis helps to follow the progress of the clinical picture over a time interval. This is especially stated as an important feature for chronic diseases.

- Alerts for pathological findings may be added. This requires integration into radiological and/or clinical information systems. While the radiologist typing the report or a clinician reading the report, pathological findings may be alerted to notify readers.

- Simultaneous visualization and validation of entered findings may improve report validity and reliability, and prevent erroneous entries and unclear semantic expressions. This feature also requires a tight integration with report writing software.

- System may provide extra information for decision support algorithms. This information may further help for "reasoning for diagnosis" to draw a conclusion from TRIES UI.

- One interesting request was, improvement of TRIES UI so that, provide the ability to build queries by the natural language. This will allow physician to query free text radiology reports by using questions in the natural language.

Chapter 9

CONCLUSION

In this thesis, we presented the first published information extraction system targeting free text radiology reports in Turkish. Unlike traditional information extraction systems, Turkish radiological information extraction system (TRIES) is not intended to extract or identify a particular phrase, term, entity or concept; instead, it processes the complete report to transform into a target information model. So that, the report contents are made available to many other applications such as decision support systems or data mining for further use with clinical research purposes.

9.1 Discussion

Traditionally, information extraction systems simply ignore morphological analysis [10]. Since English morphological structures are not too complex, the morphological analysis is overlooked in most of the IE systems designed for English texts. On the other hand, since Turkish has a rich morphological structure, TRIES brings morphological analysis into notice as a required step for information extraction in agglutinative languages with following contributions that improve the performance of information extraction systems:

- Initially, morphological analysis helps to identify root words. Since suffixes may change the root word, morphological analysis will restore it.
- Lexicon provided by morphological analyzer helps to identify syntactic properties of the root word. This increases the performance of entity recognition.

- TRIES uses results of morphological analysis during term analysis. Since many terms such as "safra kesesi (safra kese+POSS3SG = GallBladder)" includes some morphological structures as the part of the term, these morphemes must be distinguished for a successive entity recognition in Turkish.
- Many morphological structures also contribute to the semantics of phrases and sentences. TRIES rules frequently use morphological elements to increase the flexibility of relation extraction and syntactic abilities.

TRIES also implements a spell-corrector component, which cooperates with morphological analyzer. This helps to avoid 91% of the typo errors, which prominently improves the performance of TRIES by preventing the break of the rules because of unrecognized terms.

TRIES introduces its own ontology to use in radiological information extraction. Although, there are few publications on ontologies in information extraction systems, these systems utilize ontology with a very limited functionality such as semantic tagging at the named entity identification task [29-31, 34], or extracted data as final outcome [30, 32, 33]. On the other hand, ontology of TRIES is tightly integrated with all parts of the system with following contributions:

- TRIES uses its ontology to model the domain knowledge. A particular domain ontology has been developed parallel to expected information content of reports by the help domain experts.
- It uses this domain ontology as the route for transferring domain knowledge from experts into information extraction tasks. This ontology incorporates the knowledge of relevant concepts and their semantic relations into the system. So that, the system learns the entities, attributes and the relationships between them. By the help of this ontology, it knows how to handle individual concept identified.

- Ontology is used during term analysis to recognize entities and entity classes, attributes and value candidates.
- Ontology elements are used within rule templates to improve semantic abilities of the rule extractor. These elements are natively used in rule patterns that will directly match against sentences.
- During rule extraction, resolution of ambiguity problems caused by missing entities, subentities or attributes in sentences are solved. Some of the missing terms are determined by the constraints implied by TRIES ontology. The extracted semantic knowledge is also constrained by the rule templates, the rule constraints and the ontological relations used within the rule templates. The usage of ontology concepts provides flexibility in the design of rule templates.
- Term analyzer and rule extractor works integrated by utilization of a common ontology. This helps TRIES to maintain the system consistency and cooperation between different components of the TRIES system.
- An information model has been developed in parallel to ontology. The structure of TRIES ontology also determines the information model that describes the structure of the extracted semantic information. This information model is roughly equivalent to leaflets of the ontology tree, implementing on full ontologic relationships.

The use of ontology is an important tool for the adaptation of the system to another domain. TRIES ontology is relatively a small ontology designed to model the concepts appearing on abdominal ultrasonography reports. It is not a general purpose ontology, and specifically developed regarding the knowledge requirements of an information extraction system and how the entities are described in reports with a point of angle of domain experts. A future work may concern a statistical formation of a bigger ontology to model all the concepts appearing on different radiology reports.

TRIES proposes a context mechanism that holds the history of referred entities, is also used to figure out the missing terms. This unique approach provides an important tool to convey proper information elements between the sentences.

TRIES describes an information model for structured radiological reports. This information model is in a close relationship with TRIES ontology, which is especially important for the re-usability of the extracted data in different applications. TRIES adopts entity-value-attribute model for physical storage of report data. At the final stage, right before a record is created, TRIES concepts are mapped to corresponding SNOMED concepts to increase the utilization of the repository.

TRIES achieved 93% recall and 98% precision results in the performance evaluations. The scores are very high when compared with other IE systems. The reason for these high scores can be the usage of effective hand-coded rules and ontology in the information extraction. In general, better performance of unsupervised systems is already known, compared to supervised systems. Ontology helps direct transfer and utilization of information into system directly.

TRIES also introduces an experimental user interface (UI) to allow physicians to directly access and query this data repository. Application allows using several comparison operators such as "contains", "greater or equal" or "between … and …" to query the report data. Additionally, it is possible to combine multiple conditionals to create complex queries. Evaluation results of this UI showed that, the tool is very useful for scientific research on free text reports (avg. 4.9 out of 5) and it is sufficient to construct required flexible queries (avg. 4.8 out of 5).

9.2 Future Work

Although, TRIES UI successfully implements a search application to allow clinicians to access and utilize TRIES data repository, it does not fully exploit the complete opportunities

provided by TRIES extracted data. Information that is extracted by TRIES can be utilized by various other applications.

TRIES extracted data can be used for text summarization purposes. Some medical documents (e.g. discharge summaries) require a shorter version covering the most important parts of these reports. TRIES can summarize this information by distinguishing normal and abnormal findings. So, instead of disseminating the full report, a minimized version only covering key points within the report may be generated.

Another use of TRIES repository is the report visualization. Verbal expression of reports may be schematized to generate figures representing the findings in a particular report. This data may be used with instant validation of reports during the report entry to detect semantic ambiguities to improve report reliability. It may be used in follow up of lesions over a period of time, alert systems on an abnormal finding, comparison and/or merging of information from different types of reports like Computerized Tomography and Magnetic Resonance Imaging, and many more. But further applications of TRIES should regard that TRIES extract explicitly expressed information in reports and not the implied ones.

BIBLIOGRAPHY

1. Corrigan JM, Donaldson MS, Kohn LT. Crossing the quality chasm: A new health system for the 21st century. Washington, DC: National Academies Press; 2001.

2. Hirschman L, Grishman R, Sager N. From text to structured information: automatic processing of medical reports. In: Proceedings of the June 7-10, 1976, national computer conference and exposition. New York, New York: ACM; 1976. p. 267-275.

3. Hirschman L, Grishman R. Fact Retrieval from Natural Language Medical Records. In: Proceedings of the Second World Conference on Medical lnformatics. Amsterdam, Holland: IFIP World Conference Series in Medical Informatics; 1977. p. 247-251.

4. Grishman R. Syntax Analysis. In: Computational Linguistics: An Introduction. New York, NY, USA: Cambridge University Press; 1986. p. 10-89.

5. Roark B, Sproat R. The Formal Characterization of Morphological Operations. In: Computational Approaches to Morphology and Syntax. Oxford University Press, USA; 2007. p. 23-61.

6. Kaplan RM, Kay M. Phonological rules and finite-state transducers. In: Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting. 1981. p. 27–30.

7. Kaplan RM, Kay M. Regular models of phonological rule systems. Computational linguistics. 1994;20(3):331–378.

8. Karttunen L, Chanod JP, Grefenstette G, Schille A. Regular expressions for language engineering. Natural Language Engineering. 1996;2(04):305–328.

9. Grishman R. Semantic Analysis. In: Computational Linguistics: An Introduction. New York, NY, USA: Cambridge University Press; 1986. p. 90-139.

10. Friedman C, Johnson SB. Natural Language and Text Processing in Biomedicine. In: Biomedical informatics: computer applications in health care and biomedicine. Springer; 2006. p. 312-343.

11. Appelt DE. Introduction to information extraction. AI Communications. 1999;12(3):161-172.

12. Grishman R, Sundheim B. Message understanding conference-6: A brief history. In: Proceedings of the 16th conference on Computational linguistics-Volume 1. 1996. p. 471.

13. Humphreys K, Gaizauskas R, Azzam S, Huyck C, Mitchell B, Cunningham H, et al. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In: Proceedings of the Seventh Message Understanding Conferences (MUC-7). Citeseer; 1998.

14. Evans DA, Brownlow ND, Hersh WR, Campbell EM. Automating concept identification in the electronic medical record: an experiment in extracting dosage information. In: Proceedings of the American Medical Informatics Association Annual Fall Symposium. 1996. p. 388-392.

15. Guarino N. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In: International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. London, UK: Springer-Verlag; 1997. p. 139-170.

16. Sager N, Friedman C, Lyman MS. Medical language processing: computer management of narrative data. 1st ed. Boston, MA: Addison-Wesley Publishing Co.; 1987.

17. Rassinoux AM, Wagner JC, Lovis C, Baud RH, Rector A, Scherrer JR. Analysis of medical texts based on a sound medical model. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. 1995. p. 27.

18. Archbold AA, Evans DA. On the Topical Structure of Medical Charts. In: Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care. Washington, DC: IEEE Press; 1989. p. 543-547.

19. Turmo J, Ageno A, Català N. Adaptive information extraction. ACM Computing Surveys (CSUR). 2006;38(2):4.

20. Uschold M, Gruninger M. Ontologies: Principles, methods and applications. Knowledge Engineering Review. 1996;11(2):93-136.

21. Gruber TR. Toward principles for the design of ontologies used for knowledge

sharing. International Journal of Human Computer Studies. 1995;43(5):907-928.

22. Gruber TR. A translation approach to portable ontology specifications. Knowledge acquisition. 1993;5(2):199-220.

23. Rector AL, Rogers JE, Zanstra PE, van der Haring E. OpenGALEN: Open Source Medical Terminology and Tools. AMIA Annu Symp Proc. 2003;2003:982-982.

24. Jr JLM, Rubin DL, Brinkley JF. FMA-RadLex: An Application Ontology of Radiological Anatomy derived from the Foundational Model of Anatomy Reference Ontology. AMIA Annu Symp Proc. 2008;2008:465-469.

25. Rubin D. Creating and Curating a Terminology for Radiology: Ontology Modeling and Analysis. Journal of Digital Imaging. 2008;21(4):355-362.

26. Marwede D, Fielding M, Kahn T. RadiO: A Prototype Application Ontology for Radiology Reporting Tasks. In: AMIA Annual Symposium Proceedings. 2007. p. 513.

27. Witte R, Kappler T, Baker CJ. Ontology design for biomedical text mining. Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences. 2006;

28. Bontcheva K, Cunningham H, Kiryakov A, Tablan V. Semantic Annotation and Human Language Technology. In: Semantic Web Technologies: Trends and Research in Ontology Based Systems. p. 29-50.

29. Müller H, Kenny EE, Sternberg PW. Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. PLoS Biol. 2004;2(11):e309.

30. Embley DW, Campbell DM, Smith RD, Liddle SW. Ontology-based extraction and structuring of information from data-rich unstructured documents. In: Proceedings of the seventh international conference on Information and knowledge management. New York, NY, USA: ACM Press; 1998. p. 52-59.

31. Todirascu A, Romary L, Bekhouche D. Vulcain - An Ontology-Based Information Extraction System. In: Natural Language Processing and Information Systems. Stockholm, Sweden: 2002. p. 64-75.

32. Buitelaar P, Cimiano P, Racioppa S, Siegel M. Ontology-based information extraction with soba. In: Proceedings of the International Conference on Language Resources and Evaluation. 2006. p. 2321-2324.

33. Wood M, Lydon S, Tablan V, Maynard D, Cunningham H. Populating a Database from Parallel Texts Using Ontology-Based Information Extraction. In: Natural Language Processing and Information Systems. 2004. p. 357-365.

34. McDowell L, Cafarella M. Ontology-driven information extraction with ontosyphon. The Semantic Web-ISWC 2006. 2006;:428–444.

35. Sager N, Lyman M, Nhan NT, Tick LJ. Medical language processing: applications to patient data representation and automatic encoding. Methods of information in medicine. 1995;34(1-2):140-146.

36. Haug PJ, Ranum DL, Frederick PR. Computerized extraction of coded findings from free-text radiologic reports. Work in progress. Radiology. 1990 Feb;174(2):543-548.

37. Haug P, Koehler S, Lau LM, Wang P, Rocha R, Huff S. A natural language understanding system combining syntactic and semantic techniques. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. 1994. p. 247.

38. Haug PJ, Koehler S, Lau LM, Wang P, Rocha R, Huff SM. Experience with a mixed semantic/syntactic parser. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. 1995. p. 284.

39. Worsley DF, Alavi A, Aronchick JM, Chen JT, Greenspan RH, Ravin CE. Chest radiographic findings in patients with acute pulmonary embolism: observations from the PIOPED Study. Radiology. 1993;189(1):133.

40. Friedman C. Towards a comprehensive medical language processing system: methods and issues. In: AMIA ANNUAL FALL SYMPOSIUM. 1997. p. 595-599.

41. Friedman C, Alderson P, Austin J, Cimino J, Johnson S. A general natural-language text processor for clinical radiology. J Am Med Inform Assoc. 1994;1(2):161-174.

42. Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking Clinical Data from Narrative Reports: A Study of Natural Language Processing. Ann Intern Med. 1995 May 1;122(9):681-688.

43. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics. 2001;17(Suppl 1):S74-82.

44. Zweigenbaum P. MENELAS: an access system for medical records using natural

language. Computer Methods and Programs in Biomedicine. 1994;45(1-2):117-120.

45. Zweigenbaum P, Bachimont B, Bouaud J, Charlet J, Boisvieux JF. A multi-lingual architecture for building a normalised conceptual representation from medical language. In: Proceedings of the Annual Symposium on Computer Applications in Medical Care. 1995. p. 357-361.

46. Hahn U, Romacker M, Schulz S. MEDSYNDIKATE a natural language system for the extraction of medical information from findings reports. International Journal of Medical Informatics. 2002 Dec 4;67(1-3):63-74.

47. Mykowiecka A, Marciniak M, Kupsc A. Rule-based information extraction from patients' clinical data. Journal of Biomedical Informatics. 2009 Oct;42(5):923-936.

48. Berrut C. Indexing medical reports: The rime approach. Information Processing & Management. 1990;26(1):93-109.

49. Bekhouche D, Pollet Y, Grilheres B, Denis X. Architecture of a medical information extraction system. In: Natural Language Processing and Information Systems. Berlin / Heidelberg: Springer; 2004. p. 522-531.

50. Johnson DB, Taira RK, Cardenas AF, Aberle DR. Extracting information from free text radiology reports. International Journal on Digital Libraries. 1997 Dec 25;1(3):297-308.

51. Amaral MBD, Satomura Y. Structuring medical information into a language-independent database. Informatics for Health and Social Care. 1994;19(3):269.

52. Maedche A, Neumann G, Staab S, Saarbruecken G. Bootstrapping an ontology-based information extraction system. Studies In Fuzziness And Soft Computing. 2003;111:345-362.

53. Erozel G, Cicekli NK, Cicekli I. Natural language querying for video databases. Information Sciences. 2008 Jun 15;178(12):2534-2552.

54. Oflazer K. Two-level description of Turkish morphology. Literary and Linguistic Computing. 1994;9(2):137-148.

55. Antworth EL. PC-KIMMO: a two-level processor for morphological analysis. Occasional Publications in Academic Computing. 1990;16.

56. Lewis G. Turkish Grammar. 2nd ed. Oxford University Press, USA; 2001.

57. Göksel A, Kerslake C. Turkish: A Comprehensive Grammar. Bilingual. Routledge; 2005.

58. Temizsoy M, Cicekli I. An Ontology-Based Approach to Parsing Turkish Sentences. In: Machine Translation and the Information Soup. Springer; 1998. p. 124-135.

59. Noy NF, Fergerson RW, Musen MA. The knowledge model of Protege-2000: Combining interoperability and flexibility. Lecture Notes in Computer Science. 2000;:17–32.

60. Stead W, Hammond W, Straube M. A Chartless Record—Is It Adequate? Proc Annu Symp Comput Appl Med Care. 1982 Nov 2;:89-94.

61. Chen RS, Nadkarni P, Marenco L, Levin F, Erdos J, Miller PL. Exploring Performance Issues for a Clinical Database Organized Using an Entity-Attribute-Value Representation. J Am Med Inform Assoc. 2000;7(5):475-487.

62. Ganslandt T, Mueller M, Krieglstein C, Senninger N, Prokosch H. A Flexible Repository for Clinical Trial Data Based on an Entity-Attribute-Value Model. Proc AMIA Symp. 1999;:1064.

63. Nadkarni PM, Brandt C, Frawley S, Sayward FG, Einbinder R, Zelterman D, et al. Managing Attribute—Value Clinical Trials Data Using the ACT/DB Client—Server Database System. J Am Med Inform Assoc. 1998;5(2):139-151.

64. Nadkarni PM, Brandt C. Data Extraction and Ad Hoc Query of an Entity—Attribute— Value Database. J Am Med Inform Assoc. 1998;5(6):511-527.

65. Nadkarni PM. QAV: querying entity-attribute-value metadata in a biomedical database. Computer Methods and Programs in Biomedicine. 1997;53(2):93-103.

66. Dinu V, Nadkarni P. Guidelines for the effective use of entity-attribute-value modeling for biomedical databases. International journal of medical informatics. 2007;76(11-12):769-79.

67. Hripcsak G, Kuperman GJ, Friedman C, Heitjan DF. A Reliability Study for Evaluating Information Extraction from Radiology Reports. J Am Med Inform Assoc. 1999 Apr;6(2):143-150.

APPENDIX A: A SAMPLE RADIOLOGY REPORT

ABDOMEN US

Karaciğer konum ve boyuttadır. Sağ lob vertikal uzunluğu 14 cm dir. Parankim ekosu homojendir.

Kitle saptanmamıştır.

Portal ven ve hepatik venler tabii görünümdedir.

Safra kesesi duvarı normal kalınlıktadır. Kese lümeni içerisinde taş veya kitle saptanmamıştır. Intra ve ekstrahepatik safra yolları normal genişliktedir.

Dalak boyutları 10.5x3.5 cm düzeyindedir. Parankimi homojendir.

Pankreasta patoloji saptanmamıştır.

Böbreklerin yeri, şekli, büyüklükleri ve parankim eko yapıları normaldir. Taş, kitle, hidronefroz bulgusu saptanmamıştır.

Abdominal aorta normal görünümdedir.

Vena kava inferior tabiidir.

Batında kitle, assit, paraaortik LAP saptanmamıştır.

APPENDIX B: DATA EXTRACTED FROM THE SAMPLE RADIOLOGY REPORT

```
<ABDOMEN-US>
 <Liver>
  <size no="1">normal</size>
  <location no="1">normal</size>
  <height no="2">14 cm</ height>
  <Parenchyma>
   <echo_structure no="3">homojen</echo_structure>
  </Parenchyma>
  <Mass>
   <appearance no="4">none</appearance>
  </Mass>
 </Liver>
 <PortalVein>
  <appearance no="5">tabii</appearance>
 <PortalVein>
 <HepaticVeins>
  <appearance no="5">tabii</appearance>
 <HepaticVeins>
 <GallBladder>
```

```
 <Wall>
  <thickness no="6">normal</thickness>
 </Wall>
 <Lumen>
  <Mass>
   <appearance no="7">none</appearance>
  </Mass>
  <Stone>
   <appearance no="7">none</appearance>
  </Stone>
 </Lumen>
</GallBladder>
<IntraHepaticBileDucts>
 <diameter no="8">normal</diameter>
</IntraHepaticBileDucts>
<ExtraHepaticBileDucts>
 <diameter no="8">normal</diameter>
</ExtraHepaticBileDucts>
<Spleen>
 <size no="9">10.5x3.5 cm</size>
 <Parenchyma>
  <echo_structure no="10">homojen</echo_structure>
 </Parenchyma>
```

```
</Spleen>
<Pancreas>
 <appearance no="11">normal</appearance>
</Pancreas>
<LeftKidney>
 <location no="12">normal</location>
 <shape no="12"> normal</shape>
 <size no="12"> normal</size >
 <Parenchyma>
  <echo_structure no="12">normal</echo_structure>
 </Parenchyma>
 <Stone>
  <appearance no="13">none</appearance>
 </Stone>
 <Mass>
  <appearance no="13">none</appearance>
 </Mass>
 <Hydronephrosis>
  <appearance no="13">none</appearance>
 </Hydronephrosis >
</LeftKidney>
<RightKidney>
 <location no="12">normal</location>
```

```
 <shape no="12"> normal</shape>
 <size no="12"> normal</size >
 <Parenchyma>
  <echo_structure no="12"> normal</echo_structure>
 </Parenchyma>
 <Stone>
  <appearance no="13">none</appearance>
 </Stone>
 <Mass>
  <appearance no="13">none</appearance>
 </Mass>
 <Hydronephrosis>
  <appearance no="13">none</appearance>
 </Hydronephrosis >
</RightKidney>
<AbdominalAorta>
  <appearance no="14">normal</appearance>
</AbdominalAorta>
<InferiorVenaCava>
  <appearance no="15">tabii</appearance>
</InferiorVenaCava>
<Mass>
 <appearance no="16">none</appearance>
```

```
 </Mass>
 <Ascite>
  <appearance no="16"> none </appearance>
 </Ascite>
 <LymphAdenoPathy>
  <paraaortic>
   <appearance no="16"> none </appearance>
  </paraaortic>
 </LymphAdenoPathy>
</ABDOMEN-US>
```

APPENDIX C: TRIES MORPHOLOGY ANALIZER RULES

```
; DEFINITIONS of letter sets

CONS = b c ç d f g ğ h j k l m n p q r s ş t v w x y z D Y Z N K T P J Q L R ^

SCONS = b c ç d f g ğ h j k l m n p q r s ş t v w x y z

;front vowels

Vi = i e

;back vowels

Vbk = ı a

;round vowels

Vrd = ü ö

;back + round vowels

Vbkrd = u o

Ve = e i ö ü

Va = a ı o u

VOWEL = a ı o u e i ö ü A H E

SVOWEL = ı i o ö u ü a e

; *** RULES ****

;dropping y(Y) buffer (1-2)

RULE Y:y    VOWEL:SVOWEL +:0 _

RULE Y:0    CONS:SCONS  +:0  _

;dropping s(Z) buffer (3-4)
```

```
RULE Z:s    VOWEL:SVOWEL +:0 _

RULE Z:0    CONS:SCONS +:0 _

; n(N) buffer deletion (5)

RULE N:0    :SCONS +:0 _

RULE N:n    :SVOWEL +:0 _

; Vowel harmony (6-7)

RULE A:a    :Va :SCONS * _

RULE A:e    [:Ve | E: | L: ] :SCONS * _

; H: {u, ü, ı, i} based on vowel harmony (8-11)

RULE H:u    [ Vbkrd:0 | :Vbkrd ]  :SCONS * _

RULE H:ü    [ Vrd:0 | :Vrd | L: ] :SCONS * _

RULE H:ı    [ Vbk:0 | :Vbk  ]    :SCONS * _

RULE H:i    [ Vi:0 | :Vi  | E: | L: ] :SCONS * _

; D lexical representation (12-13)

RULE D:t [ :h | :ç | :s | :ş | :k | :p | :t | :f ] +:0  _

RULE D:d [ :b | :c | :d | :g | :ğ | :j | :l | :m | :n | :r | :v | :y | :z | :SVOWEL] +:0 _

; final (de)voicing (14-15)

RULE {K, T, P, Q, J}:{ğ, d, b, g, c}    _ +:0 :SVOWEL

RULE {K, T, P, Q, J}:{k, t, p, k, ç}    _ [ # | :SCONS ]

!; drop the vowel if suffix begins with a vowel (16)

RULE SVOWEL:0  _ +:0 H y o r

; passive voice forms (17-18)

RULE R:n   [ :l | :SVOWEL ] +:0 H: _
```

```
RULE R:l   :SCONS +:0 H: _

; doubling: hatta (19-24)

RULE ^:0      _ #

RULE ^:0      _ +:0 @:SCONS

RULE ^:r      r _ @:SVOWEL

RULE ^:t      t _ @:SVOWEL

RULE ^:k      k _ @:SVOWEL

RULE ^:f      f _ @:SVOWEL
```

APPENDIX D: TRIES EXTRACTION RULE SAMPLES

Macros

```
+COP : ( +dir, +dır, +dür, +dur )

+LOC : ( +de, +da )

+NEG : ( +me, +ma )

CONJ = { "," "ve", "ya da", "ile", "olup", ...}

LIST(x) => x [ <CONJ> x ]*

OBJorSUB() => [<O:Attribute A> | <O:SubObject> <S:Attribute> ]

OBJorSUBe(x) => [<O:Attribute A> x | <O:SubObject> <S:Attribute> x ]

EXIST = {"izle", "gör", "sapta", "mevcut", "dikkati çekmiş",...}

NOT-EXIST = { "izle", "gör", "sapta", ... } +NEG

IN = { "içinde", "dahilinde", +LOC }
```

Sample Rules

- ➔ [<VisibleStructure O>]? <O:SubObject S> <S:Attribute> <Value> +PERS3SG +COP
- ➔ <VisibleStructure O> <Value> <O:Attribute> +LOC +COP <CONJ> <O:SubObject S> +POSS3PL <S:Attribute> +PERS3SG +COP
- ➔ <VisibleStructure O> <NOT-EXIST>
- ➔ [<VisibleStructure O>]? <Value> LIST(<O:Attribute>) +LOC +COP

➔ [<VisibleStructure O>]? <O:SubObject S> +POSS3SG <Value> <S:Attribute> +LOC +COP

➔ <VisibleStructure O> +LOC <PATHOLOGY> <NOT-EXIST>

➔ [<VisibleStructure O>]? LIST(<O:SubObject S>) bulgusu <NOT-EXIST>

➔ LOCATION LIST(<O:SubObject>) <NOT-EXIST>

➔ LIST(LOCATION) alan +PL? +LOC <O:SubObject S> <NOT-EXIST>

➔ <VisibleStructure O> <O:SubObject S> +POSS3SG <Value> <S:Attribute> +LOC +COP <CONJ> <VisibleStructure O> <O:SubObject S> +POSS3SG +LOC LIST(<O:SubObject S>) <NOT-EXIST>

➔ <VisibleStructure O> <Value> <O:Attribute> +LOC <CONJ> <O:SubObject S> +POSS3SG <Value> <S:Attribute> +LOC +COP

➔ LIST(<VisibleStructure O>) <Value> <O:Attribute> +LOC +COP

➔ [<VisibleStructure O>]? LIST(<Value> <O:Attribute> +LOC) +COP

➔ <VisibleStructure O> <Value> <O:Attribute> +LOC <CONJ> <O:SubObject S> <Value> +PERS3SG? +COP

➔ <VisibleStructure O> +LOC LIST(LOCATION? <O:SubObject S>) <NOT-EXIST>

➔ <VisibleStructure O> LIST(OBJorSUBe(+POSS3SG) <Value>) +COP

➔ [<VisibleStructure O>]? <O:SubObject S> +POSS3SG? <Value> <EXIST>

➔ <VisibleStructure O> <O:SubObject S> <Value> <S:Attribute> +LOC +COP

➔ <VisibleStructure O> LIST(OBJorSUBe(+POSS3SG?)) <VisibleStructure O> <O:SubObject S> <S:Attribute> +POSS3SG <Value> +PERS3SG? +COP

➔ [<VisibleStructure O>]? <O:Attribute> +POSS3SG <Value> +PERS3SG +COP

➔ [<VisibleStructure O>]? LIST(OBJorSUBe(+POSS3SG?)) <Value> +PERS3SG? +COP

➔ <VisibleStructure O> +GEN LIST(OBJorSUBe(+POSS3SG?)) <Value> +PERS3SG? +COP

➔ <VisibleStructure O> <Value> <O:Attribute> +LOC <CONJ> <Value> <EXIST>

➔ <VisibleStructure O> <O:Attribute> +GEN <Value> +PERS3SG? +COP

➔ <VisibleStructure O> <O:Attribute> +POSS3SG <Value> +PERS3SG <CONJ> <O:SubObject> <Value> +PERS3SG? +COP

➔ <VisibleStructure O> <O:Attribute> +POSS3SG <Value> <CONJ> <O:SubObject> <Value> <O:Attribute> +LOC +COP

➔ <VisibleStructure O> <O:Attribute> +POSS3SG <Value> <CONJ> <O:SubObject S> <Value> <O:Attribute> +LOC +COP

➔ <VisibleStructure O> LIST([<O:SubObject> | <O:Attribute>]) <NOT-EXIST>

➔ <VisibleStructure O> <O:Attribute> +POSS3SG <Value> +PERS3SG? +COP

➔ <VisibleStructures O> +GEN LIST(OBJorSUBe(+POSS3SG?)) <CONJ> <VisibleStructure O> <O:SubObject S> <S:Attribute> +POSS3SG? <Value> +PERS3SG? +COP

APPENDIX E: TRIES UI EVALUATION SURVEY

Dear participant,

This survey was developed to collect your opinions the system developed. Survey covers the key elements about the use of the system. Please, mark your opinion by an (X) sign to the corresponding column as 1: strongly disagree, 2: disagree, 3: undecided, 4: agree or 5:strongly agree. Your opinions in this matter are of great importance for our research. Your survey responses will be used only in the context of this research, individual responses will not be shared with third parties strictly.

Thank you for your contributions to our research.

1- strongly disagree
2- disagree
3- undecided
4- agree
5- strongly agree

Ease of Use	**1**	**2**	**3**	**4**	**5**
It was easy to use for me.					
System is practical to use in daily practice.					
Learning to use the system was easy.					
Interaction with the system was clear and comprehensible.					
System eases my work.					

Usability	**1**	**2**	**3**	**4**	**5**
Allows physicians to utilize free text reports in clinical applications.					
Improves processing of free texts used in clinical researches.					
Tools provided in user interface are sufficient to construct required flexible queries.					
Query screen is comprehensible and easy to use.					
Organ and organ attributes are sufficient to query target reports in query screen.					
Query speed and performance are adequate.					

We can also use the system for:

_____

_____

_____

_____

_____

_____

_____

_____

These features were unsatisfactory:

_____

_____

_____

_____

_____

_____

_____

_____

These features can be added to the system:

_____

_____

_____

_____

_____

_____

_____

_____

THANK YOU ALL FOR YOUR TIME AND YOUR VALUABLE COMMENTS HERE …

INDEX

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Soysal, Ergin

Nationality: Turkish (TC)

Date and Place of Birth: 10 July 1966, Antalya

Marital Status: Married

Phone: +90 312 595 69 65

email: esoysal@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
PhD	METU, Institute of Informatics, Dept. of Health Informatics	2010
Urology Specialist	Ankara University, Dept. of Urology	1999
MD	Hacettepe University, Faculty of Medicine	1993

WORK EXPERIENCE

Year	Place	Enrollment
2006-Present	Ankara University	Dept of Medical Education and Informatics, Research Assistant
2002-2006	Pleksus Bilişim Teknolojileri	R&D Director
2001-2002	NuSphere Corp.	Software Architect
1999-2001	Pleksus Bilişim Teknolojileri	Lead Developer
1994-1999	Ankara University	Dept of Urology, Research Assistant

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. Soysal E., Özdiler E. "Epididimal Maturasyonun Seminal Parametrelerle İlişkisi" Yeni Üroloji Dergisi, 4(2), 83-92 (2008).

2. Gokmen Soysal H, Soysal E, Markoc F, Ardic F. "Basal Cell Carcinoma of the Eyelids and Periorbital Region in a Turkish Population" Ophthalmic Plastic & Reconstructive Surgery, 24(3), 201-206 (2008).

3. Soysal E. "Ulusal Sağlık Bilgi Ağları". In: Editör Musoğlu E. 2000 yılların Türkiye'sinde Sağlıkta Bilgi Stratejileri. Tübitak, (2001).

4. Türkölmez K. Göğüş O., Bozlu M., Soysal E., Seçkiner İ., Müftüoğlu Y.Z. "Üreteropelvik darlıklı olgulardaki dismembered piyeloplasti operasyonlarında eksternal

ve internal diversiyon yöntemlerinin karşılaştırılması" Üroloji Bülteni, 10(2), 117-120 (1999).

5. Türkölmez K, Bozkurt ÖF, Yağcı C, Şafak M, Baltacı S, Soysal E. "Renal hücreli kanserlerin tedavisinde radikal nefrektomiye ilave olarak adrenalektomi rutin yapılmalı mıdır?" Üroloji Bülteni, 10(1), 25-29 (1999).