Semantic Similarity Measures in MeSH Ontology and their application to Information Retrieval on Medline

Angelos Hliaoutakis

October 3, 2005

Contents

List of Tables

List of Figures

Abstract

Semantic Similarity relates to computing the similarity between concepts, having the same meaning or related information, which are not lexicographically similar. This is an important problem in Natural Language Processing and Information Retrieval Research and has received considerable attention in the literature. Several algorithmic approaches for computing semantic similarity have been proposed. We investigate approaches for computing semantic similarity by mapping terms or concepts to an ontology and by examining their relationships in that ontology. Comparing concepts that belong to different ontologies is far more difficult problem. Some of the most popular semantic similarity approaches are implemented and evaluated based on WordNet [1] as the underlying reference ontology. We also propose a method for comparing terms in different ontologies. In this work we examined similarity between terms basically from MeSH [2] (medical) and WordNet ontologies.

Building upon the idea of semantic similarity we also propose an information retrieval methodology capable of detecting similarities between documents containing semantically similar but not necessarily identical terms. Our proposed Information Retrieval model has been evaluated for retrieval of documents in Medline [3] database. The experimental results demonstrated that our proposed model (although slower) achieves significant performance improvements compared to the state-of-the-art approach based on the Vector Space Model.

[1] http://wordnet.princeton.edu
[2] http://www.nlm.nih.gov/mesh/meshhome.html
[3] http://medline.cos.com/

Acknowledgements

This research was supported by the Intelligent Systems Laboratory of the Techical University of Crete. I would like to thank Dr. Petrakis for the advice, encouragement and support he provided to me in supervising this thesis. I would also like to thank Dr. Millios, for his critical analysis and recommendations. Special thanks go to Varelas Giannis, Epimenidis Voutsakis, Paraskevi Raftopoulou and Nikos Hurdakis. Finally, I feel grateful to Dr.Qiufen Qi, who created the dataset for the evaluation of medical terms, and for carrying out all the evaluations on Medline.

Chapter 1

Introduction

Information retrieval (IR) in the subject of intensive research efforts during the last twenty years [3]. The purpose of information retrieval is to assist users in locating information they are looking for. Information retrieval is currently being applied in a variety of application domains from database systems to web information search engines. The main idea is to locate documents that contain terms that users specify in queries. Retrieval, by classical information retrieval models (eg. Vector Space, Probabilistic, Boolean), is based on plain lexicographic term matching between terms (eg. a query and a document term are considered similar if they are lexicographically the same). However, plain lexicographic analysis and matching is not generally sufficient to determine if two terms are similar and consequently whether two documents. Two terms can be lexicographically different although they have the same meaning (eg. they are synonyms). The lack of common terms in two documents does not necessarily mean that the documents are irrelevant. Similarly, relevant documents may contain semantically similar but not necessarily the same terms. Semantically similar terms or concepts may be expressed in different words in the documents and the queries, and direct comparison between them is not effective (eg. VSM will not recognize synonyms or semantically similar terms). In this work we propose discovering semantically similar terms using the MeSH ontology for retrieving medical documents in Medline.

First we study several semantic similarity measures. We implemented a framework for studying and evaluating the performance of several similarity measures using MeSH [44] as the underlying reference ontology of medical terms. We also proposed a new measure that can be used to compute the semantic similarity between terms that belong to the same or different ontologies. This is a far more difficult problem compared to the one of measuring similarity between terms from a single ontology. An architecture and a system for evaluating the performance of the similarity measures

is also implemented. The system is available on the Web [1]. Building upon the idea of semantic similarity we also propose a document retrieval method suitable for retrieving documents from the Medline database [2].

Semantic similarity measures are classified into four main categories

1. **Edge Counting Measures :** measure the similarity between two concepts c_1, c_2 as a function of the path linking the terms in the taxonomy and of the position of the terms in the taxonomy.

2. **Information Content Measures :** measure the difference of information content of the two terms as a function of their probability of occurrence in a corpus.

3. **Feature based Measures :** measure the similarity between terms as a function of their properties or based on their relationships to other similar terms.

4. **Hybrid Measures :** combine the above ideas.

Semantic similarity measures can also be distinguished between

a) **Single ontology** measures assuming that he terms belong to the same ontology.

b) **Cross ontology** measures that are capable of comparing terms from different ontologies.

Because the structure and information content between different ontologies cannot be compared directly, cross ontology approaches usually call for hybrid or feature based measures. For example, two terms are similar if they have similar spelling or definition or they are related with other terms which are similar. Notice also that cross ontology methods may also be used to measure semantic similarity between terms from the same ontology.

We evaluated the performance of several measures based on their capability of providing results similar to results obtained by humans. We also studied, the problem of determining similarity of terms that belong in different ontologies. For example how much alike are the terms "pain" from the MeSH ontology with "pain" from the Wordnet ontology?

A novel information retrieval model based on the integration of semantic similarity measures in document matching, based on the MeSH ontology is also proposed.

[1] http://www.ece.tuc.gr/mesh_similarity

[2] http://www.nlm.nih.gov/pubs/factsheets/medline.html and http://medline.cos.com/

The main idea behind our approach is that different terms in a document or query representation are no longer considered as independent but they are related by virtue of their semantic similarity. The model suggests the idea of enhancing a text representation with other semantic similar terms. To measure similarity between text representation, we abandon the idea of vector similarity and we introduce a model based on a *Similarity Matrix* that better captures the notion of dependency (or similarity) between non-identical terms in the two documents which are matched. In the proposed model, terms in queries and in documents are treated as concepts whose similarity is computed algorithmically by semantic similarity measures rather than plain lexicographic matching as in the Vector Space Model (VSM). Initially our method computes $tf * idf$ weights to term representations of documents. These representations are then augmented by semantically similar terms. The weights of new and pre-existing terms are then recomputed and, finally, document similarity is computed by associating semantically similar terms in the documents and in the queries and by accumulating their similarities. Furthermore, we make use of the *semantic neighborhood* aspect in order to enrich the user query with terms that possibly interest the user.
Briefly, the proposed model has the following characteristics

1. *Expands* the Query Vector with other semantic similar terms
2. *Re-weights* the Query Terms based on their semantic similarity. This mechanism also computes the weights for terms that did not originally existed in the text
3. Ranks the results according to our proposed *Similarity Matrix Measure*

The rest of this thesis is organized as follows:

Chapter 2 will provide the necessary background and a critical analysis of related work.

Chapter 3 will define the semantic similarity problem, introduce our similarity model, describe the developed system and present the experimental results.

Chapter 4 will define the problem of information retrieval on the Web, introduce our *Similarity Matrix Model* based on semantic similarity, describe the system used in order to make the experiments and present the experiment results.

Chapter 5 will highlight the conclusions drawn from the study, and propose future work.

Appendix A will describe some technical issues about the developed software and get in detail about many unclear parts that took place in the implementation process.

Chapter 2

Background and Related Work

This chapter introduces the semantic similarity problem, highlights some of the key technical issues and discusses related work. It concludes by identifying critical questions that have not yet been adequately addressed in the literature.

2.1 Writing and Using Ontologies

Ontologies can be regarded as general tools of information representation on a subject. They can have different roles depending on the application domain and the level of specificity at which they are being used. In general, ontologies can be distinguished into *domain ontologies*, representing knowledge of a particular domain, and *generic ontologies* representing common sense knowledge about the world [45].

There are several examples of general purpose ontologies available including: (a) WordNet [1] [4, 29] attempts to model the lexical knowledge of a native speaker of English. It can be used as both a thesaurus and a dictionary. English nouns, verbs, adjectives, and adverbs are organized into synonym sets, called *synsets*, each representing a concept. (b) SENSUS [2] [20] is a 90,000-node concept thesaurus (ontology) derived as an extension and reorganization of WordNet. Each node is SENSUS represents one concept, i.e., one specific sense of a word, and the concepts are linked in a IS-A hierarchy, becoming more general towards the root of the ontology. (c) The Cyc [3] Knowledge Base (KB) [33, 35] consists of terms and assertions relating those terms, contains a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. At the present time, the Cyc KB contains nearly two hundred thousand terms and several dozen hand-entered assertions about/involving each term.

Examples of domain specific ontologies include among others ontologies designed

[1] http://www.cogsci.princeton.edu/∼wn/
[2] http://mozart.isi.edu:8003/sensus2/
[3] http://www.cyc.com/, http://www.opencyc.org/

around (a) medical concepts such as UMLS [4] [31], SNOMED [5], the MeSH [6] ontology [30] that we use in this study, (b) genomic data such as GO [7] [5, 12] and (c) spatial data such as SDTS [8]. The Unified Medical Language System (UMLS) contains a very large, multi-purpose and multi-lingual thesaurus concerning biomedical and health related concepts. In particular, it contains information about over 1 million biomedical concepts and 2.8 million concept names from more than 100 controlled vocabularies and classifications (some in multiple languages) used in patient records, administrative health data, bibliographic and full-text databases and expert systems. Furthermore, all the names and meanings are enhanced with attributes and inter-term relationships. UMLS includes other meta thesaurus source vocabularies, such as Medical Subject Headings (MeSH) that is the National Library of Medicine's vocabulary thesaurus. MeSH consists of sets of terms naming *descriptors* in a hierarchical structure. Gene Ontology (GO) is a structured network of defined terms that describe gene proteins and concerns all organisms. The Spatial Data Transfer Standard (SDTS) contains an ontology used to describe the underlying conceptual model and the detailed specifications for the content, structure, and format of spatial data, their features and associated attributes. Concepts in SDTS are commonly used on topographic quadrangle maps and hydrographic charts.

Intensive research efforts during the last few years have focused on providing tools for coherent, unambiguous and easy manipulation of information represented as ontologies. Such tools include languages providing the necessary syntax for the efficient representation of concepts and of their semantics as well as tools in the form of algorithms and graphic interfaces for viewing and manipulating the content of ontologies.

Languages for Writing Ontologies

The Resource Description Framework (RDF [9]) is a language for representing information about resources in the Web [1, 18]. It is particularly intended for representing meta data about Web resources, such as the title, author, and modification date of a document. RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved, as for exam-

[4] http://www.nlm.nih.gov/research/umls
[5] http://www.snomed.org
[6] http://www.nlm.nih.gov/mesh
[7] http://www.geneontology.org
[8] http://mcmcweb.er.usgs.gov/sdts/
[9] http://www.w3.org/RDF

ple information about items available from on-line shopping facilities (e.g., information about specifications, prices, and availability). RDF is intended for situations in which this information needs to be processed by applications, as it provides a common framework for expressing this information so it can be exchanged between applications without loss of meaning. RDF is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs), and describing resources in terms of simple properties and property values, which enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values. RDF also provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs. Although, RDF provides a way to express simple statements about resources, using named properties and values, it does not define the terms used in those statements. That is the role of RDF Schema (RDF-S [10]) that provides the facilities needed to describe such classes and properties, and to indicate which classes and properties are expected to be used together [25]. The RDF-S facilities are themselves provided in the form of an RDF vocabulary; that is, as a specialized set of predefined RDF resources with their own special meanings.

DAML+OIL [11], which was the result of an initial joint effort by US and European researchers, is a semantic markup language for Web resources [15, 14]. It builds on RDF and RDF-S, and extends these languages with richer modeling primitives. In particular, DAML+OIL assigns a specific meaning to certain RDF triples. The model-theoretic semantics [12] specify exactly which triples are assigned a specific meaning, and what this meaning is.

The WWW Consortium (W3C) created the Web Ontology Working Group to develop a semantic markup language for publishing and sharing ontologies and the resulting language is Web Ontology Language (OWL [13]). OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent content on the Web. OWL is a revision of the DAML+OIL Web ontology language, adding more relations between classes (e.g., disjointness), cardinality (e.g., "exactly one"), equality, more properties, more characteristics of properties (e.g., symmetry), and enumerated classes.

[10] http://www.w3.org/TR/rdf-schema
[11] http://www.daml.org/language/
[12] http://www.daml.org/2000/12/daml+oil.daml
[13] http://www.w3.org/TR/owl-features

To conclude, if machines are expected to perform useful reasoning tasks on Web resources, some language must be used in order to go beyond raw data, to express the semantics of the data and to extract knowledge from it. A summary of the existent recommendations related to the Semantic Web follows.

- XML provides a syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- RDF is a data model describing resources and relations between them and provides simple semantics for this data model. The data models can be represented in an XML syntax.
- RDF-S is a vocabulary for describing properties and classes of RDF resources.
- DAML+OIL assigns specific meaning to certain RDF triples.
- OWL adds more vocabulary for describing properties and classes.

There are also efforts for describing the semantics of Web services, resulting in the DAML-S [14] [39] and OWL-S [15] [17] languages.

Tools for Manipulating Ontologies

Examples of tools for manipulating ontologies include Protege-2000 [16] [32] and Chimaera [17] [26, 27, 28]. Protege-2000 allows users to construct domain ontologies, contains a platform that can be extended with graphical widgets for tables, diagrams, animation components to access other knowledge-based systems embedded applications, and has a library that other applications can use to access and display knowledge bases. Chimaera is a software system that supports users in creating and maintaining distributed ontologies on the Web. It supports two major functions that is merging multiple ontologies together and diagnosing [18] individual or multiple ontologies. It also provides users with tasks such as loading knowledge bases in different formats, reorganizing taxonomies, resolving name conflicts, browsing ontologies and editing terms.

[14] http://www.daml.org/services
[15] http://www.mindswap.org/2004/owl-s/
[16] http://protege.stanford.edu
[17] http://www.ksl.stanford.edu/software/chimaera/
[18] used as an ontological sketchpad, and creating classes for example.

2.2 MeSH

MeSH (Medical Subject Headings) [30, 44] is a taxonomic hierarchy of medical and biological terms suggested by the U.S National Library of Medicine (NLM) [19]. NLM has adopted the Extensible Markup Language (XML) [20] as the description langauge for MeSH. The MeSH vocabulary file is available in XML format. All terms in MeSH are organized in a hierarchy with most general terms (e.g "Chemicals and Drugs") higher in the taxonomy than most specific terms (e.g "Aspirin"). There are 21,973 main headings, termed descriptors, in MeSH (22,568 in 2004). Moreover, the structure of MeSH is a hierarchical tree, where a term can appear in different subtrees. There are 15 tree hierarchies (subtrees) in the MeSH ontology (see Figure A.4), of ISA kind of relationship between nodes (concepts) in each subtree. MeSH concepts correspond to MeSH objects which are described with terms of several properties (see chapter A.6 in page 71), the most important of them being:

MeSH Headings (MH): These are term names or identifiers. They are used in MEDLINE as the indexing terms for documents. Every document in Medline have some MeSH terms that are indexed with. A MH term belongs to a concept, and is preferred to label the meaning that the corresponding concept reflects; its use indicates the topic discussed by the document.

Entry Terms: These terms are used as pointers to the MH, there are mostly the synonym terms of the MH, naming the same concept, the MH. We say "mostly" because there is not quite a synonymy relation in those terms with the MH. In most cases it is, but there can be terms that designate the MH in an opposite way like "anions" and "cations". They are also referred to as *quasi-synonyms*. The set of entry terms that points to a MH are the terms that represent the concept introduced by the MH. So, we made an admission in this study that all entry terms are synonyms with the MH.

MeSH Tree Number: The tree numbers indicate the positions of the terms in the MeSH taxonomy. For example D is the code name of the "Chemical and drugs" subtree (1 of 15) and the term "Acids" has a tree number D01.029, meaning that "Acids" belongs to D subtree (see Figure A.4).

MeSH Scope Note: Mainly the text descriptions of the MeSH terms. This short

[19] http://www.nlm.nih.gov/
[20] http://www.w3.org/XML/

piece of free text provides a type of definition, in which the meaning of the MH is circumscribed.

Main Headings (descriptor records) are distinct in meaning from other Main Headings in the thesaurus (ie. their meanings do not overlap). Moreover, descriptor names reflect the broad meaning of the concepts involved. The hierarchical relationships can be intellectually accessible by users of MeSH (e.g., clinician, librarian, and indexer). An indexer is able to assign a given Main Heading to an article and a clinician can find a given Main Heading in the tree hierarchy. The relationship between entry terms and main headings is one of the most essential in the thesaurus.

2.3 Comparing Concepts

This section presents methods of computing the similarity between semantic entities with some semantic meaning, (eg. concepts or classes) represented in ontologies, or elements (i.e., resources) represented in schemas. These methods, referred to as *semantic similarity measures*, exploit the fact that the entities which are compared may have properties (e.g., in the form of attributes) associated with them, taking also into account the level of generality (or specificity) of each entity within the ontology as well as their relationships with other entities or concepts. Notice that, keyword-based similarity measures cannot use this information. Semantic similarity measures might be used for performing tasks such as retrieving results to user queries, for representation and for redundancy of retrieved resources, and for checking ontologies for consistency or coherency.

Semantic Similarity Measures

Many measures of semantic similarity with a variety of interesting properties have been proposed. In what follows, we present measures of similarity followed by a short discussion of their properties. Semantic similarity measures can be generally partitioned in four categories: those based on how close the two concepts in the taxonomy are, those based on how much information the two concepts share, those based on the properties of the concepts, and those based on combinations of the previous options.

Let $\mathcal{C}$ be the set of concepts in an IS-A taxonomy. We want to measure the similarity of two concepts $c_1, c_2 \in \mathcal{C}$.

2.3.1 Edge-Counting Measures

In the first category we place measures that consider *where* two concepts c_1 and c_2 are in the taxonomy. The following measures are based on a simplified version of *spreading activation theory* [8, 40]. One of the assumptions of the theory of spreading activation is that the hierarchy of concepts is organized along the lines of semantic similarity. Thus, the more similar two concepts are, the more links there are between the concepts and the more closely related they are [34].

Shortest path [34, 6]: The first measure has to do with how close in the taxonomy the two concepts are.

$$sim_{sp} = 2MAX - L \tag{2.1}$$

where MAX is the maximum path length between two concepts in the taxonomy and L is the minimum number of links between concepts c_1 and c_2. This measure is a variant on the *distance* method [34] and is principally designed to work with hierarchies. It is motivated by two observations: the behavior of conceptual distance resembles that of a metric, and the conceptual distance between two nodes is often proportional to the number of edges separating the two nodes in the hierarchy. A measure like this might be implemented in an information retrieval system that is based on indexing documents and queries into terms from a semantic hierarchy, or might be applied to help rank the documents to the query. There are many specific questions about the cognitive realism of shortest path measure, however it is a simple and powerful measure in hierarchical semantic nets.

Weighted links [37]: Extending the above measure, the use of weighted links is proposed to compute the similarity between two concepts. The weight of a link may be affected by: (a) the density of the taxonomy at that point, (b) the depth in the hierarchy, and (c) the strength of connotation[21] between parent and child nodes. Then, computing the distance between two concepts is translated into summing up the weights of the traversed links instead of counting them.

Hirst and St-Onge [13]: The idea behind this measure is that two concepts c_1 and c_2 are semantically close if they are connected by a path that is not too long

[21] The *connotation* of a term is the list of membership conditions for the denotation. The *denotation* of a term is the class of things to which the term correctly applies. For example, the connotation of the general term "square" is "rectangular and equilateral", while its denotation is all squares.

and that does not change direction too often.

$$sim_{H\&S}(c_1, c_2) = C - L - kd \tag{2.2}$$

where d is the number of changes of direction in the path, and C, k are constants. Although this measure gives a different perspective of similarity between two concepts, it seems to poorly perform (see [7]) mainly because it lies in its tendency to wander than in the use of concept relationships.

Wu and Palmer [48]: This similarity measure considers the position of concepts c_1 and c_2 in the taxonomy relatively to the position of the most specific common concept c. As there may be multiple parents for each concept, two concepts can share parents by multiple paths. The most specific common concept c is the common parent related with the minimum number of IS-A links with concepts c_1 and c_2.

$$sim_{W\&P}(c_1, c_2) = \frac{2H}{N_1 + N_2 + 2H} \tag{2.3}$$

where N_1 and N_2 is the number of IS-A links from c_1 and c_2 respectively to the most specific common concept c, and H is the number of IS-A links from c to the root of the taxonomy. It scores between 1 (for similar concepts) and 0.

Li et al. [22]: The following similarity measure, which was intuitively and empirically derived, combines the shortest path length between two concepts c_1 and c_2, L, and the depth in the taxonomy of the most specific common concept c, H, in a non-linear function.

$$sim_{Li}(c_1, c_2) = e^{-\alpha L} \cdot \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \tag{2.4}$$

where $\alpha \geq 0$ and $\beta > 0$ are parameters scaling the contribution of shortest path length and depth respectively. Based on [22] the optimal parameters are $\alpha = 0.2$ and $\beta = 0.6$. This measure is motivated by the fact that information sources are infinite to some extend while humans compare word similarity with a finite interval between completely similar and nothing similar. Intuitively the transformation between an infinite interval to a finite one is non-linear. It is thus obvious that this measure scores between 1 (for similar concepts) and 0.

Leacock and Chodorow [21]: The relatedness measure proposed by Leacock and Chodorow is

$$sim_{ch}(c_1, c_2) = \log\left(length/(2 \cdot D)\right) \tag{2.5}$$

where length is the length of the shortest path between the two synsets (using node-counting) and D is the maximum depth of the taxonomy. The fact that this measure takes into account the depth of the taxonomy in which the synsets are found means that the behavior of the measure is profoundly affected by the presence or absence of a unique root node. If there is a unique root node, then there are only one taxonomy: one for all the 15 subtrees of MeSH. If the root node is not being used, however, then there are 15 different subtrees, each with a different value for D. In this case it is possible for synsets to belong to more than one taxonomy. For example, the synset containing the term 'pain' belongs into 3 taxonomies: one rooted at 'diseases', one rooted at 'Psychiatry and Psychology' and one at 'Biological Sciences'. In such a case, the relatedness is computed by finding the LCS that results in the shortest path between the synsets. The value of D, then, is the maximum depth of the taxonomy in which the LCS is found. If the LCS belongs to more than one taxonomy, then the taxonomy with the greatest maximum depth is selected (i.e., the largest value for D).

The above mentioned measures are based only on taxonomic (IS-A) links between concepts, assuming that links in the taxonomy represent distances. However, the density of terms throughout the taxonomy is generally not constant. Typically, more general terms exist higher in the hierarchy and represent a smaller set of nodes than the larger number of more specific terms that populate a much denser space lower in the hierarchy. In Wordnet ontology for example , specify that distance between *plant* and *animal* is 2 (their common parent is *living thing*), and the distance between *zebra* and *horse* is also 2 (their common parent is *equine*). Intuitively *horse* and *zebra* seem more closely related than *plant* and *animal*. Using in our example either the *Wu & Palmer* measure, the measure based on the *Weighted Links* (if the link weights are fixed accordingly) or the *Li et al.* measure, we take into account the fact that the first two terms occupy a much higher place in the hierarchy than the latter two terms and the results will be more realistic. Furthermore, in taxonomies there is wide variability in what is covered by a single taxonomic link. For example, *safety valve* IS-A *valve* seems much narrower than *knitting machine* IS-A *machine*. The *Weighted Links* measure may take into account the strength of links if link weights are computed accordingly. Finally, experimental results presented in [22] have demonstrated that the *Li et al.* measure significantly outperforms previous measures.

In what follows, we present measures involving information content, which seem to perform better that edge-counting measures.

2.3.2 Information Content Measures

In this category, similarity measures are based on the *information content* of each concept. The notion of information content of the concept practically has to do with the frequency of the term in a given document collection. The frequencies of terms in the taxonomy are estimated using noun frequencies in some large (1,000,000 word) collection of texts [36]. Furthermore, the key to the similarity of two concepts is the extend to which they share information in common, indicated by a highly specific concept that subsumes them both.

Associating probabilities with concepts in the taxonomy, let the taxonomy be augmented by the function $p : \mathcal{C} \rightarrow [0, 1]$, such that for any concept $c \in \mathcal{C}$, $p(c)$ is the probability of encountering an instance of concept c. The concept probability is defined as $p(c) = freq(c)/N$, where N is the total number of terms in the taxonomy, $freq(c) = \sum_{n \in words(c)} n$ and $words(c)$ is the set of terms subsumed by c. This function implies that if c_1 IS-A c_2, then $p(c_1) \leq p(c_2)$, which intuitively means that the more general the concept is, the higher its associated probability. Then, the information content of a concept c can be quantified as the log likelihood, $-\ln p(c)$, which means that as probability increases, informativeness decreases, so the more abstract a concept, the lower its information content.

In order for Information Content measures to perform correctly, the limitation *if* c_1 *IS-A* c_2*, then* $p(c_1) \leq p(c_2)$ must be always satisfied. Although it is implied, the dependency of the above inequality on large text corpora, can not guarantee that this will always happen. For this reason we preferred to calculate directly the Information Content (IC) of a concept by using a method proposed by Nuno Seco [43]. This method of obtaining IC values rests on the assumption that the taxonomic structure of an ontology is organized in a meaningful and structured way, where concepts with many hyponyms convey less information than concepts that have less hyponyms or any at all (leaves). Likewise, concepts that are leaf nodes are the most informative in the taxonomy. The Information Content of a concept is commented as a function of the population of it's hyponyms.

$$ic(c) = \frac{\log \frac{hypo(c)+1}{max_c}}{\log \frac{1}{max_c}} \tag{2.6}$$

where the function *hypo* returns the number of hyponyms of a given concept and max_c is a constant that is set to the maximum number of concepts that exist in the taxonomy. The denominator which corresponds to the most informative concept normalizes all IC values in range [0...1]. The above formulation guarantees that the

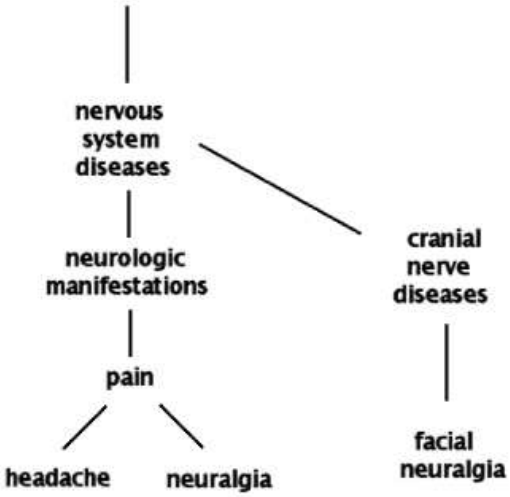

Figure 2.1: A fragment of the MeSH taxonomy

information content decreases monotonically as we traverse from the leaf nodes to the root nodes. Moreover, the information content of the imaginary top node of MeSH ontology would yield an Information Content value of 0.

Given these probabilities, several measures of semantic similarity, presented later in the section, have been defined. All these measures use the information content of the shared parents of two terms c_1 and c_2 (see Equation 2.7), where $S(c_1, c_2)$ is the set of concepts that subsume c_1 and c_2. As there may be multiple parents for each concept, two concepts can share parents by multiple paths. We take the minimum $p(c)$ when there is more than one shared parents, and then we call concept c the *most informative subsumer*.

$$p_{mis}(c_1, c_2) = min_{c \in S(c_1, c_2)}\{p(c)\} \tag{2.7}$$

For example, in Figure 2.1 *pain*, *neurologic manifestations*, etc. are all members of $S(neuralgia, headache)$, but the term that is structurally the minimal upper bound is *pain*, and will also be the most informative subsumer. The information content of the most informative subsumer will be used to quantify the similarity of the two words.

Lord et al. [24]: The first way to compare two terms is by using a measure that simply uses the probability of the most specific shared parent.

$$sim_{Lord}(c_1, c_2) = 1 - p_{mis} \tag{2.8}$$

The probability-based similarity score takes values between 1 (for the very similar concepts) and 0. It is used in order to access the extend to which similarity judgements might be sensitive to frequency per se, rather than information content.

Resnik [36]: The next measure uses the information content of the shared parents.

$$sim_{Resnik}(c_1, c_2) = -\ln p_{mis} \tag{2.9}$$

This measure signifies that the more information two terms share in common, the more similar they are, and the information shared by two terms is indicated by the information content of the term that subsume them in the taxonomy. As p_{mis} can vary between 0 and 1, this measure varies between infinity (for very similar terms) to 0. In practice, if N is the number of terms in the taxonomy, the maximum value of p_{mis} is $1/N$ (see Equation 2.7), and the maximum value of the measure is defined by $-\ln(1/N) = \ln(N)$. Thus, this measure provides us with information such as the size of the corpus; a large numerical value indicates a large corpus. Furthermore, the score from comparing a term with itself depends on where in the taxonomy the term is, with less frequently occurring terms having higher scores, and thus the measure reveals information about the usage within corpus of the part of the ontology queried.

Lin [23]: This measure uses both the amount of information needed to state the *commonality* of two terms and the information needed to fully *describe* these terms.

$$sim_{Lin}(c_1, c_2) = \frac{2 \ln p_{mis}(c_1, c_2)}{\ln p(c_1) + \ln p(c_2)} \tag{2.10}$$

As $p_{mis} \geq p(c_1)$ and $p_{mis} \geq p(c_2)$, the values of this measure vary between 1 (for similar concepts) and 0. In this case, a term compared with itself will always score 1, hiding the information revealed by the *Resnik* measure. However, the *Resnik* measure depends solely on the information content of the shared parents, and there are only as many discrete scores as there are ontology terms. By using the information content of both the compared terms and the shared parent the number of discrete scores is quadratic in the number of terms appearing in the ontology [24], thus augmenting the probability to have different scores for different pairs of terms. Consequently, using this measure to compare the terms of an ontology can have a better ranking of similarity than the *Resnik* measure.

Jiang et al. [16]: Contrary to the above similarity measures, this measure is of *semantic distance*.

$$dist_{Jiang}(c_1, c_2) = -2 \ln p_{mis}(c_1, c_2) - (\ln p(c_1) + \ln p(c_2)) \tag{2.11}$$

Thus, the similarity between two concepts c_1 and c_2, $sim_{Jiang}(c_1, c_2)$, is computed as $1 - dist_{Jiang}(c_1, c_2)$. This measure can give arbitrarily large values, like the *Resnik* measure, although in practice has a maximum value of $2 \ln(N)$, where N is the size of the corpus. Furthermore, it combines information content from the shared parent and the compared concepts, as the *Lin* measure. Thus, this measure seems to combine the properties of the above presented measures, i.e. provides us with both information about the size of the ontology and ranking of different term pairs.

2.3.3 Feature-Based Measures

Up to now, the features of the terms in the ontology are not taken into account. However, the features of a term contain valuable information concerning knowledge about the term. The following measure considers also the features of terms in order to compute similarity between different concept, while it ignores the position of the terms in the taxonomy and the information content of the term.

Tversky [46]: This measure is based on the *description sets* of the terms. We suppose that each term is described by a set of words indicating its properties or features. Then, the *more common* characteristics two terms have and the *less non-common* characteristics they have, the more similar the terms are.

$$sim_{Tversky}(c_1, c_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \kappa |C_1 \setminus C_2| + (\kappa - 1)|C_2 \setminus C_1|} \tag{2.12}$$

where C_1, C_2 correspond to description sets of terms c_1 and c_2 respectively and $\kappa \in [0, 1]$ defines the relative importance of the non-common characteristics. This measure scores between 1 (for similar concepts) and 0, it increases with commonality and decreases with the difference between the two concepts. In reverse to all the above presented measures, it has nothing to do with the taxonomy and the subsumers of the terms, and seems to better exploit the properties of the ontology used.

In the above presented measure, the determination of κ is based on the observation that similarity is not necessarily a symmetric relation: the common, as opposed to

the different, features between a subclass and its superclass have a larger contribution to the similarity evaluation than the common features in the inverse direction. Given this assumption, it provides a systematic approach to determine the asymmetry of a similarity evaluation.

2.3.4 Hybrid Measures

The next approaches used to compare two concepts c_1 and c_2 combine some of the above presented approaches, considering the path connecting the two terms in the taxonomy, the IS-A links of the terms with their parents in the graph and the features of the terms.

Rodriguez et al. [38]: This similarity measure can be used both for single or cross ontology similarities. The similarity function determines similar entity classes by using matching methods over *synonym sets*, *semantic neighborhoods* and *distinguishing features* that are further classified into *parts*, *functions* and *attributes* (eg. considering the term *college*, a function is *to educate*, its parts may be *roof* and *floor*, and other attributes can be *architectural properties*). The similarity function is a weighted sum of the similarity values for synonym sets, neighborhoods and features.

$$S(a^p, b^q) = \omega_w \cdot S_w(a^p, b^q) + \omega_u \cdot S_u(a^p, b^q) + \omega_n \cdot S_n(a^p, b^q) \tag{2.13}$$

where ω_w, ω_u and $\omega_n \geq 0$ and $\omega_w + \omega_u + \omega_n = 1$. For each type of distinguishing features, S_w, S_u and S_n a similarity function $sim_{Tversky}(c_1, c_2)$ is used based on the Tversky feature-matching model.

The functions S_w , S_u , and S_n are the similarity between synonym sets, features, and semantic neighborhoods between entity classes **a** of ontology **p** and **b** of ontology **q** and are calculated using Equation 2.14. Weights w_w, w_u, and w_n are the respective weights of the similarity of each specification component.

$$S(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha|A \setminus B| + (1 - \alpha)|B \setminus A|} \tag{2.14}$$

The difference between the above Equation 2.14 and the Tverksy function (Equation 2.12) is in the way κ is computed (in this method α). In this method α is computed according to Equation 2.15. In Tversky 2.12 function, κ defines the relative importance of the non-common characteristics, but here α is com-

puted as a factor of the depth where the two compared concepts are in each taxonomy.

$$\alpha(c_1, c_2) = \begin{cases} \frac{d(c_1, c_{mis})}{d(c_1, c_2)}, & d(c_1, c_{mis}) \leq d(c_2, c_{mis}); \\ 1 - \frac{d(c_1, c_{mis})}{d(c_1, c_2)}, & d(c_1, c_{mis}) > d(c_2, c_{mis}). \end{cases} \tag{2.15}$$

where $d(c_1, c_2) = d(c_1, c_{mis}) + d(c_2, c_{mis})$.

Knappe [19]: This measure is primarily based on the aspect that there may be *multiple paths* connecting two concepts. Taking all possible paths involves a substantial increase in complexity. Thus, the general idea puts emphasis on the "shared" concepts and a similarity measure representing the part of the ontology covering the compared concepts is defined. Furthermore, there is the notion of complex concepts that allows a concept to be constituted by more that one term.

Initially, the *term decomposition* $\tau(c)$ of a concept c into a set C is defined, and then the *upwards expansion* $\varpi(C)$ of C is performed. The term decomposition of c is defined as the set of all concepts included in c (if c is a complex concept, otherwise this set includes only c) and all attributes of these concepts. If for example, the initial concept c is "dog" the term decomposition could be the set $C = \{dog, colour\}$. The upwards expansion, $\varpi(C)$, involves the IS-A links of all elements in C.

Let $u(c)$ be the set of nodes upwards reachable from c, that is $u(c) = \varpi(\tau(c))$. The reachable nodes shared by both c_1 and c_2 are $u(c_1) \cap u(c_2)$. Then, we consider the upward and downward directions in the graph as *generalization* and *specialization* respectively. Three major desirable properties are considered in defining the similarity function: (a) the cost of generalization should be significantly higher than the cost of specialization, indicating that the similarity function cannot be symmetrical, (b) the cost for traversing edges should be lower when nodes are more specific and (c) further specialization implies reduced similarity.

$$sim_{Knappe}(c_1, c_2) = \rho \frac{|u(c_1) \cap u(c_2)|}{|u(c_1)|} + (1 - \rho) \frac{|u(c_1) \cap u(c_2)|}{|u(c_2)|} \tag{2.16}$$

where $\rho \in [0, 1]$ determines the degree of influence of generalizations.

This measure scores between 1 (for matching concepts) and 0. The purpose of this similarity measure is to introduce soft rather than crisp evaluation, since we

usually want to look for similar rather than exactly matching values. Furthermore, the idea of concept expansion leads the similarity matching towards a set comparison, incorporating in the similarity measure the knowledge represented by the ontology.

The key properties of the similarity measures presented in the previous sections are summarized in Table 2.1. We consider whether the similarity measures are affected by the common characteristics of the compared concepts and whether the differences between the concepts cause the measures to decrease. Furthermore, we think the relation of the similarity measures with the taxonomy and the taxonomic relations, i.e. whether the position of the concepts in the taxonomy and the number of IS-A links are considered. It is also presented whether the similarity measures are taking into account the information content of the concepts, whether they are bounded or return infinite values, whether they are symmetric (i.e., $sim(c_1, c_2) = sim(c_2, c_1)$), and whether they give different perspectives.

Property	*Knappe*	*Rodriguez*	*Tversky*	*Jiang*	*Lin*	*Resnik*	*Lord*	*Li*	*Wu & Palmer*	*Hirst & St-Onge*	*shortest path*
increase with commonality	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	no
decrease with difference	no	yes	yes	yes	yes	no	no	yes	yes	yes	yes
information content	no	no	no	yes	yes	yes	yes	no	no	no	no
position in hierarchy	yes	no	no	yes	yes	yes	yes	yes	yes	yes	yes
path length	no	yes	no	no	no	no	no	yes	yes	yes	yes
max value = 1	yes	yes	yes	no	yes	no	yes	yes	yes	no	yes
symmetric	no	no	no	yes	yes	yes	yes	yes	yes	no	yes
different perspectives	yes	yes	yes	yes	yes	no	no	yes	yes	yes	no

Table 2.1: Comparison between similarity measures

Chapter 3

Problem Definition and Proposed Solution

3.1 Semantic Similarity Measures on MeSH

It is commonly argued that language semantics are mostly captured by nouns (and noun phrases) so that it is common to built retrieval methods based on noun representations extracted from documents and queries. MeSH terms and their Is-A relationships are nouns, mostly noun phrases. Several methods for determining semantic similarity between terms have been proposed in the literature and most of them have been tested on WordNet. Similar results on MeSH havent been reported in the literature. Semantic similarity methods are classified into four main categories, while the methodology for each one category is described as follows:

Edge-Counting Measures: The base idea in those measures is to find the shortest path that links the two terms or the most specific common term with the minimum number of Is-A links. Moreover, for the *weightedLinks* measure we also need to specify the maximum depth for each of the 15 sub-trees of the MeSH ontology and assign a weight for each term of the shortest path according to its position in its subtree hierarchy. Having computed the features for the desired measure respectively we can easily derive to a result of a semantic similarity as introduced in section 2.3.1 at page 11.

Information Content Measures: In these measures we need to compute the Information Content (IC) value of the *Most Informative Subsumer* as it described in section 2.3.2 at page 14. Actually the IC should be computed by the frequency of a term in a collection of documents, in a large text corpus, in other words by statistical analysis in a specific corpus. We needed though this IC value to be independent from a corpus, so we compute the IC value for each MeSH term by the function described in [43], by the number of hyponyms of a term caompared to the total number of hyponyms of the subtree taxonomy

(total 15 MeSH subtrees) that the term belongs.

$$IC_{term}(c) = 1 - \frac{\log\left(hypo(c) + 1\right)}{\log\left(max_{terms}\right)} \tag{3.1}$$

where the function *hypo* returns the number of hyponyms of a given concept c and max_{terms} is a constant that is set to the maximum number of concepts that exist in the selected subtree taxonomy.

Then the computation of any similarity measure of this category between two terms is simple a simple operation denoted by the similarity function of each measure.

Hybrid Measures: The only confusing part in the rodriguez similarity measure is what features there are in the MeSH ontology that we can use. *Part* features in MeSH ontology do not exist. Also *function* features do not exist (based on the sense of the Wordnet Ontology structure as described in section A.4 in page 60). Finally neither *attribute* features exist in the MeSH ontology. We are only able to extract information about hypernyms, hyponyms and definitions. So, the result of the rodriguez similarity measure is computed by a Tversky-like function (common and different characteristics of the compared terms) for each of the two actually functions, S_w and S_n , that take place in the specified measure.

3.2 Proposed Similarity Measure

The Rodriguez [38] similarity measure do not follow the restriction that both terms are from the same ontology, like other measures do. The proposed similarity function (Equation 3.2) determines similar entity classes, or more simply calculates the similarity between two terms, by using a matching process over *synonym sets*, *semantic neighborhoods* and *distinguishing features*.

$$S(a^p, b^q) = \omega_w \cdot S_w(a^p, b^q) + \omega_u \cdot S_u(a^p, b^q) + \omega_n \cdot S_n(a^p, b^q) \tag{3.2}$$

Specific weight values are not mentioned anywhere in the literature for a promising similarity result and can only be specified experimentally.

Experimental results using different ontologies indicate that the model gives good enough results when ontologies have complete and detailed representation of entity classes. Also, the combination of word matching and semantic neighborhood matching

is adequate for detecting equivalent entity classes and feature matching allows for discriminating among similar but not necessarily equivalent entity classes.

We find their approach very promising, and building upon their method we propose a method that can be used for computing semantic similarity between terms that belong to MeSH ontology as well as comparing terms that belong to different ontologies (ie. Wordnet and MeSH). We propose modifying the Rodriguez et al. measure in the following ways:

- Glosses come with every synonym set (synset) in most ontologies providing us with a description of the term meaning or a scope note for the referred term and it is found in most ontologies including MeSH and Wordnet. This kind of information should also be taken into account for computing the similarity between terms. We propose to replace Feature Mathcing (S_u) with Gloss Similarity that is computed by the following formula

$$S_{gloss}(a^p, b^q) = \frac{|A \cap B|}{|A \cup B|} \tag{3.3}$$

  where A and B are sets containing terms form the glosses of concepts a^p, b^q.

- In the original method, *semantic neighborhood matching* is computed in the following way: the similarity function puts all terms in radius r from term a (similar terms above similarity threshold r) into a single set and compares them to another set that contains terms in the same radius of term b. The similarity between the two sets is computed again using Equation 2.14. We propose the following formula in order to calculate *semantic neighborhood matching*

$$S_{nm}(a^p, b^q) = \max_i \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \tag{3.4}$$

  where i denotes relations (ie. hyponyms, part-meronyms, holonyms and so on, in the Wordnet ontology while in MeSH ontology there are only hyponyms and hypernyms) that both a^p and b^q have. A_i, B_i are sets containing all terms derived from the i-th relation of concepts a^p, b^q. In other words, we propose taking the maximum similarity between the two terms in one part of the neighborhood and not in all the area. That is the base idea, but while in MeSH there are only a hyponymy and hypernymy (plus definition-description) relation from the denoted Wordnet relations for a concept, the above equation 3.4 will derive a single similarity value, the one of descendant or antecedent terms.

- No information regarding the structure of the ontologies should be taken into account when comparing concepts from different ontologies. A set similarity is used instead of the Tversky-like function in the cross ontology experiment.

- Combining the above formulas in a linear way we propose 3.5 formula in order to compute Semantic Similarity

$$S_{proposed}(a,b) = \omega_w S_w(a^p, b^q) + \omega_{gloss} S_{gloss}(a,b) + \omega_{nm} S_{nm}(a^p, b^q) \qquad (3.5)$$

This is rather an adaptation of the Rodriguez [38] similarity measure for use with MeSH (and WordNet) ontology rather than a new similarity measure. We believe that our modified version of Rodriguez et al. measure will achieve a good performance towards many other similarity measures, when a comparison of concepts from one ontology (WordNet or MeSH) or different ontologies (MeSH and Wordnet) takes place.

3.3 Evaluation of Semantic Similarity Measures

We evaluated the results obtained by applying the semantic similarity measures discussed in section 2.3 at page 10, by correlating their similarity scores with the scores obtained by human judgments. In accordance with previous research we evaluated the results by applying the semantic similarity measures of section 2.3 withthe similarity scores obtained by human judgements as in the experiments by Miller and Charles [29]. We asked Dr. Qiufen Qi [1], an independent medical expert at Dalhousie University to compile a set MeSH term pairs. Dr. Qiufen Qi proposed a a set of 49 pairs. Their similarity was evaluated by doctors, giving a score to each pair between 0 (not similar) and 4 (perfect similarity). The average rating (by all doctors) of each pair represents an estimate of how similar each pair is according to human judgement. The similarity values obtained by all competitive computational methods are correlated with the average scores obtained by the humans. The higher the correlation of a method the better the method is (i.e, the more it approaches the results of human judgements).

3.3.1 Experiment

We created a form based Web page, containing 50 term pairs. This form is available on the Web at *http://www.intelligence.tuc.gr/mesh/* and is still accepting results

[1] http://users.cs.dal.ca/ qiufen

by experts. Those term pairs that where selected by Dr. Qiufen Qi at Dalhouse University are terms from MeSH ontology except one term which we had to exclude. For the remaining 49 pairs, each evaluator (doctors in most cases) were asked to provide a similarity value between 0 to 4 where:

Score 0: The terms are completely unrelated (there is no relation between them)

Score 1: The terms are almost unrelated (not completely unrelated)

Score 2: The terms mean different things. Sometimes (e.g. in some medical cases) the terms are somehow related

Score 3: The terms are very similar (not exactly similar)

Score 4: The terms have the same meaning (the one can be used in place of the other)

Each user had to evaluate the all pairs and submit the results to our database (Figure 3.1).

The analysis of the results revealed that additional factors had to be taken into account:

- Some medical terms are more involved, or ambiguous leading to ambiguous evaluation. We had to exclude these terms from the experiment.
- Users often gave different, or out of the common bound of similarity, values that all other users submitted. Some medical experts were not all at the same high level of experience as others and gave non-reliable results.

The evaluation needed to be treated in such way that only the reliable term pairs and users should be taken into account of the evaluation. For reliability reasons we provided two criteria, one that could provide us with the reliable term pairs, and a second one that could provide us with the reliable users. Reliable term pairs and users were only taken into account for the evaluation.

Users integrity

The main idea is to exclude users that gave significantly different results than the majority of the others.

Figure 3.1: Evaluation form on the Web

$$U_{reliability} = \sum_{i=1}^{49} |U_i - \mu_{user}| \tag{3.6}$$

The above equation's meaning is that users integrity value ($U_{reliability}$) is computed as the sum for all pairs of the absolute value (U_i) that the user submitted for the pair i, minus the mean value (μ_{user}) of all the other users except his for the specified pair i.

For each user we get a ($U_{reliability}$) value. We do not take into account users that deviate much from the mean ($U_{reliability}$) value. So far we have evaluations from 12 doctors but the evaluation form is still online so we still hope for more users to distribute into the experiment. The integrity values per user is shown in Figure 3.2

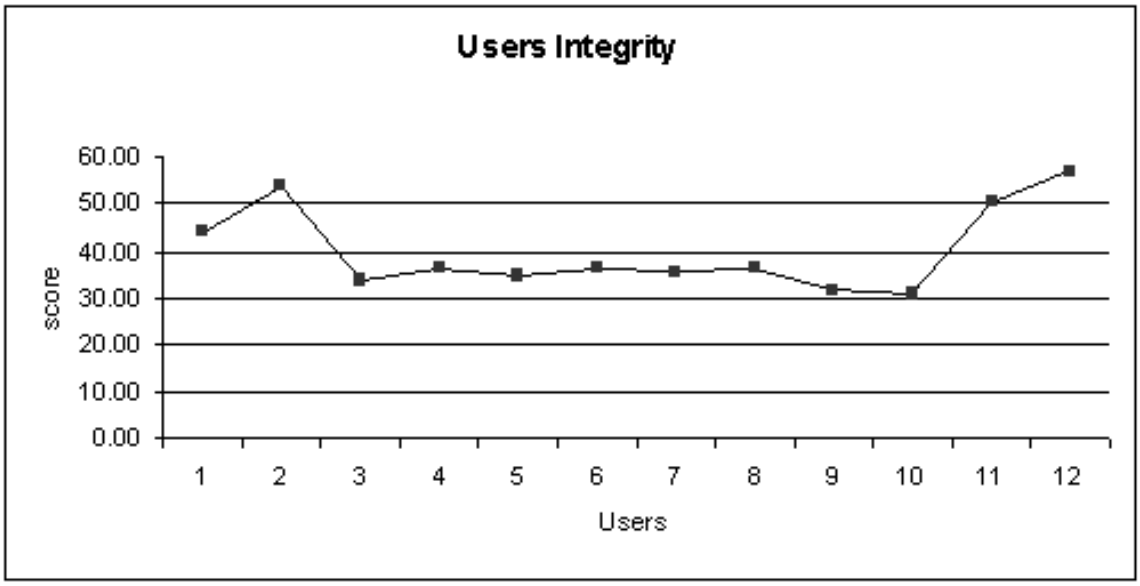

Figure 3.2: User integrity Diagram

It is obvious that users with id *1*, *2*, *11* and *12* are unreliable because they do refrain from the others in the curve. Their evaluation was not taken into account for this experiment.

Term pairs integrity

The main idea is to disregard term pairs with standard deviation σ (computed over all reliable users) higher than a specified user defined threshold t. We put $t = 0.8$ so that we did not disregard many term pairs.

So, the function for the term pairs integrity is computed as follows

$$TermPair_{reliability} = \begin{cases} \sigma > 0.8 & \text{,unreliable} \\ \sigma < 0.8 & \text{,reliable} \end{cases} \tag{3.7}$$

The integrity values (σ) per term pair is shown in Figure 3.3. As you can see from the diagram there are quite enough term pairs with $\sigma > 0.8$ that were not taken into account for this experiment.

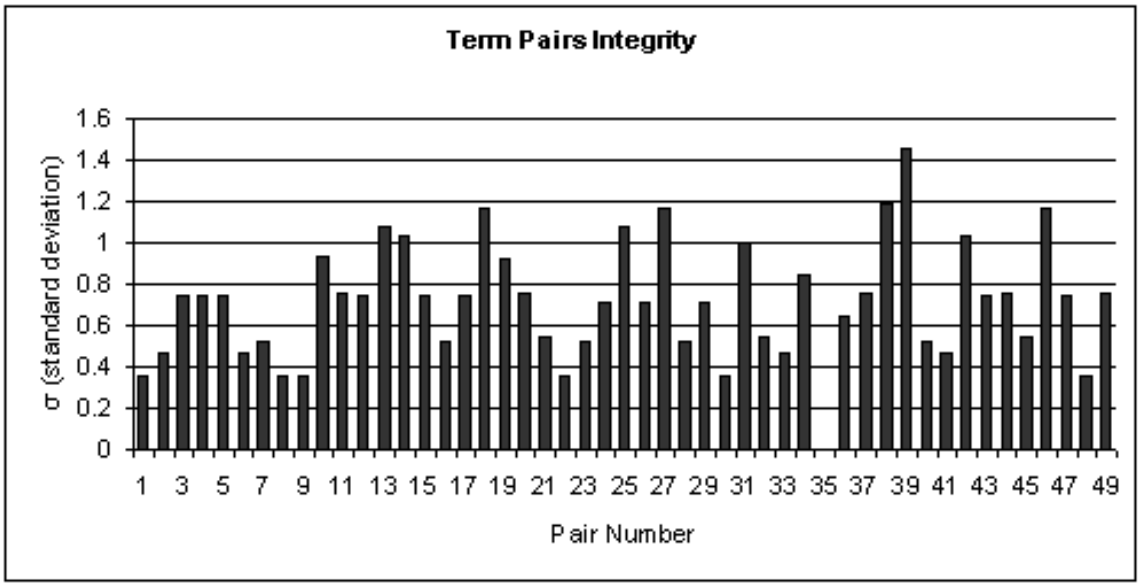

Figure 3.3: User integrity Diagram

The overall results of the analysis are:

- 4 out of the 12 users were disqualified. Only the evaluation of the rest was taken into account.
- 13 out of 49 term pairs were also disqualified. Only the evaluation by the reliable users for the rest of the pairs was taken into account.

The reliable term pairs are indicated in section A.7 on page 74.

3.3.2 Cross ontology experiment

The cross-ontology similarity experiment was designed in order to evaluate the performance of measures that can be used in such experiments, like our proposed method. We decided to experiment with WordNet and MeSH ontologies. Forty pairs of medical terms were carefully selected. These pairs were given for evaluation as described in the previous section, in order to calculate the degree of similarity between them.

We decided to perform the experiment in the following way: Each term of a pair belongs to a different ontology (ie. one term is from the MeSH and the other from the WordNet ontology). Then, we apply the similarity measure in each pair and retrieve the results. The performance of each measure is evaluated again as the correlation of these results with the human judgement.

3.4 Experimental Results

In section 3.3 in page 24 we mentioned that we wanted to evaluate the semantic similarity measures by correlating their similarity scores with the scores obtained by human judgments on a given set of MeSH terms (pairs). Correlation was computed using the Pearsons correlation function (Equation 3.8). Suppose we have two variables X and Y, with means $\overline{X}$ and $\overline{Y}$ respectively and standard deviations σ_X and σ_Y respectively. The correlation is computed as

$$r = \frac{\sum_{i-1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)\sigma_X\sigma_Y} \tag{3.8}$$

So as X and Y in our experiment stands for the similarity values of each of the similarity measures and of the human evaluation respectively. Table 3.1 presents results from*edge-counting* similarity measures. Table 3.2 present results obtained by *Information Content* similarity measures. Table 3.3 present the results we obtained from *Hybrid* measures (combinational), including our proposed method and *feature based measures*, the one proposed by *Tversky* [46]. Results from the cross ontology experiment are presented in Table 3.4.
Below is a list of the similarity measures that we tested in this experiment.

- H — Human Judgment
- SP — Shortest Path method (Edge Count)
- WP — Wu and Palmer method (Edge Count)
- Li — Li et al method (Edge Count)
- LC — Leacock and Chodorow method (Edge Count)
- WL — Shortest Path with Weighted Links method (Edge Count)
- Res — Resnik proposed method (IC)

- Lin — Lin method (IC)
- Lord — Lord et al method (IC)
- J — Jiang et al method (IC)
- Rodz — Rodriguez proposed method (Hybrid)
- T — Tversky function as similarity method (Feature Based)
- Our — Our proposed method based on Rodriguez (Hybrid)

From the *Edge Counting Measures*, the one by Leacock and Chodorow [21] performs quite well and we believe it is very promising. It has a correlation score of 0.74. We also believe that the performance of this method can be explained from the fact that it takes into account some very important facts : 1) the depth of the taxonomy (ie. "how deep an ontology is?"). The deeper the terms are, the more specific they are, and 2) the shortest path length between the compared terms.

Information Content Measures perform as well as the Edge Counting ones, and specially Lin's [23] measure is very close to what Resnik [36] proposed, where not only the IC of the MIS must be taken into account but the Ic of the compared terms as well. Their correlation values are 0.723 and 0.718 respectively, while the other measures do keep up .

Regarding *Hybrid Measures*, our proposed measure mentioned in section 3.2 in page 22 , performed as good as the other measures in this family. Regarding the single ontology experiment, the correlation of (0.69) is not as high as the results obtained by *Leacock & Chodorow* and *Lin* measures, but performs as good as the original Rodriguez measure. This means that the modifications we propose did not have a negative effect on it's performance. As for the cross ontology experiment, *measure1* and *measure2* are the modified rodriguez measure proposed, where the functions that exist in the measure are actually set similarities, instead of Tversky functions, for synsets, hypernyms, hyponyms and definitions. Their difference is that *measure1* have the same weight for every function, though *measure2* has the maximum value of all functions. Moreover in *measure2* we agreed that if a term inside the synset (synonym set) of the first compared concept is equivalent with one term of the synset of the second concept, then their similarity value must be 1, meaning that the compared concepts are actually synonyms. There is an increment (12%) in the performance compared with *measure1* and an overall correlation of **r=0.6972**. We find this prommising for several reasons. This method is suitable for cross ontology

Term Pairs		H	SP	WP	LC	WL	Li
Anemia - Appendicitis		0.031	0.65	0.22	0.91	0.61	0.13
Dementia Atopic Dermatitis		0.062	0.6	0.2	0.79	0.58	0.1
Bacterial Pneumonia - Malaria		0.156	0.65	0.22	0.91	0.61	0.13
Osteoporosis - Patent Ductus Arteriosus		0.156	0.6	0.2	0.79	0.58	0.1
Amino Acid Sequence - Anti-Bacterial Agents		0.156	0	0	1.1	0.31	0
Acquired Immunodeficiency Syndrome - Congenital Heart Defects		0.062	0.7	0.25	1.05	0.65	0.16
Otitis Media - Infantile Colic		0.156	0.55	0.18	0.69	0.55	0.08
Meningitis - Tricuspid Atresia		0.031	0.65	0.22	0.91	0.61	0.13
Sinusitis - Mental Retardation		0.031	0.65	0.22	0.91	0.61	0.13
Hypertension - Kidney Failure		0.5	0.65	0.22	0.91	0.61	0.13
Hyperlipidemia - Hyperkalemia		0.156	0.85	0.66	1.61	0.88	0.51
Hypothyroidism - Hyperthyroidism		0.406	0.9	0.75	1.89	0.92	0.63
Sarcoidosis - Tuberculosis		0.406	0.5	0.16	0.59	0.53	0.07
Vaccines - Immunity		0.593	0	0	1.5	0.43	0
Asthma - Pneumonia		0.375	0.85	0.66	1.6	0.88	0.52
Diabetic Nephropathy - Diabetes Mellitus		0.5	0.95	0.85	2.3	0.95	0.77
Lactose Intolerance - Irritable Bowel Syndrome		0.468	0.75	0.61	1.2	0.85	0.36
Urinary Tract Infection - Pyelonephritis		0.656	0.8	0.6	1.38	0.86	0.42
Neonatal Jaundice - Sepsis		0.187	0.7	0.25	1.05	0.65	0.16
Sickle Cell Anemia - Iron Deficiency Anemia		0.437	0.75	0.61	1.2	0.85	0.36
Psychology - Cognitive Science		0.593	0.972	0.88	2.89	0.97	0.8
Adenovirus - Rotavirus		0.437	0.75	0.54	1.2	0.83	0.35
Migraine - Headache		0.718	0.6	0.33	0.79	0.7	0.17
Myocardial Ischemia - Myocardial Infarction		0.75	0.95	0.9	2.3	0.97	0.8
Hepatitis B - Hepatitis C		0.562	0.9	0.83	1.89	0.94	0.66
Carcinoma - Neoplasm		0.75	0.85	0.57	1.61	0.87	0.45
Pulmonary Valve Stenosis - Aortic Valve Stenosis		0.531	0.9	0.8	1.89	0.93	0.66
Failure to Thrive - Malnutrition		0.625	0.65	0.22	0.91	0.61	0.13
Breast Feeding - Lactation		0.843	0.75	0.18	1.28	0.6	0.08
Antibiotics - Antibacterial Agents		0.937	1	1	3.58	1	0.99
Seizures - Convulsions		0.843	0.95	0.9	2.3	0.97	0.81
Pain - Ache		0.875	1	1	2.99	1	0.99
Malnutrition Nutritional Deficiency		0.875	1	1	2.99	1	0.98
Measles - Rubeola		0.906	1	1	2.99	1	0.99
Chicken Pox - Varicella		0.968	1	1	2.99	1	0.99
Down Syndrome Trisomy 21		0.875	1	1	2.99	1	0.99
Correlation		1	0.509	0.679	**0.740**	0.640	**0.705**

Table 3.1: Correlation of Edge Counting Measures

Term Pairs		H	Lin	Lord	J	Res
Anemia - Appendicitis		0.031	0	0	0.19	0
Dementia Atopic Dermatitis		0.062	0	0	0.16	0
Bacterial Pneumonia - Malaria		0.156	0	0	0.29	0
Osteoporosis - Patent Ductus Arteriosus		0.156	0	0	0.03	0
Amino Acid Sequence - Anti-Bacterial Agents		0.156	0	0	0.15	0
Acquired Immunodeficiency Syndrome - Congenital Heart Defects		0.062	0	0	0.27	0
Otitis Media - Infantile Colic		0.156	0	0	0.07	0
Meningitis - Tricuspid Atresia		0.031	0	0	0.19	0
Sinusitis - Mental Retardation		0.031	0	0	0.36	0
Hypertension - Kidney Failure		0.5	0	0	0.21	0
Hyperlipidemia - Hyperkalemia		0.156	0.39	0.286	0.47	0.33
Hypothyroidism - Hyperthyroidism		0.406	0.72	0.48	0.75	0.65
Sarcoidosis - Tuberculosis		0.406	0	0	0.25	0
Vaccines - Immunity		0.593	0	0	0.52	0
Asthma - Pneumonia		0.375	0.8	0.4	0.87	0.52
Diabetic Nephropathy - Diabetes Mellitus		0.5	0.74	0.44	0.79	0.58
Lactose Intolerance - Irritable Bowel Syndrome		0.468	0.47	0.37	0.47	0.47
Urinary Tract Infection - Pyelonephritis		0.656	0.6	0.37	0.67	0.47
Neonatal Jaundice - Sepsis		0.187	0	0	0.19	0
Sickle Cell Anemia - Iron Deficiency Anemia		0.437	0.72	0.45	0.76	0.6
Psychology - Cognitive Science		0.593	0.77	0.46	0.81	0.62
Adenovirus - Rotavirus		0.437	0.32	0.23	0.45	0.26
Migraine - Headache		0.718	0.26	0.2	0.37	0.23
Myocardial Ischemia - Myocardial Infarction		0.75	0.84	0.43	0.89	0.57
Hepatitis B - Hepatitis C		0.56	0.82	0.47	0.86	0.64
Carcinoma - Neoplasm		0.75	0.62	0.21	0.85	0.24
Pulmonary Valve Stenosis - Aortic Valve Stenosis		0.531	0.78	0.48	0.81	0.65
Failure to Thrive - Malnutrition		0.625	0	0	0.18	0
Breast Feeding - Lactation		0.843	0	0	0.04	0
Antibiotics - Antibacterial Agents		0.937	1	0.63	1	1
Seizures - Convulsions		0.843	0.89	0.55	0.9	0.8
Pain - Ache		0.875	1	0.57	1	0.86
Malnutrition Nutritional Deficiency		0.875	1	0.46	1	0.62
Measles - Rubeola		0.906	1	0.6	1	0.92
Chicken Pox - Varicella		0.968	1	0.63	1	1
Down Syndrome Trisomy 21		0.875	1	0.63	1	1
Correlation		1	**0.723**	0.701	0.710	**0.718**

Table 3.2: Correlation of IC based Measures

Term Pairs		H	T	Rodz	Our
Anemia - Appendicitis		0.031	0.2	0.1	0.113
Dementia Atopic Dermatitis		0.062	0.2	0.1	0.092
Bacterial Pneumonia - Malaria		0.156	0.2	0.1	0.091
Osteoporosis - Patent Ductus Arteriosus		0.156	0.2	0.1	0.099
Amino Acid Sequence - Anti-Bacterial Agents		0.156	0	0	0.02
Acquired Immunodeficiency Syndrome - Congenital Heart Defects		0.062	0.25	0.125	0.099
Otitis Media - Infantile Colic		0.156	0.16	0.083	0.077
Meningitis - Tricuspid Atresia		0.031	0.2	0.1	0.081
Sinusitis - Mental Retardation		0.031	0.2	0.1	0.078
Hypertension - Kidney Failure		0.5	0.2	0.1	0.077
Hyperlipidemia - Hyperkalemia		0.156	0.6	0.3	0.21
Hypothyroidism - Hyperthyroidism		0.406	0.75	0.375	0.27
Sarcoidosis - Tuberculosis		0.406	0.14	0.07	0.069
Vaccines - Immunity		0.593	0	0	0.0458
Asthma - Pneumonia		0.375	0.6	0.3	0.225
Diabetic Nephropathy - Diabetes Mellitus		0.5	0.75	0.375	0.176
Lactose Intolerance - Irritable Bowel Syndrome		0.468	0.57	0.78	0.558
Urinary Tract Infection - Pyelonephritis		0.656	0.5	0.25	0.165
Neonatal Jaundice - Sepsis		0.187	0.25	0.125	0.102
Sickle Cell Anemia - Iron Deficiency Anemia		0.437	0.57	0.285	0.231
Psychology - Cognitive Science		0.593	0.8	0.4	0.258
Adenovirus - Rotavirus		0.437	0.5	0.25	0.195
Migraine - Headache		0.718	0.28	0.143	0.124
Myocardial Ischemia - Myocardial Infarction		0.75	0.83	0.58	0.384
Hepatitis B - Hepatitis C		0.562	0.83	0.416	0.348
Carcinoma - Neoplasm		0.75	0.4	0.2	0.105
Pulmonary Valve Stenosis - Aortic Valve Stenosis		0.531	0.8	0.4	0.264
Failure to Thrive - Malnutrition		0.625	0.2	0.1	0.066
Breast Feeding - Lactation		0.843	0.16	0.083	0.055
Antibiotics - Antibacterial Agents		0.937	1	1	1
Seizures - Convulsions		0.843	0.83	0.75	0.48
Pain - Ache		0.875	1	1	1
Malnutrition Nutritional Deficiency		0.875	1	1	1
Measles - Rubeola		0.906	1	1	1
Chicken Pox - Varicella		0.968	1	1	1
Down Syndrome Trisomy 21		0.875	1	1	1
Correlation		1	0.67	**0.71**	**0.69**

Table 3.3: Performance of Hybrid Methods

WordNet Term	MeSH term		Human	measure1	measure2
Anemia	Appendicitis		0.0312	0	0
Dementia	Atopic Dermatitis		0.0625	0	0
Bacterial Pneumonia	Malaria		0.1562	0.028	0.113
Osteoporosis	Patent Ductus Arteriosus		0.1562	0.0306	0.122
Immunodeficiency Syndrome	Congenital Heart Defects		0.0625	0.021	0.084
Otitis Media	Infantile Colic		0.1562	0	0
Meningitis	Tricuspid Atresia		0.0312	0.006	0.025
Sinusitis	Mental Retardation		0.0312	0	0
Hyperlipidemia	Hyperkalemia		0.1562	0.045	0.182
Hypothyroidism	Hyperthyroidism		0.4062	0.096	0.387
Sarcoidosis	Tuberculosis		0.4062	0	0
Asthma	Pneumonia		0.375	0.026	0.07
Diabetic Nephropathy	Diabetes Mellitus		0.5	0.065	0.205
Lactose Intolerance	Irritable Bowel Syndrome		0.4687	0.01	0.047
Urinary Tract Infection	Pyelonephritis		0.6562	0.0075	0.03
Neonatal Jaundice	Sepsis		0.1875	0	0
Sickle Cell Anemia	Iron Deficiency Anemia		0.4375	0.044	0.14
Psychology	Cognitive Science		0.5937	0.069	0.25
Adenovirus	Rotavirus		0.4375	0.0558	0.16
Migraine	Headache		0.7187	0.011	0.042
Myocardial Ischemia	Myocardial Infarction		0.75	0.1188	0.47
Hepatitis B	Hepatitis C		0.5625	0.118	0.42
Carcinoma	Neoplasm		0.75	0.072	0.17
Failure to Thrive	Malnutrition		0.625	0.0108	0.043
Breast Feeding	Lactation		0.8437	0	0
Antibiotics	Antibacterial Agents		0.9375	0.022	1
Pain	Ache		0.875	0.063	1
Malnutrition	Nutritional Deficiency		0.875	0.13	1
Chicken Pox	Varicella		0.9687	0.25	1
Down Syndrome	Trisomy 21		0.875	0.18	1
Correlation			1	0.5738	**0.6972**

Table 3.4: Cross Ontology experiment results

similarity matching, making no apriori assumption on the structure and the properties of the ontologies where the terms belong to. Also, the correlation of 0.69 in the single ontology experiment in not bad compared with all the other methods. By further investigating the subject we believe that it's performance may increase even more.

3.5 System architecture

Below we describe how the similarity measures software works in general, and why it works that way. In Appendix A you can find usage samples of the software plus some technical details. Our system has a key advantage that has to be mentioned. Because of it's nature (accepts XML files as input, and a database that holds the Information Content (IC) for each concept, as this is computed by [43]), our system can be used as-is from anyone who wants to calculate Semantic Similarity between two terms based on ANY ontology that can produce XML files that can be validated by the XML-Schema that describes the WordNet ontology, without information loss (of course for the IC similarity measures, IC values for each concept of an ontology must be computed first respectively). For example as described in section A.4 in Appendix A, to compare two terms from the *MeSH* ontology, the system can produce accurate results if the XML files describing the MeSH-terms can be validated by the XML-Schema that describes the WordNet ontology. Furthermore, *it's plugable architecture*, allowing for expansion with minimum effort. One can write a new similarity measure in Java and just plug the Class produced in the system. The use of the new similarity measure is now done by just selecting it. The system consists of four main parts (see Figure 3.4

Ontology: The MeSH ontology (database). We need to have the ontology, actually the derived MeSH database, available in order to be able to extract information for each term we want to compare. For example, the service (see section A.5.3 in page 66) that creates the XML files, wouldn't work without the ontology, neither we could calculate the Information Content (IC) value for each concept (term).

XML repository: The second part is the repository that holds the XML files for all MeSH terms. Each file holds all the information extracted from the ontology associated with a term, (ie. dementia.xml fully describes the term "dementia", in hierarchical structured way, with hyponyms ,hypernyms etc). Each XML file contains information for all the senses of a term.

Information Content (IC) database: The third part is a database (postgresql database [2]) that holds the IC of each term as this is calculated by Equation 3.1. The database is accessed every time we need to retrieve an IC value of a term. The database is relational and all information content values are stored in tables according to the number of the subtrees respectively . Each table corresponding to a subtree of MeSH has two columns: the first one contains the Concept as as string and the second hold the IC value of the Concept.

Base System: The machine that calculates the similarity between two given terms. All similarity measures are implemented here. Given two XML files and a similarity measure, the system calculates the similarity between them, returning the result to the user, either by command line or a web interface. User has the option to compare a specific sense of a term with a specific sense of another term, or to compare by their *Most Common Sense* (MCS) or by all the senses of the compared terms.

To summarize, the user has the following options

- *Sense selection* The user has the option to select which sense of the term to compare (MCS or all senses).
- *Similarity method selection* A list of 10 similarity measures is available from all categories.
- *Ontology selection* Our system is Ontology independent in a way that described in the beginning of this section. The methods can be used within any ontology that conforms to Wordnet's specific schema or that can be mapped to this schema
- *Cross ontology similarity* by selecting two terms from different ontologies and compare them.

A complete API was developed in order to be used from anyone who wants to make use of the system functionality.
A complete working interface of our system, having all the above characteristics, can be found at the web at **http://www.ece.tuc.gr/mesh_similarity**. Additive to the above characteristics, the web user can find information about all similarity methods and the ontologies used.

[2]http://www.postgresql.org

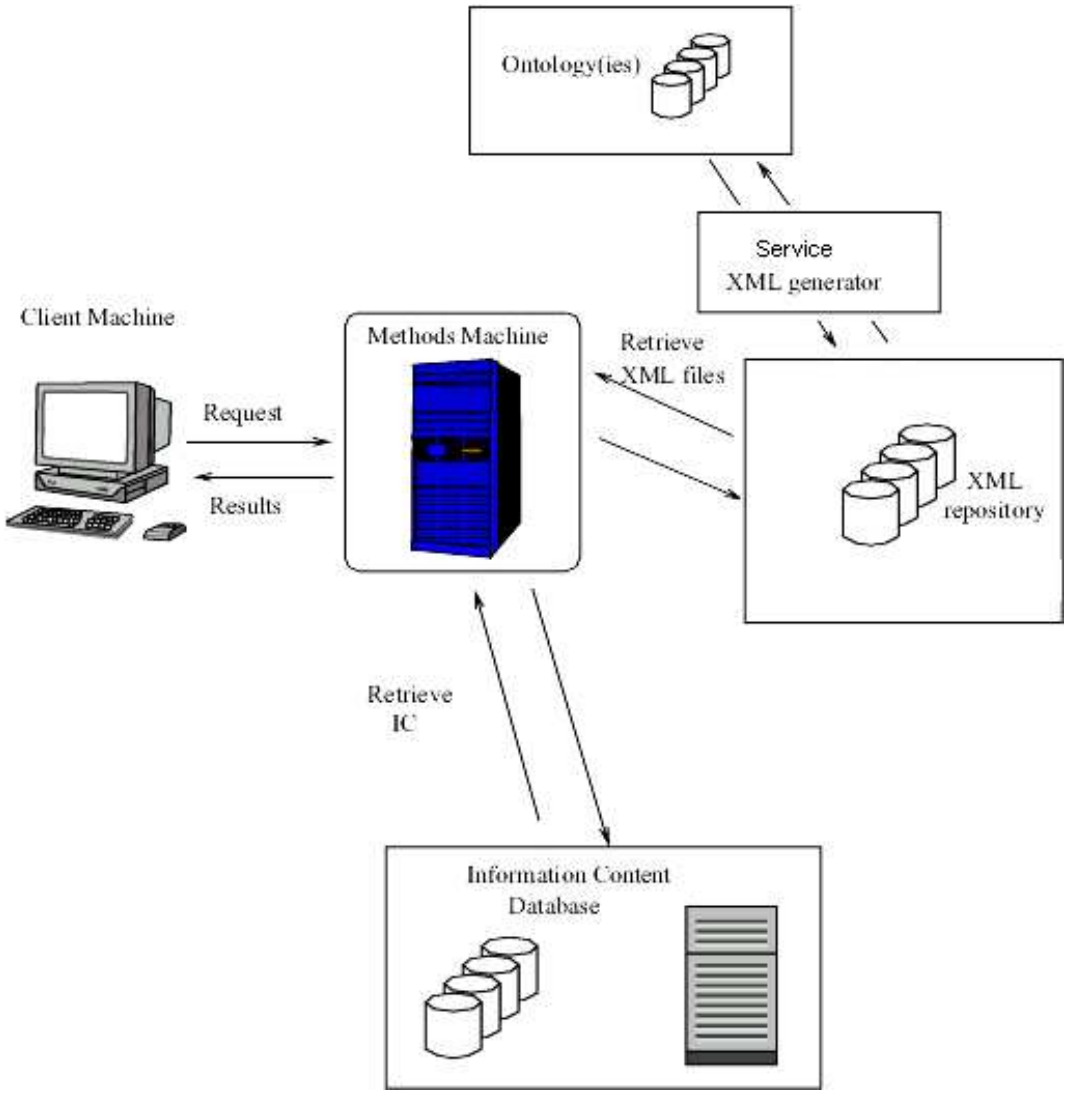

Figure 3.4: Architecture of the implemented Semantic Similarity System

Samples and images of the interface can be found in the Appendix at section A.10 at page 76.

Chapter 4

Information retrieval application on Medline

We introduced and implemented some similarity measures based on the MeSH ontology. In this chapter we describe an application were similarity measures can be used. An information retrieval application on Medline database, since similarity measures where applied upon MeSH ontology. We propose a similarity model for document retrieval on Medline rather than the standard Vector Space Model, based on a similarity measure. Finally, we evaluated the performance based on the results obtained by intergrading the proposed model in the application, by precision and recall diagrams.

4.1 Medline

MedLine Database is a metadata collection referred to biomedical articles. Metadata collections are sets of documents with additional information about the document, information on the organization of the data, the various data domains, and the relations between them. Publications in the MEDLINE database are manually indexed by NLM using MeSH terms, with typically 10-12 descriptors assigned to each publication. Hence, the MeSH annotation defines for each publication a highly descriptive set of features. Of the over 7 million MEDLINE publications that contain abstracts, more than 96% are currently indexed [10]. The articles stored in MedLine have both Descriptive and Semantic Metadata. So , MedLines documents have more information than the simple article reference. Figure 4.1 shows the structure of a MedLines document.

The most important difference in a MedLine document is the MH field that gives us the meaning (Semantic Data) of an article. We used **TI**, **AB** and **MH** fields to find relevant information for a query in our application.

PMID:	PubMed Identifier
UID	Unique Identifier
TI	The article's tittle
AU:	The article's authors
LA	Language of publication
MH	MeSH Term related
PT	Publication type
DA	Date of acceptance
DP	Date of publication
AB	Abstract
SO	Source of publication

Figure 4.1: Medline document structure

4.2 Proposed method: Similarity Matrix Model

In information retrieval (IR) applications the state of the art model for extracting relevant information, or documents in our case, is the vector space model [41] (VSM). The problem in IR is the relevance of the extracted information to the query, and VSM manage this quite well. However, it considers that every term is independent, and this is not true. For example, let's say that a user query has the term "ache" instead of "pain". Although the two terms are synonyms, VSM considers them independent and only the first one will taken into account. The similarity aspect between terms that we introduce to our proposed model tries to overcome this independence and score documents by the concept that a query may implies.

The similarity between two documents d_i and d_j (we consider the query as a document) is computed according to (VSM) as the cosine of the inner product between their term vectors

$$Sim(d_i, d_j) = \frac{\sum w_{it} w_{js}}{\sqrt{\sum_t w_{it}^2}\sqrt{\sum_s w_{js}^2}} \tag{4.1}$$

where w_{it} and w_{js} are the weights in the two vector representations. Given a query, all documents are ranked according to their similarity with the query.

The lack of common terms in two documents does not necessarily mean the documents are irrelevant. Semantically similar terms or concepts may be expressed in different ways in documents. For example, VSM will not recognize synonyms or semantically similar terms (e.g., pain - ache). The proposed model can achieve more than synonymy relation with the help of the MeSH ontology.

The proposed model is based on a semantic similarity measure by relating MeSH terms. The similarity measure of Li et al. [22] has been shown particularly effective, and was selected for our model implementation. The proposed approach works in three steps:

Re-weighting: The re-weighting of the query terms is done in the following way:

The weight of a term is adjusted due to its relation with other semantically similar terms within the same vector as follows

$$w_i = w_i + \sum_{\substack{j \neq i \\ sim(i,j) \geq t}} w_j sim(i,j) \tag{4.2}$$

This step help us to understand if the user is trying to emphasize in one specific area of interest by finding semantically similar terms in the vector. The re-weighting scheme will only affect the results if there are more than 2 terms in the query Vector and their similarity value is greater than threshold T. We selected a threshold t=0.8 for this study. This formula suggests assigning higher weights to semantically similar terms within the query. The weights of non-similar terms remain unchanged. The similarity between terms in the query is computed by the measure of Li et al. Original term weights in query were assigned value 1.

Expansion: Term that are related with query terms, are quite close in MeSH taxonomy (see Figure 4.2). This fact can be used to retrieve information related to a query term.

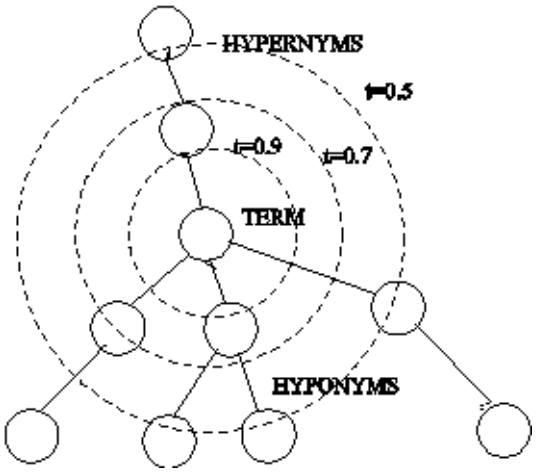

Figure 4.2: Term neighborhood in MeSH taxonomy

The expansion procedure is done with the following way. We augment the original query vector with hyponyms and hypernyms for every query term that belongs to the MeSH ontology.For each term r in the vector augment the vector

by its hypernyms and hyponyms s with $sim_(r, s) \geq t$ (e.g., T=0.8). By this expansion, each *new* term s in the vector (which now included new terms) is assigned a weight as follows

$$w_s^{'} = w_s + \sum_{sim(r,s) \geq T \ and \ r \in Q} \frac{1}{n} sim(r, s) \tag{4.3}$$

where r is the original term (which is expanded), n is the number of hyponyms of each expanded term s, w_s is the weight of term s before expansion and Q is the subset of the set of original query terms that led into new terms added to the expanded query. The above formula suggests taking the weights prior to expansion into account. It also suggests that the contribution of the original term r is normalized by the number n of its hyponyms. For hypernyms, $n = 1$.

We propose the expansion to be done with terms that are semantically similar to the query term in a radius T. Each term is expanded with terms with similarity values $\geq T$. As the value of T decreases, the radius increases. For a given T, the numbers of terms augmented, depend on the term position in the MeSH taxonomy. Very specific terms are expected to be very similar with their direct hypernyms and hyponyms. Very general terms (higher in the hierarchy) are expected to be less similar with their hypernyms and hyponyms. Moreover, if a big radius is chosen then the retrieved documents might not be focused on the initially specified topic (topic drift). We assume that higher thresholds will not retrieve documents diffused with query specified topic. Therefore special attention must be given in threshold, with respect to the similarity measure for expanding the terms in the query vector.

The expansion is done only for the *Most Common Sense* (MCS) as this is defined from the Ontology and not for all the senses of each term. This is not exactly intuitively correct as we don't know what sense of the term the user means, but we assume for the purpose of this study that the most frequent sense is desired.

Similarity Matrix: Expanding and re-weighting is fast for queries (queries are short in most cases specifying only a few terms) but not for document vectors with many terms. An approximation would be not to expand and re-weight the document vector. In this case, their similarity function must take into account relationships between semantically similar terms (something that the cosine similarity method cannot do). Then the similarity between an expanded and re-weighted query vector q and a non expanded and re-weighted document vector

d is computed as

$$Sim(q,d) = \frac{\sum_i \sum_j d_i q_j sim(i,j)}{\sum_i \sum_j d_i q_j} \tag{4.4}$$

where i and j are terms of the document and the query respectively. Also, d_i is the document i-term weight. Each term in the document is represented by its weight. The weight of a term is computed as a function of its frequency of occurrence in the document collection and can be defined in many different ways. The term frequency - inverse document frequency (tf * idf) model [42] is used for computing the weight. Typically, the weight d_i of a term i in a document is computed as $d_i = tf_i * idf_i$ where tf_i is the frequency of term i in the document and idf_i is the inverse frequency of i in the whole document collection. Weight q_j is the query j-term weight computed as mentioned before. The document similarity score is actually the enumerator of the above equation. The similarity $Sim(q,d)$ is normalized in the range[0,1].

A MeSH term is often consisted of two or more words. For example "abdominal pain" is a MeSH term. It is consisted of the words "abdominal" and "pain". An issue that needs special attention here is how can mesh terms be extracted, from query and document, in order to be compared within the Similarity matrix. We found a simple approach of a *sequential* process. We check if a word combined with the next one that come across in the text (of query or document) consists a MeSH term. If they do, then we check both of them with the next one if they consist a MeSH term, and so on. If they do not, then a) if a MeSH term was found until then, we keep the term and continue checking words after this term, b) if if a mesh term was not found until then, then we keep the word as is and continue checking with the others. For example,

"Abdominal pain in children"

→ stopwords to remove: in
check: abdominal? NO
check: abdominal pain? YES
check: abdominal pain children? NO (END of text)
→ found MeSH term? YES (keep term)
→ continue checking after MeSH term
check: children? NO (END of text)

→ found MeSH term? NO (keep word)
checked text: “abdominal_pain children”

After this serial processing MeSH terms CAN be found in text in order to be compared within the Similarity Matrix, otherwise MeSH terms consisted of one word could ONLY be compared.

4.3 MedSearch: Semantic IR Application on Medline

We have created an online searching application, a retrieval system of documents from Medline database. Medline documents were indexed by title (TI), abstract (AB) and MeSH headinds-terms (MH). Our proposed model was implemented upon this system, in order to evaluate the performance of the semantic similarity notion that we endorse. The MeSH ontology is used in order to calculate the semantic similarity between the Query and the Document term vectors, according to Equation 4.4 with Li et al. [22]. This demo is available at **http://www.ece.tuc.gr/medSearch** for the time being.

There are four fields in our application interface: a field for specifying users query, a search method drop down menu, a threshold (for expansion) selection for our proposed model and a desired results per page field. The available options for the method that a user can use for IR are:

Vector Space Model: This option uses the standard Vector Space Model formula in order to rank the results.

Vector Space Model with Query Expansion: The original query is expanded by Entry terms if any and the ranking is done using again Vector Space Model.

Similarity Matrix Model: With this option, the query is expanded (4.3), re-weighted (4.2) and then matched with the similarity formula (4.4) in order to rank the results. In this study Li et al. (2.4) similarity measure was used within the similarity formula as the referenced similarity measure.

All methods were implemented on top of Lucene (see A.11.1). Medline documents were indexed by title, abstract and MeSH terms (MeSH Headings) fields. These descriptions were syntactically analyzed and reduced into separate vectors of MeSH terms which were matched against the queries according to 4.4 (as similarity between expanded and re-weighted vectors). The weights of all MeSH terms were initialized to 1 while the weights of titles and abstracts were initialized by tf idf. The similarity between a query and a document is computed as

$$Sim(q,d) = Sim(q, d_{MeSHterms}) + Sim(q, d_{title}) + Sim(q, d_{abstract}) \quad (4.5)$$

This formula suggests that a document is similar to a query if most of its components are similar to the query.

Some detailed information about the MedSearch platform can be found in section A.11 in page 79.

4.4 Results: Precision/Recall and Evaluation

The experiment for the evaluation of the MedSearch application was conducted upon a set of 15 medical queries (see section A.8 on page 75, prepared by Dr.Quifen Qi. Each one of the 15 queries was specified between 3 to 5 words and we retrieved the best 20 answers. The results were also evaluated by Dr.Qiufen Qi. The methods we evaluated were:

Vector Space Model (*VSM*)
1.Vector Space Model with Query Expansion by Entry terms (*VSMEXP*)
2.Similarity Matrix with threshold t=1 for the query expansion (*SMM*)
3.Similarity Matrix with threshold t=0.9 for the query expansion (*SMM09*)
4.Similarity Matrix with threshold t=0.8 for the query expansion (*SMM08*)

Each method is represented by a precision/recall curve. Each point on a curve is the average precision and recall over all queries. As mentioned there was 20 answers for each query, so the precision/recall plot of each method contains exactly 20 points representing the average precision and recall over the 15 queries. Precision and recall values are computed from each answer set after each answer. The top-left point of a precision/ recall curve corresponds to the precision/recall values for the best answer or best match while, the bottom right point corresponds to the precision/recall values for the entire answer set. A method is better than another if it achieves better precision and recall. Due to the large size of the data set, it is practically impossible to compare every query with each document. To compute recall, for each query, the answers obtained by all candidate methods are merged and this set is considered to contain the total number of correct answers. This is a valid sampling method known as pooling method [47]. This method allows for relative judgements (e.g., method A retrieves 10% more relevant answers than method B) but does not allow for absolute judgements (e.g., method A retrieved 10% of the total relevant answers).

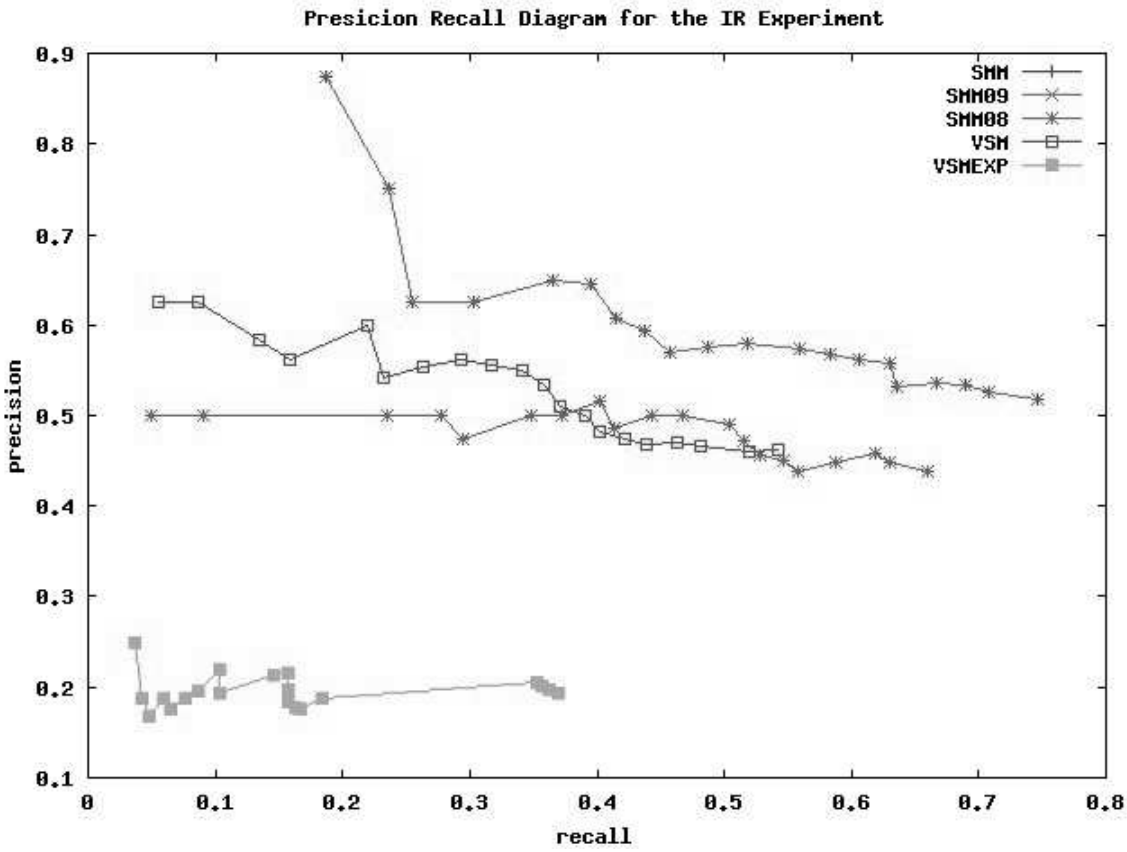

Figure 4.3: Precision/recall ALL methods

Fisrt of all VSM has a significant advantage in time performance when compared with Similarity SMM. In figure 4.3 method *VSMEXP* show having a very bad performance, even from VSM. This is because entry terms, as we mentioned earlier are not all synonyms, but we admitted that they are. Moreover, entry terms are too many for a MeSH term and expansion by all of them may diffuse the topic specified by the query. An example of the term "pain" entry terms is showed in section A.9 on page 76. Although SMM08 is much more effective than VSMEXP, meaning that expansion was done by neighbor terms not with entry terms, it seems that expansion generally diffuses the topic of the original query. We think that this an intrinsic problem of the MeSH ontology. MeSH ontology was created for clustering reasons on medical documents. It has a categorizing structure over terms rather than a semantic one, like Wordnet is. There is no expansion for SMM09 that is why its curve overlaps with the one of the SMM.

Finally, figure 4.4 shows that semantic information retrieval by SMM is more effective than classic information retrieval by VSM achieving up to 20 % better precision

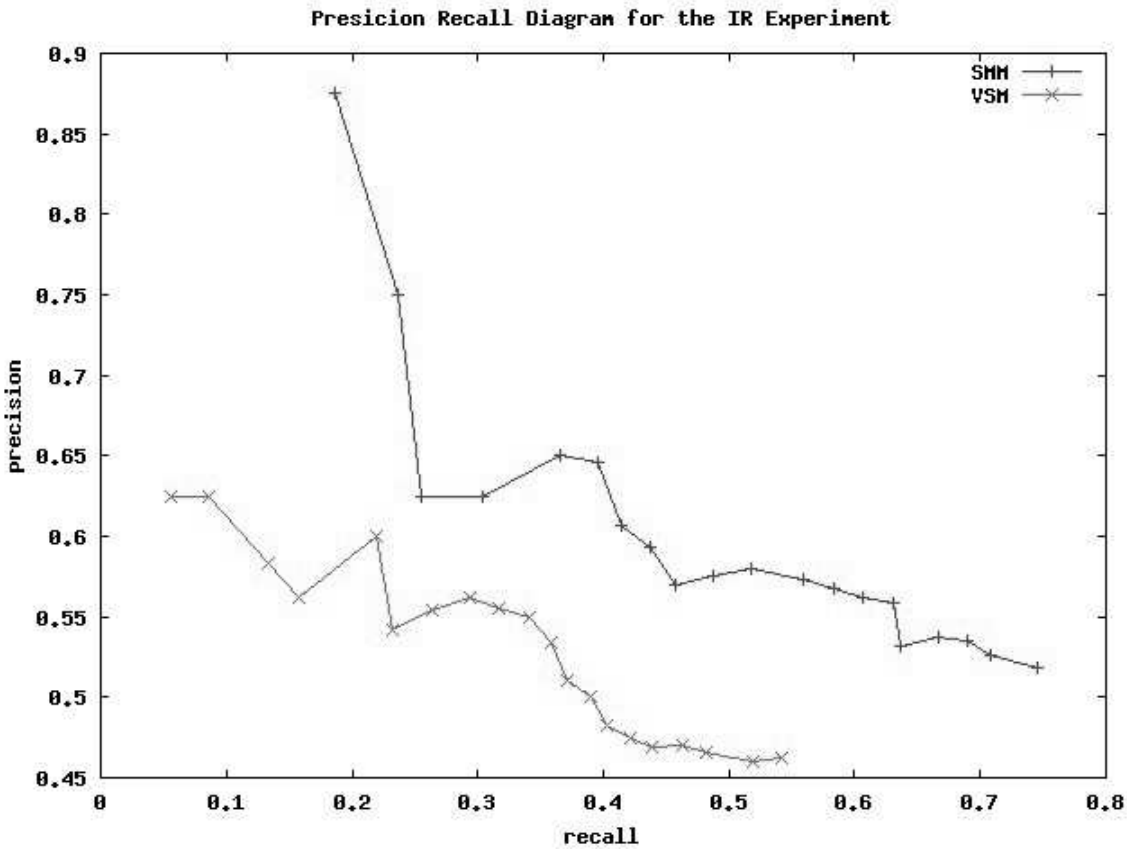

Figure 4.4: Precision/recall for SMM and VSM only

and recall. The efficiency of SMM is mostly due to the contribution of non-identical but semantically similar terms (as well as the non-semantically similar terms).

Chapter 5

Conclusions

Several semantic similarity measures for computing the conceptual similarity between MeSH terms (as well as in Wordnet in co-operation) were examined and implemented. The experimental results indicate that it is possible for these measures to approximate algorithmically the human notion of similarity reaching correlation up to 74% for the MeSH ontology. Based on this observation, we demonstrated that it is possible to exploit this information (as embedded in taxonomic ontologies and captured algorithmically by semantic similarity methods) for improving the performance of retrievals in applications such as the Web medical information systems. For this purpose, the Similarity Matrix Model (SMM), a novel document retrieval model that incorporates conceptual similarity into its retrieval mechanism is proposed and evaluated as part of this study. SMM can work in conjunction with any taxonomic ontology (e.g., application specific ontologies). The evaluation demonstrated very promising performance improvements over the Vector Space Model (VSM) as well, the classic document retrieval method. All methods are available on the Web.

Future work includes experimentation with more ontologies and experimentation with more application domains (e.g., document clustering, document searching in P2P systems). SMM can also be extended to work with more term relationships (in addition to the Is-A relationships) and with terms and term relationships not existing in an ontology (e.g., obtained from a thesaurus), or with co-occurrent terms. Moreover, SMM makes the assumption that the user is searching for the most common sense of the entered query term. Sense Disambiguation would help the model to understand which sense of the entered query term a user is searching for, while for medical concepts this is quite difficult to achieve. Also, more elaborate query expansion methods (e.g., methods for specifying thresholds for query expansion, when and where a term should be expanded) need to be investigated.

Bibliography

[1] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle, B. Amann, I. Fundulaki, M. Scholl, and A.-M. Vercoustre. Managing RDF Metadata for Community Webs. In *Proceedings of the ER'00 2nd International Workshop on the World Wide Web and Conceptual Modeling (WCM'00)*, pages 140–151, Salt Lake City, Utah, 9-12 October 2000.

[2] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web primer*. 2004.

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman, 1999.

[4] R. Beckwith and G. A. Miller. Implementing a Lexical Network. Technical Report 43, Princeton University, 1992.

[5] J.A. Blake and M. Harris. The Gene Ontology Project: Structured vocabularies for molecular biology and their application to genome and expression analysis. In A.D. Baxevanis, D.B. Davison, R. Page, G. Stormo, and L. Stein, editors, *Current Protocols in Bioinformatics*. Wiley and Sons, Inc., New York, 2003.

[6] H. Bulskov, R. Knappe, and T. Andreasen. On Measuring Similarity for Conceptual Querying. In T. Andreasen, A. Motro, H. Christiansen, and H.L. Larsen, editors, *Proceedings of the 5th International Conference on Flexible Query Answering Systems (FQAS'02)*, volume 2522 of *LNAI*, pages 100–111, Copenhagen, Denmark, 27-29 October 2002.

[7] Alexander Butanisky and Graeme Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In ..., 1999.

[8] P.R. Cohen and R. Kjeldsen. Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Information Processing and Management*, 23(4):255–268, 1987.

[9] Intalio Inc. and Contributors ExoLab Group. Using Castor XML. online documentation, 2005.

[10] Jorma Boberg Jouni Jrvinen Tapio Salakoski Filip Ginter, Sampo Pyysalo. Ontology-Based Feature Transformations: A Data-Driven Approach. *Lecture Notes in Computer Science*, 3230:279–290, Oct 2004.

[11] Jean Paoli C. M. Sperberg-McQueen Eve Maler Francois Yergeau, Tim Bray. *Extensible Markup Language (XML) 1.0 (Third Edition)*. W3C Recommendation, 2004.

[12] D.P. Hill, J.A. Blake, J.E. Richardson, and M. Ringwald. Extension and Integration of the Gene Ontology (GO): Combining GO vocabularies with external vocabularies. *Genome Res*, 12:1982–1991, 2002.

[13] Graeme Hirst and David St-Onge. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In *Proceedings of Fellbaum*, pages 305–332, 1998.

[14] I. Horrocks, P.F. Patel-Scheiner, and F. van Harmelen. Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic Web. In *Proceedings of 18th National Conference on Artificial Intelligence (AAAI'02)*, 2002.

[15] Ian Horrocks. DAML+OIL: a Reasonable Web Ontology Language. In *Proceedings of the International Conference on Extending DataBase Technology*, March 2002.

[16] J.J. Jiang and D.W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistic*, Taiwan, 1998.

[17] Lalana Kagal, Grit Denker, Tim Finin, Massimo Paolucci, Naveen Srinivasan, and Katia Sycara. An Approach to Confidentiality and Integrity for OWL-S. In *Proceedings of the 1st International Semantic Web Services Symposium (ISWSS'04)*, AAAI'04 Spring Symposium Series, 22-24 March 2004.

[18] G. Karvounarakis, V. Christophides, D. Plexousakis, and S. Alexaki. Querying RDF Descriptions for Community Web Portals. In *Proceedings of the 17iémes Journees Bases de Donnees Avancees (BDA'01)*, pages 133–144, Agadir, Maroc, 29 October - 2 November 2001.

[19] R. Knappe, H. Bulskov, and T. Andreasen. On Similarity Measures for Content-Based Querying. In O. Kaynak, editor, *Proceedings of the 10th International Fuzzy Systems Association World Congress (IFSA'03)*, pages 400–403, Instsnbul, Turkey, 29 June - 2 July 2003.

[20] K. Knight and S. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of thr National Conference on Artificial Intelligence (AAAI'94)*, Seattle, WA, 1994.

[21] Claudia Leacock and Martin Chodorow. *Filling in a sparse training space for word sense identification. ms.* March 1994.

[22] Yuhua Li, Zuhair A. Bandar, and David McLean. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE*

Transactions on Knowledge and Data Engineering, 15(4):871–882, July/August 2003.

[23] D. Lin. Principle-Based Parsing Without Overgeneration. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL'93)*, pages 112–120, Columbus, Ohio, 1993.

[24] P.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Investigating Semantic Similarity Measures across the Gene Ontology: the Relationship between Sequence and Annotation. *Bioinformatics*, 19(10):1275–83, 2003.

[25] A. Magkanaraki, S. Alexaki, V. Christophides, and D. Plexousakis. Benchmarking RDF Schemas for the Semantic Web. In *Proceedings of the 1st International Semantic Web Conference (ISWC'02)*, Sardinia, Italy, 9-12 June 2002.

[26] D.L. McGuinness. Conceptual Modeling for Distributed Ontology Environments. In *Proceedings of the 8th International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS'00)*, Darmstadt, Germany, 14-18 August 2000.

[27] D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, Breckenridge, Colorado, USA, 12-15 April 2000.

[28] D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The Chimaera Ontology Environment. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00)*, Austin, Texas, 30 July - 3 August 2000.

[29] G. A. Miller, R. Bechwith, C. Felbaum, D. Gross, and K. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[30] S.J. Nelson, D. Johnston, and B.L. Humphreys. Relationships in Medical Subject Headings. In C.A. Bean and R. Green, editors, *Relationships in the Organization of Knowledge*, pages 171–184. Kluwer Academic Publishers, New York, 2001.

[31] S.J. Nelson, T. Powell, and B.L. Humphreys. The Unified Medical Language System (UMLS) Project. In A. Kent and C.M. Hall, editors, *Encyclopedia of Library and Information Science*, pages 369–378. Marcel Dekker, Inc., New York, 2002.

[32] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[33] T. O'Hara, N. Salay, M. Witbrock, D. Schneider, B. Aldag, S. Bertolo, K. Panton, F. Lehmann, and et al. Inducing Criteria for Mass Noun Lexical Mappings

using the Cyc KB, and its Extension to WordNet. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands, 15-17 January 2003.

[34] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, January/February 1989.

[35] S. Reed and D. Lenat. Mapping Ontologies into Cyc. In *Proceedings of the AAAI'02 Conference Workshop on Ontologies For The Semantic Web*, Edmonton, Canada, July 2002.

[36] O. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity and Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[37] R. Richardson, A. Smeaton, and J. Murphy. Using WordNet as a Knowledge Base for Measuring Semantic Similarity Between Words. Technical Report Working paper CA-1294, School of Computer Applications, Dublin City University, Dublin, Ireland, 1994.

[38] M.A. Rodriguez and M.J. Egenhofer. Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Tramsactions on Knowledge and Data Engineering*, 15(2):442–456, March/April 2003.

[39] M. Sabou, D. Richards, and S. van Splunter. An Experience Report on using DAML-S. In *Proceedings of the 12th International World Wide Web Conference Workshop on E-Services and the Semantic Web (ESSW'03)*, Budapest, 2003.

[40] G. Salton and C. Buckley. On the Use of Spreading Activation Methods in Automatic Information. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 147–160. ACM Press, 1988.

[41] G. Salton and M. McGill. Introduction to Modern Information Retrieval. A vector space model for automatic indexing. *Communications of the ACM*, 11:613–620, 1983.

[42] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[43] Nuno Seco, Tony Veale, and Jer Hayes. An intrinsic information content metric for semantic similarity in wordnet. 2004.

[44] W. Douglas Johnston Stuart J. Nelson and Betsy L. Humphreys. Relationships in Medical Subject Headings (MeSH). In *National Library of Medicine, Bethesda, MD, USA*, 2002.

[45] R. Studer. Knowledge Engineering and Agent Technology. In J. Cuena and et al., editors, *Situation and Perspective of Knowledge Engineering*. IOS Press, Amsterdam, 2000.

[46] A. Tversky. Features of Similarity. *Psycological Review*, 84(4):327–352, 1977.

[47] E.M. Voorhees and D.K. Harmann. Overview of the Seventh Text REtrieval Conference (TREC-7). In *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7)*, pages 1–23, http://trec.nist.gov/pubs/trec7/t7 proceedings.html, 1998.

[48] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (ACL'94)*, pages 133–138, Las Cruces, New Mexico, 1994.

Appendix A

Implementation and Experimental Design

A.1 Tools

This study was generally implemented in Java[tm]. In the process several tools were used for the final result and many concepts in several parts of this study mentioned and must be palpable. First of all we shall introduce a main concept, the Extensible Markup Language (XML) and its grammar. The reasons why XML is used in this study was

1. **MeSH Ontology** is in XML format and the information contained had to be extracted from this representation to a database.

2. **Similarity measures** needed a uniform input that will contain information of properties, attributes, and relations of a term in order to compare it with others,and XML is the proper way to represent that kind of data as it will be presented below.

Moreover, the purpose of use of some tools for this study, like the Castor XML and JDO API and Lucene, must be understood. So, In the next sections there is an introduction to XML, explaining along why XML is appropriate for this study, and a brief presentation of the tools.

A.2 Introduction to XML

Extensible Markup Language, abbreviated XML[1], describes a class of data objects called XML documents [11]. XML like HTML (standard language where web pages are written) is based on *tags*. XML like HTML are markup languages as well, because they allow one to write some content and provide information about what role that content plays. However, all tags in XML must be closed (for example, for the tag

[1] http://www.w3.org/XML/

<example> must be a closing tag </example>) while in HTML some tags may be left open such as
. An HTML document does not contain structural information about pieces of the document and their relationships, though in an XML document every piece of information is described and their relations are defined through the nesting structure. For example a nested <year> tag appears within a <date> tag, so the nested one describes in a way the properties of the second.

```
<date>
    <year>2005</year>
    <month>May</month>
    <day>23</day>
</date>
```

Moreover, in a XML document the user may use information in various ways, define a vocabulary himself, though in a HTML document the used tags are *predefined*, because in HTML, representations are intended to display information, so the set of tags are fixed like lists bold, color and so on. Therefore, XML is a *metalanguage for markup: it does not have a fixed set of tags but allows users to define tags of their own* [2]. Moreover XML can be used as a uniform data exchange format between applications as it is used nowadays more than its originally intended use as a document markup language.

In the next section A.2.1 XML langauge is described in more detail, while the structuring of the XML documents is described in section A.2.2. The structure of XML documents must be defined either by writing a DTD (Document Data Definition) or by writing an XML Schema[2]. The later will gradually replace the DTDs.

A.2.1 The XML Language

The beginning of an XML document is consisted of an XML declaration and an optional reference to external structuring documents. For example

```
<?xml version="1.0" encoding="UTF-8"?>
```

The above declaration says that the document is an XML document, defines the version na the character encoding used. A reference to external structuring documents is like the below line

[2] http://www.w3.org/XML/Schema

<!DOCTYPE MeSH SYSTEM "MeSH.dtd">

where it says that the structuring document is a DTD file (see section A.2.2 on page 56) called MeSH.dtd. Of course the reference may be a URL. To denote a local name and a URL then instead of the SYSTEM label a PUBLIC label must be used.

XML documents is consisted of *elements* as well, which they represent the main concept of the document, the "things" that it talks about. An element is denoted inside tags, for example

<descriptorName>Sickle Cell Anemia</descriptorName>

A context of an element may be text value like the example above, or another element.

<descriptorRecord>
<descriptorName>Sickle Cell Anemia</descriptorName>
</descriptorRecord>

Of course there can be an *empty* element when there is no content like

<ECIN></ECIN> abbreviated as <ECIN/>

However an empty element can be of use because it may contain properties, called *attributes* like

<descriptorRecord name="Sickle Cell Anemia"/>

When to use nested elements instead of attributes is usually a matter of taste.

Finally before we go through the structuring of an XML document lets see some general syntactic rules.

- There is only one top element, the *root* element.
- Every *non empty* element has an opening and a closing tag, respectively.
- Overlap of tags is not allowed for example

  <descriptorRecord><descriptorName>Cancer</descriptorRecord></descriptorName>

- Attribute names are unique

A.2.2 Structuring of XML

In an XML document all the element and attributes names that may be used must be defined as well as the structure of the XML. For example what elements contain other elements what attributes those element have, their values and so on. This structuring information is presented and defined quite always in other documents, and if a XML document respects and uses this structuring information we say that it is *valid*. These other documents that defines the structure of another XML document are called DTDs (more restricted structure) and XML Schema (more extended definitions mainly in data types).

- **DTD** (Document Data Definition) components can either defined within an XML document(internal DTD) or in a seperate file(external DTD), which the later is better because their definitions can be used in several other XML documents. Now lets see an example of how a DTD looks like. Consider the following XML element

  <descriptorRecord>
  <term RecordPreferredTermYN="Y">Fever</term>
  <entryTerm>Hyperthermia</entryTerm>
  <treeNumber>C23.888.119.344</treeNumber>
  < /descriptorRecord>

  A DTD for the above element looks like this

  <!ELEMENT descriptorRecord (term, entryTerm, treeNumber)>
  <!ELEMENT term (#PCDATA)>
  <!ATTLIST term RecordPreferredTermYN (Y — N) #REQUIRED>
  <!ELEMENT entryTerm (#PCDATA)>
  <!ELEMENT treeNumber (#PCDATA)>

  The above DTD defines a *root* element, the descriptorRecord element which contains three other elements types term, entryTerm, and treeNumber. Only those elements can be used in the XML document. Also it defines that all elements except the root element may have any content (#PCDATA) the only atomic type for elements. The term element may also have an attribute of enumeration type RecordPreferredTermYN with possible values Y or N. Other predefined attribute data types that may used are

  - *CDATA* which is a sequence of characters (string).
  - *ID* which is a unique name across the entire XML document.

  - *IDREF* which is a reference to an *ID* attribute data type, where the *ID* attribute has the same value as the *IDREF* attribute.
  - *IDREFS* a sequence of *IDREF*s.

  and their values can be of type

  - *#REQUIRED*, where the attribute must always appear in every occurrence of the element.
  - *#IMPLIED* where the attribute appearance is optional.
  - *#FIXED "value"* where the attribute must always appear in every occurrence of the element with the specified value and
  - *value* where it specifies the default value of an element's attribute.

  In conclusion of this brief description of a DTD file component definitions, a element within another element may have *cardinality operators* of type: **?** if the element appears zero times or once, ***** if the element appears or more and + if it appears one or more times (a list), for example

  <!ELEMENT descriptorRecord (term, entryTerm, treeNumber+)>

  means that the element descriptorRecord may have one or more treeNumber elements.

- **XML Schema** is a structure defining document like the DTD. However it has more capabilities and possibilities on defining elements plus the capability to build schemas from other schemas. Here is presented in short the kind of these capabilities.

  - cardinality constraints on specific number of elements appearance in a XML document (minimum and maximum occurrences)
  - data types can be numerical (integer, short, byte, long, float, decimal), string including the ones specified for attributes in DTD (ID, IDREF, CDATA, Language) and date—time data types (date, time, month, year). It can also include user-defined data types of a more complex definition where *sequence* is a sequence of existing data types in a predefined order, *all* is a collection of elements and *choice* is a collection of elements where one is selected for use.
  - defined data types can be extended with new elements (inheritance), as they can also be restricted by adding constraints on certain values.

  DTD and XML Schema structuring definitions in a more detailed way can be found in the web.

A.3 Castor XML and JDO

Castor is a multifaceted software tool being developed under the auspices of exolab.org, an informal organization involved in the development of open source, enterprise software projects based on Java[tm] and XML. It's the shortest path between Java objects, XML documents and relational tables. Castor provides Java-to-XML binding, Java-to-SQL persistence, and more. XML data binding is the binding of XML documents to (Java) objects designed especially for the data in those documents. The primary function of Castor is to perform data binding. In other words, data binding is a process that facilitates the representation of one data model in another. Other popular XML Data Binding frameworks are *JAXB* [3], *JaxMe* [4] etc.

Castor XML unlike the other two main XML APIs, DOM (Document Object Model) and SAX (Simple API for XML) which deal with the structure of an XML document, enables one to deal with the data defined in an XML document through an object model which represents that data. Castor XML can marshal almost any "bean-like" Java Object to and from XML. In most cases the marshalling framework uses a set of ClassDescriptors and FieldDescriptors to describe how an Object should be marshalled and unmarshalled from XML. XML Class descriptors provide the marshalling framework with the information it needs about a class in order to be marshalled to and from XML. For those not familiar with the terms "marshal" and "unmarshal", it's simply the act of converting a stream (sequence of bytes) of data to and from an Object. The act of "marshalling" consists of converting an Object to a stream, and "unmarshalling" from a stream to an Object [9]. Two main classes are consisted in Castor XML tool, org.exolab.castor.xml.Marshaller and org.exolab.castor.xml.Unmarshaller. The below figure A.1 shows the Castor XML "binding" framework.

Although it is possible to rely on Castor's default behavior to marshal and unmarshal Java objects into an XML document, it might be necessary to have more control over this behavior. For example, if a Java object model already exists, Castor XML Mapping can be used as a bridge between the XML document and that Java object model. Castor allows one to specify some of its marshalling/unmarshalling behavior using a mapping file. This file gives explicit information to Castor on how a given XML document and a given set of Java objects relate to each other. The mapping file describes for each object how each of its fields have to be mapped into XML. A field is an abstraction for a property of an object. It can correspond directly to a public class variable or indirectly to a property via some accessor methods (setters and getters). So, with the marshalling and unmarshalling functions plus the mapping file (optional) the "binding" of XML elements - Java objects is

[3] http://java.sun.com/xml/jaxb/
[4] http://ws.apache.org/jaxme/ (open source)

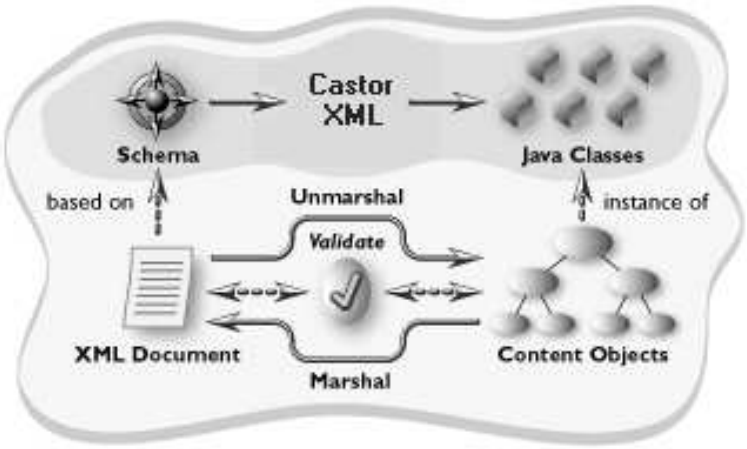

Figure A.1: How Castor XML works

rather easy. A detailed example is shown in section A.5.2 on page 63.

Castor JDO is an open source object-to-relational binding framework for Java. Using XML mapping, Castor JDO bridges Java objects with relational databases. Unlike Castor's XML marshalling – where there is a default mapping of a Java object to XML elements and attributes – no default mapping exists for binding Java objects to SQL database tables: you must use a mapping file to enable this functionality. The mapping file contains explicit information on how Castor should represent a set of Java objects in a relational database. In the mapping file (which is an XML file), each Java object is represented by a <`class`> element and each property in that object is represented by a <`field`> element. Additionally, each column from within a relational table is represented by an <`sql`> element. The idea of creating the mapping file is presented by example in section A.5.1.

The database is by default configured through a separate XML file which links to the mapping file. Besides, there are a few main classes that are consisted in Castor JDO framework. The org.exolab.castor.jdo.JDO, which defines the database name and properties and is used to open a database connection. The database configuration is loaded on demand by setting the configuration file URL with setConfiguration. Creating multiple JDO objects with the same configuration will only load the database configuration once. The org.exolab.castor.jdo.Database object represents an open connection to the database. There is little overhead involved in opening multiple Database objects, and a JDBC connection is acquired only per open transaction.

All JDO operations occur within the context of a transaction. JDO works by loading data from the database into an object in memory, allowing the application to modify the object, and then storing the object's new state when the transaction commits. All objects can be in one of two states: transient or persistent.

Transient Any object whose state will not be saved to the database when the transaction commits. Changes to transient objects will not be reflected in the database.

Persistent Any object whose state will be saved to the database when the transaction commits. Changes to persistent objects will be reflected in the database.

An object becomes persistent in one of two ways: it is the result of a query, (and the query is not performed in read-only mode) or it is added to the database using *create*(java.lang.Object) or *update*(java.lang.Object). All objects that are not persistent are transient. When the transaction commits or rolls back, all persistent objects become transient. In a client application, begin(), commit() and rollback() methods are used to in order to manage transactions. The method *create*(java.lang.Object) creates a new object in the database, or in JDO terminology makes a transient object persistent. An object created with the create method will remain in the database if the transaction commits; if the transaction rolls back the object will be removed from the database. So all we have to do is to open the MeSH database, to perform a transaction, *create* the objects unmarshalled from the XML document, commit the transaction and close finally the database. That is pretty much all.

A.4 Similarity measures on Wordnet and MeSH ontology

In cooperation with the MeSH ontology, we developed and implemented the similarity measures according to the idea of a web service that Bernard Bou has written for Wordnet [5] (version 2.0). This Java Web service produces XML output to word queries of the WordNet ontology-database. It can also represent the result of a word query in html data, in a tree rendering representation of the result or a hyperbolic rendering [6]. The XML output of a word query in Wordnet database is a suitable input for the implementation of the semantic similarity measures in MeSH ontology since the XML document DTD defines these relations like synonyms, hyponyms, hypernyms, definitions, and so on. So, we need to contain the above information in an XML document of a MeSH term, in order to compare semantic relations of one term with another with the measures described in this study. The same thing was implemented for the similarity measures in Wordnet. Therefore, a mapping between MeSH and Wordnet ontology must be provided (a mapping between the MeSH and Wordnet DTD elements to be accurate) so that we can have the desired input XML document of a MeSH term for the semantic similarity measures. This mapping is thoroughly expalined in section A.5.3 in page 66.

[5] see http://wnws.sourceforge.net/

[6] see http://jws-champo.ac-toulouse.fr:8080/wordnet/

A.5 MeSH to Database

MeSH terms contains information which is described in that large XML descriptors file. This file can not be easily handled. Meaning that an application that would query and retrieve specific information of a term within the XML file is quite time expensive and non practical. We needed the ontology within a database for convenience. This section presents the creation process of the *MeSH database*. The MeSH database was created in MS SQL Server, and there was no particular reason why this specific type of relational database was selected. We could have preferred other formats, and not necessary relational. To support this, other database formats were selected for different parts of this study, like the online evaluation of the term pairs for the experiment by users (information was stored in a mySQL database) or the function described in [43] for the calculation of the IC value of a MeSH term instead of a Wordnet concept by its hyponyms, independent of specific statistical corpora analysis (values were stored in a postgresql database).

A.5.1 Creating the MeSH Database

The MeSH database is consisted of tables, and each table has columns, as in every relational database. The MeSH descriptors XML file (desc2004.xml) has a specific structure for its components, which is described in the corresponding descriptors DTD file (desc2004.dtd). Every MeSH descriptors XML file has to follow this specific structure described in its corresponding DTD file otherwise it is not valid (it cannot be previewed). The entries of this DTD file are described thoroughly in previous section (section A.2.2 on page 56). The form of the tables and columns that will be created for the MeSH database is entirely depended on that MeSH descriptors DTD file. This is because WE wanted to store in the MeSH database all the information of the MeSH descriptors XML file, and select the part that suit us afterwards. There are relations and attributes of MeSH terms of no need for our purpose but we wanted the *MeSH Database* to be completed. The "no need" part will be cleared further down.

So the base guidelines for the formation of the MeSH database is the descriptors DTD file as shown in section A.2.2 on page 56. But lets see some examples of the creation process.

1. The example below shows the correspondence of an element of the DTD descriptors file into a simple MeSH database table.

```
<!ENTITY %normal.date "(Year, Month, Day)">
<!ELEMENT Day (#PCDATA)>
<!ELEMENT DateCreated (%normal.date;)>
<!ELEMENT Month (#PCDATA)
<!ELEMENT Year (#PCDATA)>
```

⇓

DateCreated
DateCreatedPK
Year
Month
Day

There are four element entries that participate in the formation of the above table. The *DateCreatedPK* attribute–column is created as the Primary Key of the **DateCreated** table, in order to describe every instance of it explicitly. The meaning of those few lines of the DTD file is as follows:

- The element types **DateCreated**, *Year*, *Day*, and *Month* can be used in the MeSH descriptors XML document.
- A **DateCreated** element contains a *Year* element, a *Month* and a *Day* element, in that order.
- A *Year* element, a *Month* element, and a *Day* element may have any content cause their value is #PCDATA, the only atomic type for elements in DTDs.

The last two observations are responsible for the formation of a table **DateCreated**(*Year*, *Month*, *Day*) for the MeSH database, because *Year*, *Month*, and *Day* elements are consisted of atomic type values(#PCDATA), though **DateCreated** element contains the aforementioned.

2. This more complicated example demonstrates the formation of two tables for the MeSH database.

<!ELEMENT **SemanticTypeList** (**SemanticType**+)>
<!ELEMENT **SemanticType** (*SemanticTypeUI*,
SemanticTypeName)>
<!ELEMENT *SemanticTypeUI* (#PCDATA)>
<!ELEMENT *SemanticTypeName* (#PCDATA)>

⇓

SemanticTypeList
SemanticTypeListPK

SemanticType
SemanticTypePK
SemanticTypeListFK
SemanticTypeUI
SemanticTypeName

There are two tables formatted from the above elements of the MeSH descriptors DTD file. The **SemanticTypeList** and the **SemanticType** table. The element **SemanticTypeList** contains the **SemanticType** element but with a '+' symbol at the end. This means that **SemanticType** element may appear one or more times inside the **SemanticTypeList** element. So, **SemanticType** same as the previous example contains *SemanticTypeUI* and *SemanticTypeName* (which they contain #PCDATA) elements and formats a table, with a Primary Key named *SemanticTypePK*. An attribute–column *SemanticTypeListFK* (Foreign Key) for the table **SemanticType** is also created in order to denote the corresponding *SemanticTypeListPK*—of the **SemanticTypeList** table—instance that belongs to.

The formation of the *MeSH Database* is done by the above basic examples of mapping from the descriptors DTD file elements to tables and columns. Now the MeSH descriptors DTD XML file that is validated by this DTD file can be accessed, and the information that it contains can 'easily' be extracted—with the help of Castor XML and JDO API—and inserted into the corresponding tables of the database. The 'easily' said part will be supposable in the next section where Castor XML API works in the same idea of the mapping process presented here.

A.5.2 Using the Castor XML and JDO API

The use of Castor (XML and JDO API) was described further in section A.3 on page 58. Compendiously the Castor XML tool deals with the data defined inside an XML document through an object (class) model that represents the data, though the Castor JDO tool provides an interaction model of the represented data with any type of database server. So, the purpose of use of the Castor tool for this study is the creation of a stable and completed database from the original MeSH XML file.

At the end of the previous section, we said that the Castor XML API *works in the same idea* as the the mapping process of the descriptors DTD file to database. Castor XML API generates java objects (classes) from the data of a certain XML document, given the corresponding XML Schema (see section A.2.2 on page 57). Creating a database based upon the descriptors DTD file, and creating java objects with Castor XML API based upon the same structure file (in XSD format though) makes the insertion of the data inside the MeSH XML file an *easy* process. The simplicity of the process is shown using the Castor JDO API while mapping these objects—created by the XML API of Castor—to the corresponding tables of the MeSH database. Before this however, lets see an example of how a java object is created by the Castor XML API given the MeSH descriptors XSD file (XML Schema).

example. The XSD descriptors file has elements like the DTD file. The elements of the XSD below are the same as the example 1 on page 61. **DateCreated** element which

is a complexType element, meaning that it contains other elements, and to be specific a sequence of elements *Year*, *Month* and *Day*. The elements in this sequence are string type elements as shown below.

```
<xs:element name="DateCreated">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="Year" />
         <xs:element ref="Month" />
         <xs:element ref="Day" />
      </xs:sequence>
   < /xs:complexType>
</xs:element>
<xs:element name="Year" type="xs:string" />
<xs:element name="Month" type="xs:string" />
<xs:element name="Day" type="xs:string" />
```

⇓

Castor XML Object	Modified Object for Castor JDO
DateCreated	**DateCreated Modified**
	private int DateCreatedPK
private String *Year*	private String *Year*
private String *Month*	private String *Month*
private String *Day*	private String *Day*
	public String getDateCreatedPK
public String getYear()	public String getYear()
public String getMonth()	public String getMonth()
public String getDay()	public String getDay()
...	...
	public void setDateCreatedPK(int pk)
public void setYear(String year)	public void setYear(String year)
public void setMonth(String month)	public void setMonth(String month)
public void setDay(String day)	public void setDay(String day)
...	...

In the above table the object at the left is the one created from the Castor XML API. The one at the right side of the figure is the same object from the XML API but modified

with an attribute and its functions—the Primary Key attribute. This modification takes place for accommodation purposes later for the use of JDO API, in order the mapping from the object to the corresponding table of the MeSH database to be simple and precise. The modification process is the key for the simplicity of use of the JDO API as you will see below.

As it was mentioned in section A.3 on page 58, JDO API is used for defining the database configuration (connection) and managing any transactions to the database. The formation of the MeSH database is done in the previous section. All that is needed now is the mapping process for the java Objects (classes) to sql database tables, the configuration for the database connection and a transaction to the MeSH database in order to insert the java objects created from the unmarshalling process of the XML MeSH document. The mapping procedure which provided in a mapping XML file is a *simple correspondence* of every java object that was created to the same database table name. For example:

```
<class name="DateCreated" identity="dateCreatedPK" key-generator="MAX">
    <map-to table="DateCreated" />
      <field name="dateCreatedPK" type="integer" required="true">
        <sql name="dateCreatedPK" />
      </field>
      <field name="year" type="String" required="true">
        <sql name="year" />
      </field>
      <field name="month" type="String" required="true">
        <sql name=month" />
      </field>
      <field name="day" type="String" required="true">
        <sql name="day" />
      </field>
</class>
```

The above element of the mapping XML file of Castor JDO, is the mapping procedure for the **DateCreated** Object. The correspondence is obvious. After the mapping XML file is created as the above example and the configuration for the database connection is set, the proper transaction shall take place, in order to insert each descriptor record and its components—from the descriptors XML file—to the database. Because this XML file was too large and the first element of if is the *DescriptorRecordSet* we split it in smaller (Descriptor Record *Sets*) XML files. For each of this file using the unmarshalling function of Castor XML API, every instance of the java Objects (descriptor records) created, are inserted in the MeSH database with the *create* function of the CastorJDO API. After all the smaller XML files are being processed the relational *MesH Database* is completed.

In section A.1 it was mentioned that we needed a proper *input* for each of the similarity measures. The *input* shall contain all the appropriate information of a term that a similarity measure requires. So, the next step is to create the *input* XML file for a term which will include all the information that a similarity algorithm need, by extracting it from the created MeSH database.

A.5.3 Creating the similarity measures "Input" XML file

As mentioned in section A.4 in page 60, the desired *input* XML file for the semantic similarity measures is a mapping procedure of MeSH schema elements to Wordnet elements and fields, as the Wordnet DTD will be used as a structure definition document of the input XML file. The mapping process was done only for the Wordnet DTD elements that we considered useful for the similarity measures and for the ones that could be mapped from MeSH DTD, considering the structure and the information described in it. Below are presented the main elements that can or should be mapped, from the similarity measures perspective.

```
<!ELEMENT word (#PCDATA | key | pos)*>
<!ELEMENT key (#PCDATA)>
<!ELEMENT pos (sense*)>
<!ATTLIST pos name CDATA #REQUIRED>
<!ELEMENT sense (synset, links)>
<!ATTLIST sense number CDATA #REQUIRED>
<!ELEMENT synset (item*, defn)>
<!ATTLIST synset number CDATA #IMPLIED>
<!ELEMENT item (#PCDATA)>
<!ELEMENT defn(#PCDATA)>
<!ELEMENT links (hypernym?, hyponym?)>
<!ELEMENT hypernym (synset+, hypernym*)>
<!ELEMENT hyponym (synset+, hyponym*)>
```

So, the mapping procedure, according to the element definitions in section 2.2 in page 9, is as following:

word element has a string value which is mapped to element's **TermReference** string value. This element is referred to the word that the XML document describes. In our case should denote the MeSH term. The *word* element may contains the *key* or pos elements.

key element has as well a string value which is the same as the element's **TermReference** string value.

pos element may contain zero or more times the *sense* element. This element is referred to the part of speech of the word. It also has an atrribute *name* of string value and in our case is always "noun". There are not any other parts of speech in MeSH descriptors, only terms.

sense element contains the *synset* and *links* elements. This element is referred to the sense of the term. How many different meanings the word has. A MeSH term does not has different meanings, but a MeSH term can appear in many locations in the MeSH taxonomy, independently the subtree the preferred term is located. It also has a string value *number*. This number is computed by the times the related term is located in the MeSH taxonomy.

synset element may contain zero or more times the *item* element. This element defines the synset (synonym set) that a word belongs. In MeSH ontology the corresponding element is **TermList** element which is referred to the list of terms that a concept may have. It also contains the *defn* element.

item element has a string value which is is mapped to element's **TermReference** string value. Synonym terms (synset) are the ones that belongs to the same **TermList**. These terms in the list are items of the related synset.

defn element has a string value which is mapped to element's **ScopeNote** string value. This element refers to the definition of a word. In MeSH ontology, concept, a synset in other words, has a definition, a meaning which is denoted by the **ScopeNote** element.

links element may contain *hypernym* or *hyponym* elements. There are other links in Wordnet that were of no use for the similarity measures or the mapping to MeSH elements could not be done.

hypernym element contains at least once a *synset* element. It may also contain a *hypernym* element. This element is referred to the parent node of a word in Wordent taxonomy. In the MeSH taxonomy the location of each term is denoted with tree numbers. So, the mapping is denoted by the tree numbers a term may have. For example in MeSH the term "Asthenopia" the parent term node is "Eye diseases". Term "Asthenopia" has a tree number C11.093. So the parent node must be C11 which is indeed the term "Eye diseases".

hyponym element contains at least once a *synset* element. It may also contain a *hyponym* element. This element is referred to the child node of a word in Wordent taxonomy. The mapping is denoted again by the tree numbers a term may have. For example in

MeSH the term “Eye diseases”, child term nodes are “Asthenopia”, “Vitreoretinopathy, Proliferative”, “Vitreous Detachment” and so on. Term “Eye diseases” has a tree number C11. So the child nodes must be C11.093, C11.975 and C11.980 respectively, which are indeed the tree numbers for the above terms.

Now that the mapping process is understood lets see an example of a MeSH term descriptor record mapped to the “input” xml file.

```
<?xml version="1.0" ?>
<!DOCTYPE DescriptorRecordSet (View Source for full doctype...)>
- <DescriptorRecordSet>
  - <DescriptorRecord DescriptorClass="1">
      <DescriptorUI>D000143</DescriptorUI>
    - <DescriptorName>
        <String>Acids</String>
      </DescriptorName>
    + <DateCreated>
    + <DateRevised>
    + <ActiveMeSHYearList>
    + <AllowableQualifiersList>
      <Annotation>GEN; avoid; do not use for specific acids, acid-fast bacteria, acid reactions, etc.</Annotation>
    - <TreeNumberList>
        <TreeNumber>D01.029</TreeNumber>
      </TreeNumberList>
    + <RecordOriginatorsList>
    - <ConceptList>
      - <Concept PreferredConceptYN="Y">
          <ConceptUI>M0000220</ConceptUI>
        + <ConceptName>
          <ConceptUMLSUI>C0001128</ConceptUMLSUI>
          <RegistryNumber>0</RegistryNumber>
          <ScopeNote>Chemical compounds which yield hydrogen ions or protons when dissolved in water, whose hydrogen
            can be replaced by metals or basic radicals, or which react with bases to form salts and water (neutralization).
            An extension of the term includes substances dissolved in media other than water. (Grant & Hackh's Chemical
            Dictionary, 5th ed)</ScopeNote>
        + <SemanticTypeList>
        - <TermList>
          - <Term ConceptPreferredTermYN="Y" IsPermutedTermYN="N" LexicalTag="NON" PrintFlagYN="Y"
              RecordPreferredTermYN="Y">
              <TermUI>T000412</TermUI>
              <String>Acids</String>
            + <DateCreated>
            + <ThesaurusIDlist>
            </Term>
          </TermList>
        </Concept>
      </ConceptList>
    </DescriptorRecord>
</DescriptorRecordSet>
```

Figure A.2: MeSH Term “Acids” XML descriptor record file

The original term “Acids” XML descriptor record file which is stored in the MeSH database is showed in figure A.2 above. Below in figure A.3 is the mapped term “Acids” input file that similarity measures need.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE word (View Source for full doctype...)>
- <word>
    Acids
    <key>Acids</key>
  - <pos name="noun">
    - <sense number="1">
      - <synset number="1">
          <item>Acids</item>
          <defn>Chemical compounds which yield hydrogen ions or protons when dissolved in water, whose hydrogen can be
            replaced by metals or basic radicals, or which react with bases to form salts and water (neutralization). An
            extension of the term includes substances dissolved in media other than water. (Grant+Hackh s Chemical
            Dictionary, 5th ed)</defn>
        </synset>
      - <links>
        - <hypernym>
          - <synset number="2">
              <item>Inorganic Chemicals</item>
              <item>Chemicals, Inorganic</item>
              <defn>A broad class of substances encompassing all those that do not include carbon and its derivatives as
                their principal elements. However, carbides, carbonates, cyanides, cyanates, and carbon disulfide are
                included in this class.</defn>
            </synset>
          - <hypernym>
            - <synset number="1">
                <item>chemicals and drugs</item>
                <defn>medical higher concept</defn>
              </synset>
            </hypernym>
          </hypernym>
        - <hyponym>
          - <synset number="2">
              <item>Acids, Noncarboxylic</item>
              <item>Noncarboxylic Acids</item>
              <defn>Inorganic acids with a non metal, other than carbon, attached to hydrogen, or an acid radical
                containing no carbon.</defn>
            </synset>
          </hyponym>
        </links>
      </sense>
    </pos>
  </word>
```

Figure A.3: MeSH Term “Acids” XML mapped input file

In the below figure A.4 is showed the location of the term “Acids” in the MeSH taxonomy.

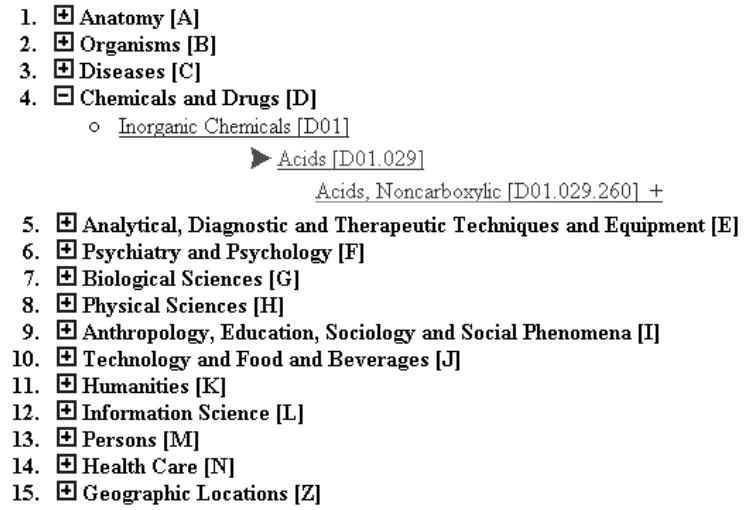

Figure A.4: Location of term "Acids" in MeSH taxonomy

XMLOutput.java is the file that do the mapping for a term from the MeSH Database to a representation in an XML file.

A.6 MeSH DTD file

```
<!-- MESH DTD file for descriptors desc2004.dtd -->

<!ENTITY  % DescriptorReference "(DescriptorUI, DescriptorName)">
<!ENTITY  % normal.date "(Year, Month, Day)">
<!ENTITY  % ConceptReference "(ConceptUI,ConceptName,ConceptUMLSUI?)">
<!ENTITY  % QualifierReference "(QualifierUI, QualifierName)">
<!ENTITY  % TermReference "(TermUI, String)">

<!ELEMENT DescriptorRecordSet (DescriptorRecord*)>
<!ELEMENT DescriptorRecord (%DescriptorReference;,
                            DateCreated,
                            DateRevised?,
                            DateEstablished?,
                            ActiveMeSHYearList,
                            AllowableQualifiersList?,
                            Annotation?,
                            HistoryNote?,
                            OnlineNote?,
                            PublicMeSHNote?,
```

```
                              PreviousIndexingList?,
                              EntryCombinationList?,
                              SeeRelatedList?,
                              ConsiderAlso?,
                              RunningHead?,
                              TreeNumberList?,
                              RecordOriginatorsList,
                              ConceptList) >
<!ATTLIST DescriptorRecord DescriptorClass (1 | 2 | 3 | 4)  "1">

<!ELEMENT ActiveMeSHYearList (Year+)>
<!ELEMENT AllowableQualifiersList (AllowableQualifier+) >
<!ELEMENT AllowableQualifier (QualifierReferredTo,Abbreviation )>
<!ELEMENT Annotation (#PCDATA)> <!ELEMENT ConsiderAlso (#PCDATA) >
<!ELEMENT Day (#PCDATA)> <!ELEMENT DescriptorUI (#PCDATA) >
<!ELEMENT DescriptorName (String) >
<!ELEMENT DateCreated (%normal.date;) >
<!ELEMENT DateRevised (%normal.date;) >
<!ELEMENT DateEstablished (%normal.date;) >
<!ELEMENT DescriptorReferredTo (%DescriptorReference;) >
<!ELEMENT EntryCombinationList (EntryCombination+) >
<!ELEMENT EntryCombination     (ECIN,ECOUT)>
<!ELEMENT ECIN (DescriptorReferredTo,QualifierReferredTo) >
<!ELEMENT ECOUT (DescriptorReferredTo,QualifierReferredTo? ) >
<!ELEMENT HistoryNote (#PCDATA)>
<!ELEMENT Month (#PCDATA)>
<!ELEMENT OnlineNote (#PCDATA)>
<!ELEMENT PublicMeSHNote (#PCDATA)>
<!ELEMENT PreviousIndexingList(PreviousIndexing)+>
<!ELEMENT PreviousIndexing (#PCDATA) >
<!ELEMENT RecordOriginatorsList (RecordOriginator,
                                 RecordMaintainer?,
                                 RecordAuthorizer? )>
<!ELEMENT RecordOriginator (#PCDATA)>
<!ELEMENT RecordMaintainer (#PCDATA)>
<!ELEMENT RecordAuthorizer (#PCDATA)>
<!ELEMENT RunningHead (#PCDATA)>
<!ELEMENT QualifierReferredTo (%QualifierReference;) >
```

```
<!ELEMENT QualifierUI (#PCDATA) >
<!ELEMENT QualifierName (String)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT SeeRelatedList (SeeRelatedDescriptor+)>
<!ELEMENT SeeRelatedDescriptor (DescriptorReferredTo)>
<!ELEMENT TreeNumberList (TreeNumber)+>
<!ELEMENT TreeNumber (#PCDATA)>
<!ELEMENT ConceptList (Concept+)>
<!ELEMENT Concept (%ConceptReference;,
                   CASN1Name?,
                   RegistryNumber?,
                   ScopeNote?,
                   SemanticTypeList?,
                   PharmacologicalActionList?,
                   RelatedRegistryNumberList?,
                   ConceptRelationList?,
                   TermList)>
<!ATTLIST Concept PreferredConceptYN (Y | N) #REQUIRED >

<!ELEMENT ConceptUI (#PCDATA)>
<!ELEMENT ConceptName (String)>
<!ELEMENT ConceptRelationList (ConceptRelation+)>
<!ELEMENT ConceptRelation (Concept1UI,
                           Concept2UI,
                           RelationAttribute?)>
<!ATTLIST ConceptRelation RelationName (NRW | BRD | REL) #IMPLIED>
<!ELEMENT Concept1UI (#PCDATA)> <!ELEMENT Concept2UI (#PCDATA)>
<!ELEMENT ConceptUMLSUI (#PCDATA)> <!ELEMENT CASN1Name (#PCDATA)>
<!ELEMENT PharmacologicalActionList (PharmacologicalAction+)>
<!ELEMENT PharmacologicalAction (DescriptorReferredTo)>
<!ELEMENT RegistryNumber (#PCDATA)>
<!ELEMENT RelatedRegistryNumberList (RelatedRegistryNumber+)>
<!ELEMENT RelatedRegistryNumber (#PCDATA)>
<!ELEMENT RelationAttribute (#PCDATA)>
<!ELEMENT ScopeNote (#PCDATA)>
<!ELEMENT SemanticTypeList (SemanticType+)>
<!ELEMENT SemanticType (SemanticTypeUI, SemanticTypeName)>
<!ELEMENT SemanticTypeUI (#PCDATA)>
```

```
<!ELEMENT SemanticTypeName (#PCDATA)>
<!ELEMENT TermList (Term+)>
<!ELEMENT Term (%TermReference;,
                DateCreated?,
                Abbreviation?,
                SortVersion?,
                EntryVersion?,
                ThesaurusIDlist?)>
<!ATTLIST Term    ConceptPreferredTermYN (Y | N) #IMPLIED
                  IsPermutedTermYN (Y | N) #IMPLIED
                  LexicalTag (ABB|ABX|ACR|ACX|EPO|LAB|NAM|NON|TRD) #IMPLIED
                  PrintFlagYN (Y | N) #IMPLIED
                  RecordPreferredTermYN (Y | N)  #IMPLIED>
<!ELEMENT TermUI (#PCDATA)>
<!ELEMENT String (#PCDATA)>
<!ELEMENT Abbreviation (#PCDATA)>
<!ELEMENT SortVersion (#PCDATA)>
<!ELEMENT EntryVersion (#PCDATA)>
<!ELEMENT ThesaurusIDlist (ThesaurusID+)>
<!ELEMENT ThesaurusID (#PCDATA)>
```

A.7 Evaluation Term pairs

Anemia - Appendicitis
Dementia Atopic Dermatitis
Bacterial Pneumonia - Malaria
Osteoporosis - Patent Ductus Arteriosus
Amino Acid Sequence - Anti-Bacterial Agents
Acquired Immunodeficiency Syndrome - Congenital Heart Defects
Otitis Media - Infantile Colic
Meningitis - Tricuspid Atresia
Sinusitis - Mental Retardation
Hypertension - Kidney Failure
Hyperlipidemia - Hyperkalemia
Hypothyroidism - Hyperthyroidism
Sarcoidosis - Tuberculosis
Vaccines - Immunity
Asthma - Pneumonia
Diabetic Nephropathy - Diabetes Mellitus

Lactose Intolerance - Irritable Bowel Syndrome
Urinary Tract Infection - Pyelonephritis
Neonatal Jaundice - Sepsis
Sickle Cell Anemia - Iron Deficiency Anemia
Psychology - Cognitive Science
Adenovirus - Rotavirus
Migraine - Headache
Myocardial Ischemia - Myocardial Infarction
Hepatitis B - Hepatitis C
Carcinoma - Neoplasm
Pulmonary Valve Stenosis - Aortic Valve Stenosis
Failure to Thrive - Malnutrition
Breast Feeding - Lactation
Antibiotics - Antibacterial Agents
Seizures - Convulsions
Pain - Ache
Malnutrition Nutritional Deficiency
Measles - Rubeola
Chicken Pox - Varicella
Down Syndrome Trisomy 21

A.8 MedSearch Evaluation Queries

1. neonatal pain protocol
2. pain assessment tool
3. sickle cell crisis pain
4. post appendectomy pain control
5. chronic abdominal pain in children
6. complex regional pain syndromes in children
7. childhood migraine headache and acupuncture
8. acute pain coping inventory
9. EMLA and premature infants
10. analgesia for circumcision in newborns
11. pain assessment for nonverbal children
12. humor and pain management
13. opioid and meningitis in children
14. sedation for bone marrow aspiration
15. suturing pain control in children

A.9 Entry terms example

Pain

Pain, Burning
Burning Pain
Burning Pains
Pains, Burning
Pain, Crushing
Crushing Pain
Crushing Pains
Pains, Crushing
Pain, Migratory
Migratory Pain
Migratory Pains
Pains, Migratory
Ache
Aches
Suffering, Physical
Physical Suffering
Physical Sufferings
Sufferings, Physical
Pain, Radiating
Pains, Radiating
Radiating Pain
Radiating Pains
Pain, Splitting
Pains, Splitting
Splitting Pain
Splitting Pains

A.10 API description

All software was developed in Java language. In order to make use of the software, two things are needed

1. The xml files describing the terms we want to compare
2. The .jar file that contains the developed libraries

To run the software as a stand-alone program, you need Ant. Ant has the ability to run everything without setting environmental variables etc. You only have to specify everything in single XML file, paths, properties, functions named "build.xml". The user has the option to select which senses of the terms to compare. Some usage samples follow.

Comparing all senses of term "pain" with term "dementia" using Li et al similarity measure:

```
ant -Dq="pain#0 abdominal_pain#0 liEtAl"

Buildfile: build.xml
init:
compile:
jar:
run:
     [java] Connection was Successfull!
     [java] liEtAl: pain vs dementia = 0.30668439095763

BUILD SUCCESSFUL Total time: 4 seconds
```

Comparing the Most Common Senses (MCS) of terms "pain" and "abdominal_pain" using Jiang et al similarity measure:

```
ant -Dq="pain#0 abdominal_pain#0 jiangEtAl"

Buildfile: build.xml
init:
compile:
jar:
run:
     [java] Connection was Successfull!
     [java] jiangEtAl: pain#0 vs abdominal_pain#0 =
     0.8727664786036737

BUILD SUCCESSFUL Total time: 3 seconds
```

The above shows that compilation has been done previously, no need to be done again. To get full usage details of the program, just type:

```
ant
```

```
Buildfile: build.xml
init:
compile:
jar:
run:

     [java] Usage: ant -Dq="[first concept] [second concept]
     [method]"
     [java]      All arguments are required.
     [java]      Don't use spaces for concepts. Replace spaces with "_".

     [java]      ---------Available methods----------------------
     [java]      Edge counting methods :
     [java]             shortestPath      (for shortest path)
     [java]             wuAndPalmer       (for wu & palmer)
     [java]             leacokChodorow    (for Leacok & Chodorow)
     [java]             weightedLinks     (for shortest path with
     weights)
     [java]             liEtAl            (for Li et al)

     [java]      Information content methods:
     [java]             lin
     [java]             lordEtAl
     [java]             jiangEtAl
     [java]             resnik

     [java]      Complex methods:
     [java]             rodriguez
     [java]             rodriguezOur
     [java]             twersky

     [java]      Cross ontology methods:
     [java]             rodriguezCross25
     [java]             rodriguezCrossMax
     [java]             rodriguezCrossFinal
```

Now, in order to use the functionality of the software in any application, just include the libraries in the code, by writing something like:

```
import thesiscode.*;
import thesiscode.tool.*;
```

somewhere in the beginning of the code.

Figures A.5 and A.6 demonstrate the JSP version running the above similarity measures online (Web app) [7].

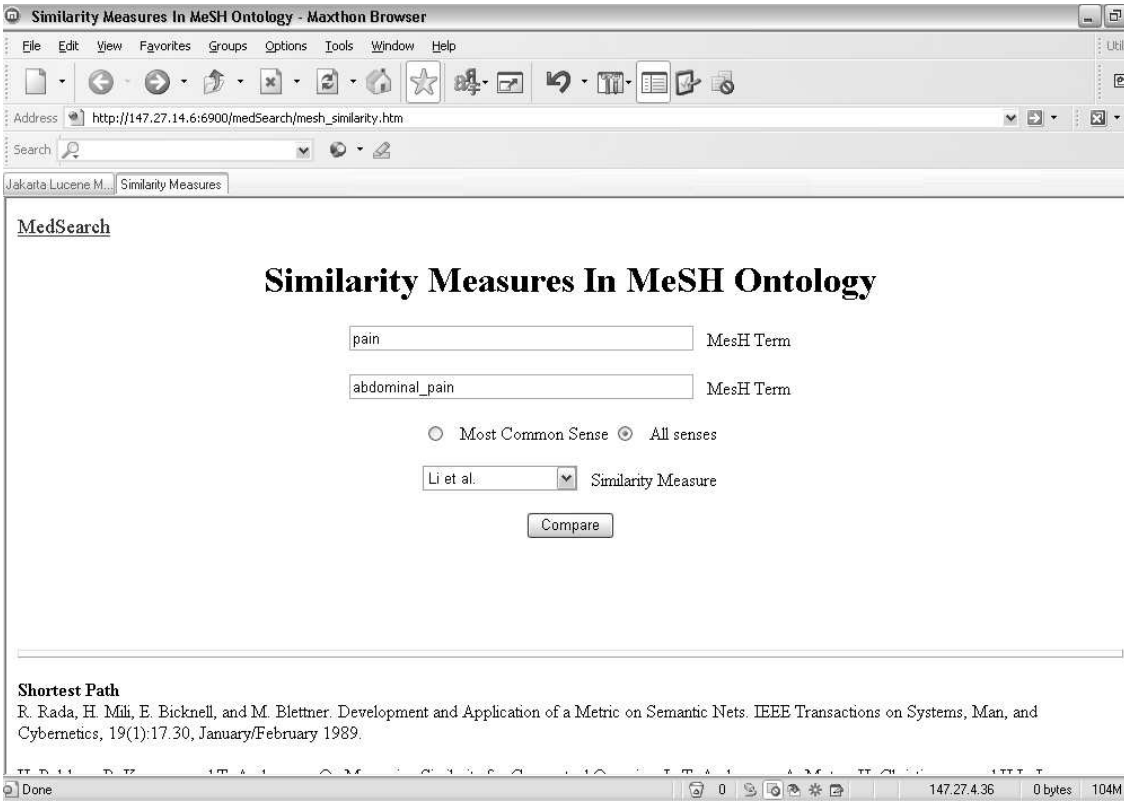

Figure A.5: Similarity measures on the Web (request)

A.11 The Information Retrieval Platform

A.11.1 Lucene

Lucene is a Java-based open source toolkit for text indexing and searching. It is easy to use, flexible, and powerful – a model of good object-oriented software architecture. Powerful abstractions and useful concrete implementations make Lucene very flexible. We use lucene in order to perform indexing, parsing and collecting documents from Medline needed by the IR application search engine we use (chapter 4.3 on page 43). Lucene if freely available at *http://lucene.apache.org*

[7] http://147.27.14.6:6900/medSearch/mesh_similarity.htm

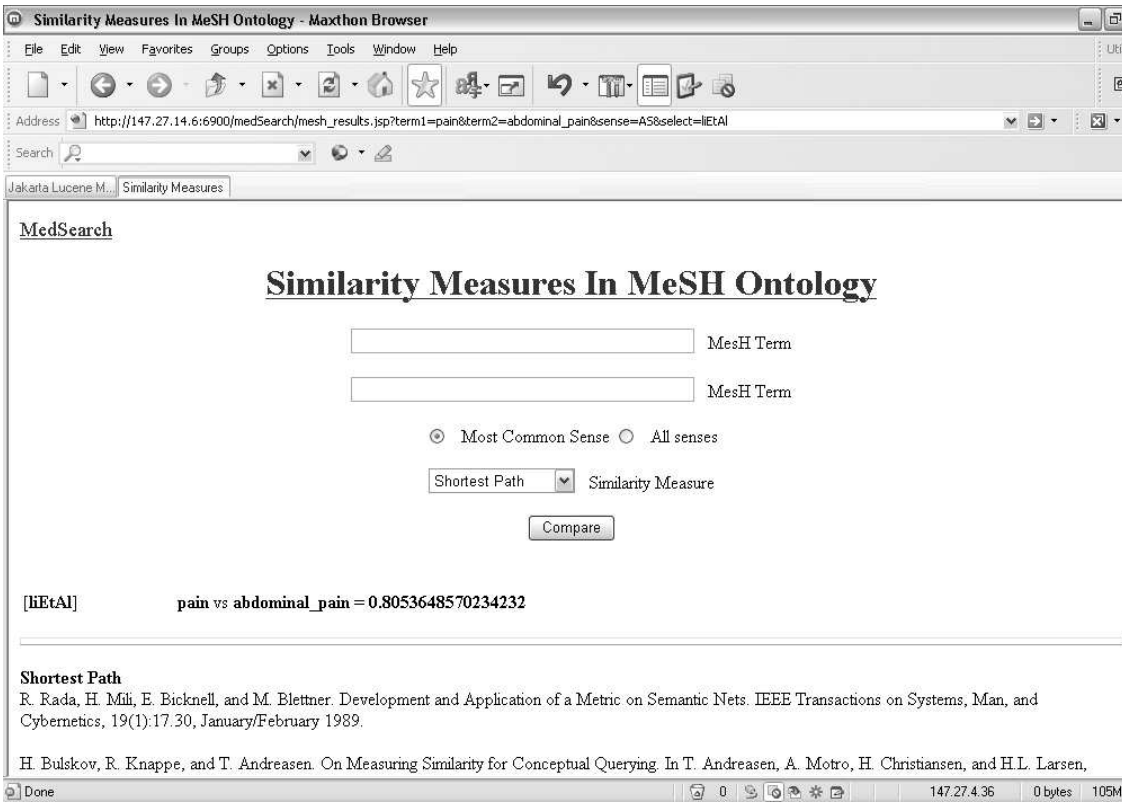

Figure A.6: Similarity measures on the Web (result)

A.11.2 MedSearch Interface

MedSearch application [8] interface is shown in Figure A.7.
The main classes needed for the IR application are:

mapterms all terms represented by XML files are loaded in a cache memory so we can advance the speed for the system.

xmlterms is the class representing the object that is saved in cache. It includes hypernyms, entry terms, hyponyms of each term object.

xml_exist_in_db this class can take an input string and return a string where all terms are only MeSH terms. Term in string are separated by space, while 2 terms represanting one by underscore, eg String = "pain abdominal_pain dementia .."

MCSsynExpansionAll makes the expansion of the query only by entry terms. Needed for method "VSM and Synonyms expansion".

termclass this class keeps the term name and his weight.

[8] http://147.27.14.6:6900/medSearch

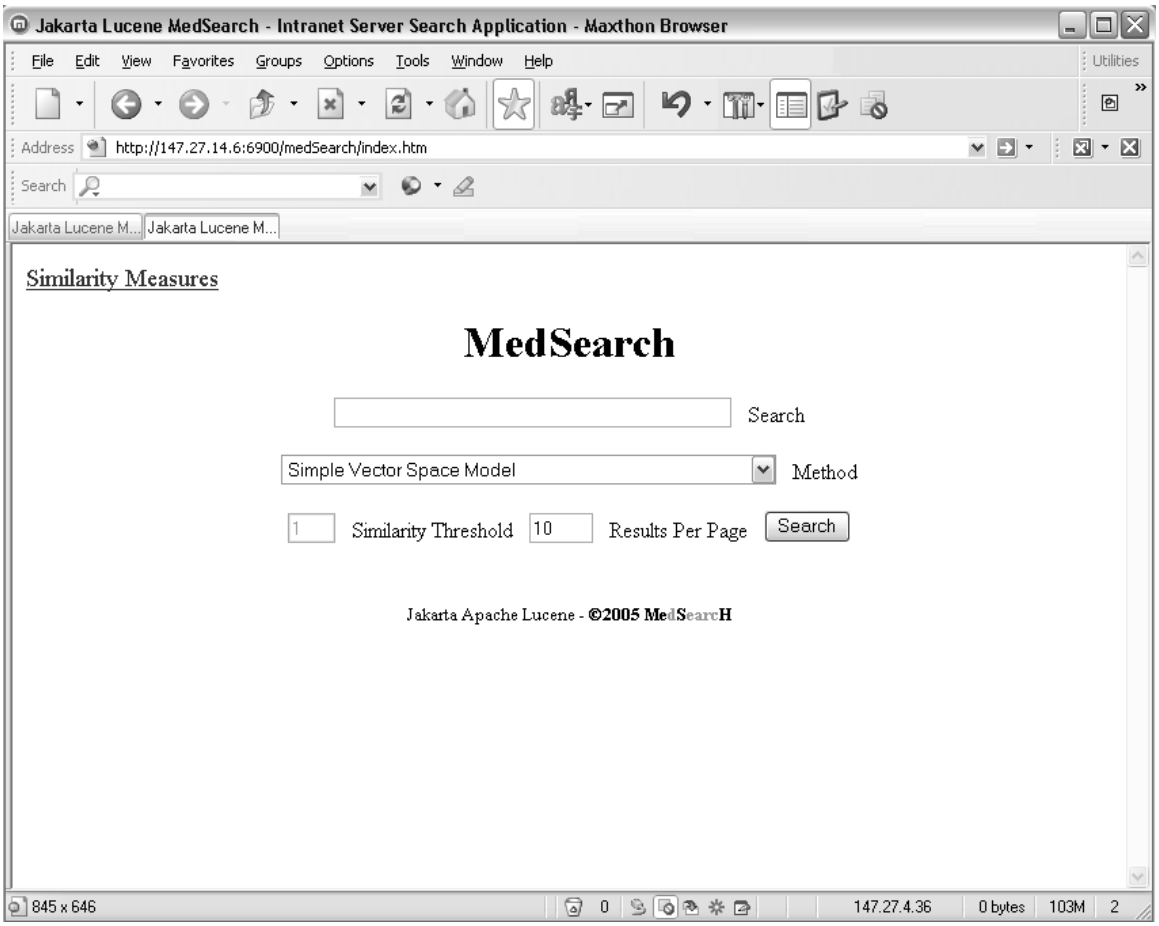

Figure A.7: MedSearch Interface

ExpansionReweight this class do everything for our proposed model SMM. It takes as input a query string and makes expansion and reweight for each (termclass) term. It also provides the SimilarityMatrix function where for a given doc vector as input string, it outputs the corresponding score when compared to the query.

MeSHCollector finally makes the ranking and the normalization for each document score for our application.