AUTOMATED SEMANTIC TAGGING OF TEXT DOCUMENTS

by

Murat Kalender

B.S, Computer Engineering, Yeditepe University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2010

ACKNOWLEDGEMENTS

I wish to express my gratitude to all the people who have given me encouragement and helped me in the completion of this study. Especially, I would like to thank my thesis supervisor, Suzan Üsküdarlı, for her guidance, understanding, and support. She gave me a free hand with my research, always had time for discussions, and asked all the right questions that provided valuable focus for this thesis.

I would like to thank the members of my committee, Pınar Yolum and Emin Erkan Korkmaz for their time and perceptive comments, which have improved the quality of this thesis.

I would especially like to present my gratitude to Jiangbo Dang and Candemir Toklu for their supervision while I was an intern at Siemens Corporate Research in Princeton, US. I also thank my friends Tuğba Külahcıoğlu and Cihan Topal from Siemens Corporate Research, Fethi Ramazanoğlu and İlhan Sezer from Princeton University, and Ali Demir, Özkan Sarı and Caner Akdemir from Yeditepe University for their help in my thesis.

I would also like to thank the members of the SosLab for their help, encouragement, and stimulating interactions. In particular, I would like to thank Ahmet Yıldırım, Nadin Kökciyan and Dağhan Dinç for insightful discussions about this thesis.

I would like to express my gratitude to TUBİTAK for supporting my research with National Scholarship Programme for M.Sc. Students - 2210. This work also is partially funded by B.U. Research Funds (BAP 08A103 and BAP 09HA102P).

Finally, I would like to thank my parents and family for their support and encouragement.

ABSTRACT

AUTOMATED SEMANTIC TAGGING OF TEXT DOCUMENTS

The exponential growth of documents is challenging the existing search and content management technology. An approach for mitigating this issue is user-generated tags, a simple method by which users associate keywords to documents. However, the improvements, from this approach are limited because tags are *i*) free from context and form, *ii*) used for purposes other than description, and *iii*) often remain ambiguous. Since user tagging is a voluntary action, many documents remain untagged. Finally, the interpretation of the tags associated with documents also remains a challenge.

To overcome these challenges, semantic web resources and technologies can be utilized to *automatically* generate *semantic tags*. Semantic tags not only reflect document content more accurately, they also enable better search results. Ontology coverage, word sense disambiguation and weighting significant ontological entities within a context are key challenges in semantic tagging systems.

The leading ontology for the English language, Wordnet, has been successfully used for semantic tagging. However, this approach falls short in tagging documents that refer to new concepts and instances.

The main focus of this work is automatically generating semantic tags for arbitrary documents. For this purpose, the first contribution is an ontological knowledge base platform called UNIpedia. UNIpedia aims to provide a knowledge base with contemporary references. Here, contemporary should be understood as in line with web pace. UNIpedia maps various ontological knowledge bases to WordNet concepts. The Wikipedia and OpenCyc knowledge bases, which are known to contain up to date instances and reliable metadata about them, were mapped to WordNet. A rule based heuristics, which uses

the ontological and statistical features of concepts and instances, is introduced for the mapping process.

UNIpedia terms may have several senses because of the natural language ambiguity. These so called polysemous terms get different meanings according to the context. A term passing in a document cannot be mapped to an UNIpedia concept or instance directly, if the term is polysemous. In order to identify the correct sense of the polysemous terms, an automated semantic tagging system called Semantic TagPrint was devised. Semantic TagPrint is the second contribution of this work that uses a linear time lexical chaining Word Sense Disambiguation algorithm for semantic annotation. In addition, Semantic TagPrint weighs and recommends semantic tags which describe the content of a document well. The semantic annotation and semantic tag weighting algorithms use both semantic and statistical features of UNIpedia.

The potential benefits of Semantic TagPrint are demonstrated by the design and implementation of the Semantic Knowledge Management Tool (SKMT). SKMT is the third contribution of this work that provides a user accessible platform for Semantic TagPrint to semantically tag documents, and performs semantic searches.

ÖZET

METİN BELGELERİNİN OTOMATİK OLARAK ANLAMSAL ETİKETLENMESİ

Belgelerin katlanarak büyümesi mevcut arama ve içerik yönetim teknolojilerini zorlamaktadır. Bu sorunu azaltmak için bir yaklaşım belgeların kullanıcılar tarafından seçilen belgelerde geçen önemli kelimelerle etiketlenmesidir. Ancak bu yaklaşımın etiketleri sınırlıdır çünkü etiketler *i*) bağlam ve form özgür, *ii*) belgeleri tanımlamadan farklı amaçlarda kullanılabiliyor *iii*) genellikle belirsiz kalıyorlar. Etiketleme gönüllü bir eylem olduğundan dolayı çok sayıda belge etiketlenmemektedir. Son olarak, belgelere atanan etiketlerin yorumlanmasıda ayrı bir zorluktur.

Anlamsal web kaynakları ve teknolojileri, bu zorlukları aşmak ve otomatik olarak semantik etiketler oluşturmak için kullanılabilir. Semantik etiketler belgelerin içeriğini daha iyi ifade etme dışında, daha iyi arama sonuçları elde etmemizi sağlamaktadır. Ontoloji kapsamı, terimlerin ontolojide doğru kavramlarla ilişkilendirilmesi ve anlamsal etiketlerin ağırlıklarının belirlenmesi anlamsal etiketleme sistemlerinde çözülmesi gereken önemli sorunlardır.

İngilizce için önde gelen ontoloji olan WordNet başarıyla anlamsal etiketleme için kullanılmaktadır. Ancak bu yaklaşım yeni kavramlar içeren belgeleri etiketlemede yetersiz kalmaktadır.

Bu çalışma belgeler için otomatik olarak anlamsal etiketler oluşturan bir sistem önermektedir. Bu amaçla, ilk katkımız ontolojik bilgi tabanı platformu olan UNIpedia' dır. UNIpedia çağdaş referansları içeren bir bilgi tabanı sağlamaktır. Burada, çağdaş kelimesi web de geçen güncel kelimeler bağlamında kullanılmaktadır. UNIpedia çeşitli ontolojik bilgi tabanlarını WordNet kavramlarıyla ilişkilendirmektedir. Güncel ve güvenilir bilgi içeren Wikipedia ve OpenCyc bilgi tabanları WordNet kavramları ile eşleştirilmiştir. Bilgi

tabanlarını ilişkilendirmek için kavramların ontolojik ve istatistiksel özelliklerini kullanan kural tabanlı sezgiseller kullanılmıştır.

Konuşma dillerinin çok anlamlılığından dolayı UNIpedia' da tanımlı terimler birden fazla anlam içerebilmektedir. Bu çok anlamlı kelimeler dökümanın içeriğine göre farklı anlamlar alabilmektedirler. Belgede geçen terimler çok anlamlıysa doğrudan UNIpedia kavramlarıyla ilişkilendirilememektedir. Terimlerin doğru anlamlarını bulabilmek için otomatik anlamsal etiketleme sistemi olan Semantic TagPrint geliştirilmiştir. Bu eserin ikinci katkısı olan Semantic TagPrint anlam belirginleştirmesi için doğrusal zamanda çalışan kelime zincirlerini kullanmaktadır. Buna ek olarak, Semantik TagPrint belgenin içeriğini açıklayan anlamsal etiketlerin önemini belirler ve önerir. Anlamsal etiketleme ve önerme algoritmaları UNIpedia da tanımlı olan kavramların istatistiksel ve anlamsal özelliklerini kullanmaktadır.

Semantik TagPrint sisteminin potansiyel yararlarını göstermek için Anlamsal Bilgi Yönetimi Aracı (SKMT) uygulanması tasarlanmış ve geliştirilmiştir. Bu eserin üçüncü katkısı olan SKMT semantik belgeleri etiketlemek için Semantik TagPrint için erişilebilir bir platform sunar ve semantik arama yapar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

AKTRO	AKT Reference Ontology
DP	Depth
F	F-measure
GP	Google Popularity
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IC	Information Content
IGP	Inverse Google Popularity
JUNG	Java Universal Network/Graph Framework
KM	Knowledge management
MRDs	Machine-readable dictionaries
NGP	Normalized Google Popularity
NLP	Natural Language Processing
OWL	Web Ontology Language
P	Precision
R	Recall
RDF	Resource Description Framework
SKMT	Semantic Knowledge Management Tool
SVM	Support Vector Machines
TBD	Taxonomy-based Disambiguation
TF	Term Frequency
WSD	Word Sense Disambiguation
W3C	World Wide Web Consortium
2D	Two Dimensional
3D	Three Dimensional

1. INTRODUCTION

Electronic documents have become a de facto part of life. The ease of publication and distribution of web documents has led to an explosion in the quantity of such documents. By the end of 2008 it was reported [1] that there were 230 million Web sites and 133 million blogs. Processing such a quantity of documents and locating desired information is very challenging, yet necessary.

With the ever increasing amount of content, we heavily rely on search engines to locate documents. However, existing search tools are experiencing difficulties, as keyword based search often returns many results with little relevance. Users waste their time locating the required information among the plenty of documents returned in the search results.

Tagging can improve search results by allowing search engines to exploit tags generated from the wisdom of crowds. It is a simple method by which users associate keywords to content such as documents, Web pages, pictures and videos [2]. Good tags provide relevant and brief information about resources. Tagging helps users to describe, find, and organize content [3]. The user generated tagging approach has resulted in improvements in locating information, and as a result is getting more popular. Currently, many popular Web sites support tagging (i.e. Delicious [4], Facebook [5], Flickr [6], and YouTube [7]). For example, Delicious is a social bookmarking Web site (saves Web addresses) where the users can tag their bookmarks with freely chosen terms [8]. With the help of these tags, the users can discover web sites relevant to their interests.

While user generated tagging can be effective, they have some drawbacks because *i*) tags are subjective and inconsistent since they are non-hierarchical keywords assigned in a random way; *ii*) tags are inherently heterogeneous because they are from different sources and people; *iii*) tags usually fail to capture exact meanings and contexts of keywords since human languages are ambiguous, e.g. there are polysemous words; and *iv*) tagging requires additional effort . Because of the above issues, many documents remain untagged.

Using programs that generate tags automatically from the documents rather than relying on the users is called automatic tagging. Automatic tagging systems analyze given documents and offer significant terms as tags. This approach provides accuracy, consistency, standardization, and convenience, while decreasing the cost at the same time. An automatic tagging system would address the first, second and the fourth drawbacks listed above. This method of tagging can be further improved by assigning ontological entities (concepts and instances) instead of keywords to documents, which is called semantic tagging.

Semantic tagging provides the required environment for realizing the semantic search, which is based on the meanings of the search terms. Thus, semantic tagging systems would address the third drawback. Semantic tagging systems use ontologies as knowledge sources. An ontology is a knowledge source that consists of representational primitives, which is used to model a domain. The representational primitives are usually concepts (classes), instances (objects), relations, and properties [9]. Concepts are the basic elements of an ontology.

There are several lexical and structural ontologies that can be used for semantic tagging. While many of them are domain-specific, some are domain-independent. WordNet [10] and Wikipedia [11] are widely used as domain-independent knowledge bases. Wordnet is a lexical knowledge base that covers most concepts defined in the English language. However, it does not contain most named instances of the concepts it covers – such as of people, organizations, geographic locations, books, and songs. Furthermore, many new (contemporary) concepts are also not covered. Therefore, many current documents often contain terms that are not found in WordNet. On the other hand, Wikipedia covers up-to-date content, but lacks formally defined hierarchical relationships among instances. Thanks to user contributions its content rapidly evolves and remains up to date, which is very desirable when dealing with current documents.

The success of semantic processing clearly depends on the term coverage of the ontologies utilized. In order to effectively process domain-independent Web documents, a comprehensive, up-to-date, and evolving ontology is required.

A term passing in a document cannot be annotated with a concept or instance directly, if the term is polysemous. Thus, a Word Sense Disambiguation (WSD) algorithm is required that automatically maps a polysemous word to an appropriate sense (meaning) according to the context it is used.

The significance of the semantic tags within a document is an important metric for ranking semantic search results. Documents, in which searched terms are significant, can be listed on top in search results. The significant terms are also very useful to briefly describe content of a document.

1.1. Proposed Solution

The goal of this thesis is to build a high performance real time semantic tagging system that automatically maps noun phrases of English text documents to entities defined in an ontology using Semantic Web approaches. Key challenges in semantic tagging systems such as ontology coverage, word sense disambiguation, and weighting significant ontological entities within a context are addressed.

A term within a document can be identified and tagged, if it is defined in an ontology. Therefore, the number of entities (coverage) defined in an ontology is an important factor that effects the tagging quality. To enhance the coverage of the Semantic TagPrint knowledge source, an ontology framework (UNIpedia) that combines knowledge bases is proposed. UNIpedia aims to serve as a high quality, comprehensive, up-to-date, domain independent and easily re-usable resource for semantic applications. UNIpedia uses WordNet as its backbone ontology, and maps instances from other knowledge bases to WordNet concepts by introducing an *isA* relationship between them.

UNIpedia terms may have several senses. These polysemous terms get different meanings according to the context. A term passing in a document cannot be annotated with an UNIpedia concept or instance directly if the term is polysemous. To annotate it to the right UNIpedia concept (sense), a linear time WSD algorithm is proposed that uses both semantic and statistical features of UNIpedia.

Once the semantic tags are identified, they are assigned weights to indicate their significance. These weights are computed based on statistical and ontological features of the terms. The semantic tags, with higher weights are recommended to Semantic TagPrint users.

A graphical user interface to visualize the results of the semantic tagging algorithm has been implemented. With this interface, input parameters of the WSD algorithm can be modified and the generated output can be viewed in two dimensional and three dimensional semantically connected graphs.

Finally, the potential benefits of Semantic TagPrint are demonstrated by the design and implementation of the Semantic Knowledge Management Tool (SKMT). SKMT provides a user accessible platform for Semantic TagPrint. SKMT uses Semantic TagPrint to extract metadata for given documents. The extracted metadata is used to enable both semantic and conventional keyword-based search. SKMT is used to compare semantic search with keyword based search on sample search scenarios.

The performance of the mapping between Wordnet and Wikipedia is evaluated with experiments in Section 5.1. Experiments show that the accuracy of the mapping between WordNet and Wikipedia is 84% for the most relevant concept name and 90% for the appropriate sense.

Comparative evaluation of the features used in Semantic TagPrint WSD algorithm is performed. The features are evaluated on the SemCor [12] sense annotated dataset. The evaluation shows that the WSD algorithm is fairly accurate.

The tag recommendation performance of Semantic TagPrint is compared to context-based (tags documents based on their context) and publicly available tagging systems Zemanta [13] and Yahoo Term Extraction [14] using a news data set. Additionally, Semantic TagPrint is compared as a baseline algorithm that suggests frequently used noun phrases within a document extracted by Semantic TagPrint. The tag recommendation algorithm performs better than other systems and algorithms.

1.2. Outline

The outline of this thesis is as follows. Chapter 2 gives background information and related work about semantic tagging. Chapter 3 presents a detailed model of the Semantic TagPrint system. Chapter 4 provides the implementation decisions in detail. Chapter 5 presents the evaluations of UNIpedia mappings, word sense disambiguation, and semantic tag recommendation algorithms. Finally, Chapter 6 presents conclusions and proposes ideas for future work.

2. BACKGROUND AND RELATED WORK

This chapter provides background regarding the use of thesis is reviewed. Semantic Web for semantic, in tagging; semantic tagging systems; and knowledge sources are introduced, which are matched for construction of UNIpedia with ontology matching approaches in the literature. Finally, detailed information about search technologies is provided.

2.1. Semantic Web Technologies

Semantic Web is an extension of the current Web, in which data is represented in a standard format with metadata that allows integration and processing of different data sources automatically by computer programs [15]. Metadata (data about data) is a structured information about content and properties of documents to support their automatic processing [16]. Semantic Web covers several standard languages for metadata representation [17]. In this study, the Resource Description Framework (RDF) [18] and the Web Ontology Language (OWL) are utilized for representing metadata. RDF is an XML-based language which is essentially a data model for knowledge representation. It represents attributes and relationships with a statement which consists subject-predicate-object [19]. Figure 2.1 shows how the knowledge of *Turkey is a country* could be represented in the standard RDF/XML format (*scr:Turkey* is the subject, *scr:isA* is the predicate, and *Country* is the object).

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:scr="http://www.scr.siemens.com/TagPrint#">
  <rdf:Description rdf:about="scr:Turkey">
    <scr:isA>Country</scr:isA>
  </rdf:Description>
</rdf:RDF>
```

Figure 2.1. RDF representation of the statement *Turkey is a country*

OWL [20] is an XML-based language used for describing ontologies. An ontology is a knowledge source that consists of representational primitives, which is used to model a do-

main. The representational primitives are usually concepts (classes), instances (objects), relations, and properties [9]. Concepts are the basic elements of an ontology. A concept is described with a definition and set of properties [21]. Concepts are generally formed in a hierarchical arrangement (taxonomy) by isA-relations (type). For example, the concept *computer science* is a type of the concept *engineering* in WordNet [10]. Instances are named entities of concepts such as of people, organizations, geographic locations, books and songs. For example *Istanbul* is defined as an instance of the concept *city* in WordNet. OWL language is capable of describing concepts, instances, and relationships among them. Figure 2.2 shows the description of the concept *apple (fruit)* in OpenCyc ontology. The *label* property states the concept name, the *prettyString* properties state the synonym names, the *subClassOf* property states the parent concept, and *seeAlsoURI* states the corresponding WordNet concept of the *apple* concept in Figure 2.2.

```
    <owl:Class rdf:about="Apple">
        <rdfs:label xml:lang="en">apple</rdfs:label>
        <prettyString xml:lang="en">apples</prettyString>
        <prettyString xml:lang="en">fruit of the Malus pumila</prettyString>
        <prettyString xml:lang="en">fruit of the apple tree</prettyString>
        <cycAnnot:label xml:lang="en">(FruitFn AppleTree)</cycAnnot:label>
        <rdfs:comment xml:lang="en">The collection of individual
apples.</rdfs:comment>
<cycAnnot:externalID>Mx8Ngh4rvVipdpwpEbGdrcN5Y29ycB4rvVjBnZwpEbGdrcN5Y29ycA</
cycAnnot:externalID>
        <rdf:type rdf:resource="DefaultDisjointEdibleStuffType"/>
        <rdf:type rdf:resource="LifeStageType"/>
        <rdf:type rdf:resource="SpatiallyDisjointObjectType"/>
        <rdfs:subClassOf rdf:resource="EdibleFruit"/>
        <quotedIsa rdf:resource="WordNetWorkflowConstant_NotFullyReviewed"/>

<seeAlsoURI>http://www.w3.org/2006/03/wn/wn20/instances/synset-apple-noun-1</
seeAlsoURI>
        <owl:sameAs
rdf:resource="http://sw.cyc.com/concept/Mx8Ngh4rvVipdpwpEbGdrcN5Y29ycB4rvVjBn
ZwpEbGdrcN5Y29ycA"/>
        <owl:sameAs
rdf:resource="http://sw.opencyc.org/2009/04/07/concept/Mx8Ngh4rvVipdpwpEbGdrc
N5Y29ycB4rvVjBnZwpEbGdrcN5Y29ycA"/>
        <owl:sameAs
rdf:resource="http://sw.opencyc.org/concept/Mx8Ngh4rvVipdpwpEbGdrcN5Y29ycB4rv
VjBnZwpEbGdrcN5Y29ycA"/>
    </owl:Class>
```

Figure 2.2. Description of the *Apple (fruit)* using OWL in OpenCyc ontology

2.2. Semantic Tagging

The generation of assignments of ontological entities (concepts and instances) to documents is called semantic tagging (or semantic annotation). Semantic tags are same as metadata for documents. Figure 2.3 shows an example mapping of a piece of textual content to semantic tags defined in an ontology and knowledge base [22]. The ontological concept classes are represented as circles (i.e. *City* and *Painter*), while the concept instances are shown as rectangles (i.e. *Philadelphia* and *Thomas Eakins*).

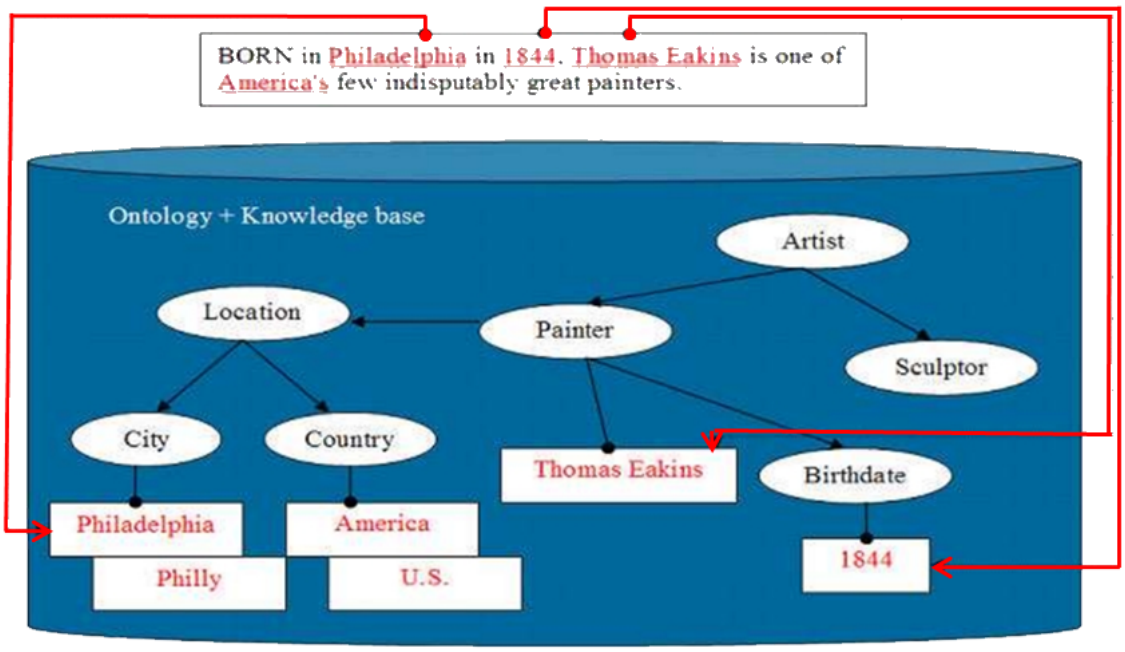

Figure 2.3. An example mapping of a piece of textual content to semantic tags defined in an ontology and knowledge base

Natural language ambiguity makes the automatic assignment of ontological entities challenging. Thus, there are three classes of semantic tagging systems in terms of the semantic annotation automation: manual, semi-automatic and automatic [22].

In manual semantic tagging systems, users tag documents with a controlled vocabulary defined in an ontology. Manual tagging is a time consuming process, which generally requires deep domain knowledge and expertise for domain specific ontologies, and which also introduces inconsistencies by human annotators.

Semi-automatic systems analyze documents and offer ontological terms, from which annotators may choose. Semi-automatic semantic tagging utilizes humans to disambiguate polysemous terms. Semi-automatic systems make the tagging process easier for the taggers in contrast to the manual tagging systems.

Automated semantic tagging systems analyze documents, disambiguate polysemous terms, and automatically tag them with ontological concepts and instances. The automated systems remove dependency to human annotators. However, this approach lowers the accuracy of the mappings because of difficulty of disambiguating terms in a context.

2.2.1. Word Sense Disambiguation

Natural languages are ambiguous; words have different meanings according to the context in which they are used. For instance, consider the following sentences:

- l live in The Java Islands.
- I prefer java over php.
- I can't wake up without my java juice.

The word *java* has different meanings in each of the above sentences, where the word *java* refers to an *island*, a *programming language* and *coffee*, respectively. Word Sense Disambiguation (WSD) is the process of automatically mapping a polysemous word (a word having many meanings) to an appropriate sense (meaning) according to the context in which it is used. In most cases, it is easy for humans to disambiguate words. However, WSD is a complex problem that is difficult to solve automatically. Consequently, low success rates are observed in WSD dependent application areas, such as machine translation, where grammatically correct yet meaningless sentences are common [23].

WSD consists of two main tasks:

- Lexical sample (or targeted WSD) disambiguates only preselected sets of words (target) for a given content. For example, the disambiguation of each occurrence of the word *java*. Supervised learning based systems are generally fall into this group.

These kind of systems uses training data sets for reasoning about preselected sets of words [23].

- All-words WSD deals with all words that occur in text documents, which are typically nouns, verbs, adjectives, and adverbs [23].

WSD algorithms use knowledge sources about words for the purpose of disambiguation. There are two main types of knowledge sources: structured and unstructured. Thesauri, machine-readable dictionaries, semantic networks are type of structured sources. Corpus and collocation resources are unstructured sources [23]. These knowledge sources are described as:

i. A thesaurus includes words and lexical relationships between words such as synonyms and antonyms. Two words are synonymous, if they have the same meaning, i.e. *buy* and *purchase*. Two words are antonymous, if they have the opposite meanings, i.e. *buy* and *sell*. Roget's International Thesaurus [1] is widely used for WSD.
ii. Machine-readable dictionaries (MRDs) are electronic dictionaries, which can be processed by computers. They may be mono- or multilingual. For example, the Oxford English Dictionary [24] is often used for WSD.
iii. Ontologies are knowledge sources that consist of concepts, instances, relations, and properties, used to model a domain. Semantic properties of words defined in an ontology is used for the disambiguation of words. WordNet, as an otology (see section 2.3.1), is widely used for WSD purposes.
iv. Corpora are collection of text documents. There are two types of corpora: raw and sense annotated. Raw corpora are used for statistical analysis in the Natural Language Processing (NLP) field. For example, the Brown Corpus [25] contains 500 text documents from different genres ranging from newspaper reports to technical writings [12]. Corpora may be sense annotated to go beyond statistical analysis. In this case, each word in a corpus is labeled with their sense and part of speech (verb, noun, adjective, etc.). SemCor [12] is the most widely used sense-annotated (or sense-tagged) corpus, which is derived from the manual semantic annotation of The Brown Corpus with WordNet senses.

[1] Roget's International Thesaurus consists of 250,000 English words

v. Collocation resources contain information about the tendency of words to occur frequently with others. The Word Sketch Engine [26] and JustTheWord [27] are examples of collocation resources [12].

There are three main approaches to WSD: *Supervised*, *Unsupervised*, and *Knowledge-Based*.

- Supervised approaches use sense annotated data sets as a training data for learning disambiguation patterns. Support Vector Machines (SVM) [28], Decision Trees [29] and Neural Networks [30] are widely used supervised WSD methods.
- Unsupervised systems use a raw corpus as a training data for learning disambiguation patterns. Word Clustering [31] and Co-occurrence Graphs [32] are widely used unsupervised methods.
- Knowledge-Based approaches use structured resources such as MRDs and ontologies for disambiguating words. These algorithms are preferred because of their wider coverage despite their lower performance in comparison to machine learning approaches. The overlap of sense definitions [33], selectional restrictions [34] and structural approaches are the most commonly used knowledge based algorithms in WSD [23]:
  i. The overlap of the sense definitions algorithm disambiguates senses by comparing and counting number of overlapping words between sense descriptions for given two or more words. For example, the definition of the sense *java (object-oriented programming language)* is "a simple platform-independent object-oriented programming language used for writing applets that are downloaded from the World Wide Web by a client and run on the client's machine". And the definition of the sense *prolog (programming language)* is "a computer language designed in Europe to support natural language processing". Definitions of these senses have the overlapping words *programming* and *language*.
  ii. Selectional restriction algorithms disambiguate senses by restricting the possible meanings of senses according to their surrounding words. An example restriction could be between the *drink* and *java* words. When these words are used in the same sentence, the word *java* could be mapped to its *drink* meaning, which is defined in WordNet.
  iii. Structural approaches disambiguate senses by semantic interrelationships of con-

cepts to disambiguate words. In a local context, semantic similarities between pairs of words within content are calculated according to similarity measures. There are a number of proposed measures, which are described in detailed in [35]. In a global context, semantically related words are connected to each other and they form sets of lexical chains or semantic graphs [36]. If a sense of a word is connected to a longer chain than the word's other meanings, it is selected as the meaning of the word in this context. Figure 2.4 shows an example of a lexical chain in which senses are connected with *kind-of* and *has-kind* semantic relations [23].

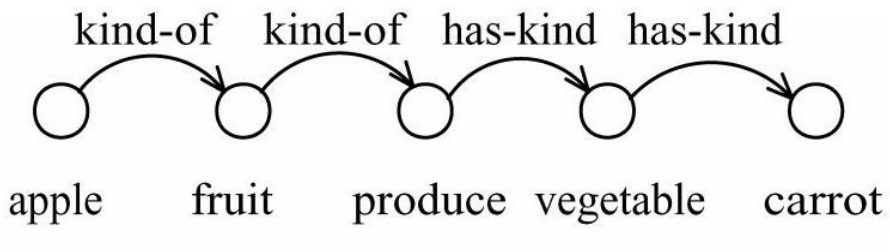

Figure 2.4. An example of a lexical chain in which senses are connected with *kind-of* and *has-kind* semantic relations

Supervised systems generally perform better than unsupervised and knowledge-based approaches in WSD tasks. However, supervised systems rely on sense annotated data sets, which is difficult to find and process.

2.2.2. Keyphrase Extraction

Keywords or keyphrases are significant terms of a document that briefly describe document's content [37]. Using programs that extracts significant terms that compose of multiple words from the documents is called *keyphrase extraction*. If the extracted terms are composed of single words, then it is called *keyword extraction*. For example, keyphrases are mostly defined after the *abstract* section of the academic articles. These keyphrases give valuable information regarding the content of the articles. There are three main approaches utilized for keyphrase or keyword extraction: extraction based on statistics, machine learning, and shallow semantic analysis.

- Statistical methods use statistical features such as term frequency [38], co-occurrence frequency of words [39], N-Gram [40], etc. Statistical methods are widely used; however they have low accuracy values [41]. Because, these kind of methods offer keyphrases without analyzing the context of documents.
- Machine learning algorithms use statistical learning methods such as Decision Trees [42], Naive Bayes [43], and Support Vector Machines [44]. Machine learning algorithms perform better than other approaches, but they heavily rely on the training data in a given domain.
- Shallow semantic analysis algorithms use semantic features for keyphrase extraction. Lexical Chains and Semantic Graphs are examples of shallow semantic analysis algorithms. These kinds of algorithms are getting popular due to the overwhelming limitations of other two approaches. Semantic analysis approaches use semantic networks and ontologies as knowledge sources. Semantically related words are connected to each other to form sets of lexical chains. These lexical chains are utilized in calculating term significance. Semantic analysis based systems are limited to the words defined in the knowledge bases, they utilize [41]. Such systems generally use either WordNet or Wikipedia, which are comprehensive knowledge bases. This is specifically relevant for tagging general purpose documents.

2.2.3. Related Work in Automated Semantic Tagging

There are several systems that extract keywords or keyphrases for a given content, using semantic features and knowledge bases [37, 45, 46].

Ercan et al. [45] use the lexical chaining algorithm and WordNet ontology for keyword extraction. They utilize hypernym, synonym, and holonym properties of WordNet to build the lexical chains. They evaluated their system on a journal data set [47]. They got 30% precision with 10 keywords for each document. In contrast, Semantic TagPrint utilizes WordNet, Wikipedia and OpenCyc and offers keyphrases instead of keywords.

Li et al. [46] also use the lexical chaining algorithm to disambiguate terms and extract keyphrases. Their system uses HowNet, which is a Chinese common-sense knowledge base. They also use statistical value word co-occurrence besides the semantic features. They

have evaluated their algorithms on a Chinese news data set, which is gathered from the web site [48]. They have observed 61% precision and 57% recall values.

Shi et al. [37] utilize semantic and statistical features for keyphrase extraction. They use Wikipedia as a knowledge base. Because of the complex hierarchical relationships among instances of Wikipedia, they utilize the methods proposed in [49] to derive acyclic instance hierarchies. Their algorithm scored 38% precision on a CNN news data set. In contrast, Semantic TagPrint uses WordNet's taxonomy and it is scored 52.5% precision for the CNN news data set created for this study.

2.3. Ontologies and Knowledge Bases for Semantic Tagging

The performance of semantic tagging and search engines is highly dependent on the ontologies. It relies on term within content can be semantically tagged and retrieved, only if it is defined in an ontology. There are several comprehensive and publicly available ontologies and ontological knowledge bases that can be used for semantic tagging [10, 11, 50, 51]. In the following sections, the knowledge resources relevant to this work in this thesis, are presented.

2.3.1. WordNet as an Ontology

WordNet [10] is a large lexical database for the English language. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. A synset has the following semantic properties in WordNet:

- Hyponymy/hypernymy properties define an *isA* relationship between synsets. Hyponymy is transitive and asymmetrical. Each synset has a single parent except the root. For example, the synset *scientist* is a hyponym (child) of the synset *person* and the synset *person* is a hypernymy (parent) of the synset *scientist*.
- Domain is another semantic property that groups synsets into different topics. For example, the synset *object-oriented programming language* is defined under the domain of *computer science*.
- Holonymy/meronymy properties define *partOf* and *hasPart* relationships between

synsets. For example, the synset *wheel* is a holonymy (part of) of synset car and the synset car has a meronymy (has part) the synset *wheel*.

Figure 2.5 shows sample synsets from WordNet with their semantic properties [23].

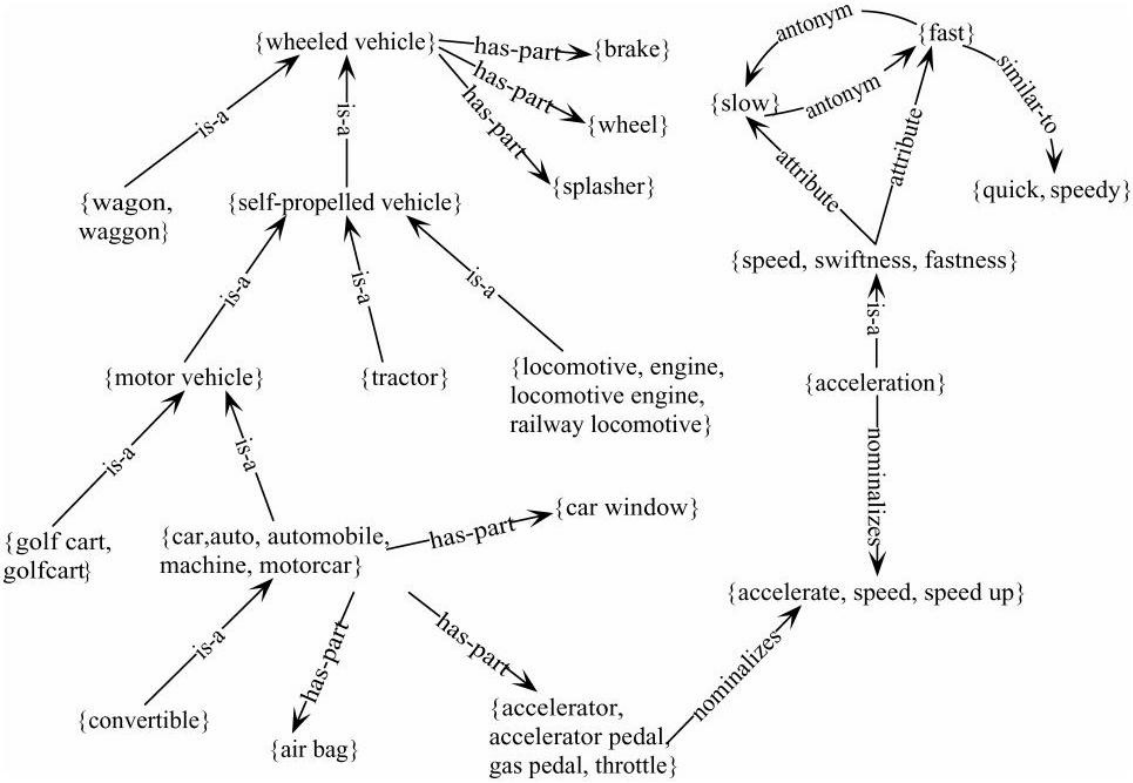

Figure 2.5. A graph showing semantic properties of sample WordNet synsets

WordNet is used as an an ontology by Semantic TagPrint in which synsets correspond to concepts that are semantically interrelated with hypernym, domain and meronymy properties. WordNet has 146,312 defined noun word sense pairs in English. While WordNet also includes some instances such as of people, organizations, geographic locations, books, songs, etc, it does not contain most named instances of the concepts it covers. Such instances are vital and should be included in ontologies in an evolutionary manner for the Semantic Web systems, which deals with the contemporary documents [52].

2.3.2. Ontological Knowledge Bases

There are several comprehensive and publicly available ontological knowledge bases [11, 50, 51]. In this thesis, Wikipedia, DBpedia, and OpenCyc are used.

Wikipedia [11] is a comprehensive online encyclopedia collaboratively written by volunteers. Each Wikipedia article has an unique title, which can be treated as named instances. Redirection links within an article can be considered links to synonymous articles. Some articles contain infoboxes [53], which summarizes key information, such as birth date and occupation of people, population and coordinates of cities, etc. Unlike in ontologies, Wikipedia articles do not have formally defined hierarchical relationships among articles. An article may be categorized in numerous ways. For example, *Barack Obama* is categorized as *Audio book narrators*, *1961 births*, *people from Hawaii*, *Presidents of US*, *Harvard Law School Alumni*, etc. Such categories provide valuable information about the article.

OpenCyc [51] is a domain independent knowledge base that contains hundreds of thousands of concepts and millions of statements. In OpenCyc, there are relationships such as *aliases*, *a type of*, *instance of*, *subtypes*, *instances*, *same as* defined between instances. It contains some links to WordNet and Wikipedia. However, the links are sparse and all its classes are not linked to WordNet and Wikipedia. In June 2008, the OWL version of OpenCyc contained 111,694 classes and 11,251 of them are linked to WordNet concepts [51].

The DBpedia [50] system extracts structured multilingual information from Wikipedia articles. It extracts information from infoboxes, categories, links, etc., and represents data in RDF format. DBpedia contains some links to WordNet. However, the links are sparse and all its classes are not linked to WordNet and Wikipedia. DBpedia version 3.5 contains 3,144,000 English Wikipedia articles and 330,000 of them are linked to WordNet concepts.

The DBpedia and OpenCyc links to WordNet ontology are part of the Linked Data [54] project. The Linked Data project aims to connect open machine readable data sets for acquiring information about common concepts on different sources automatically using

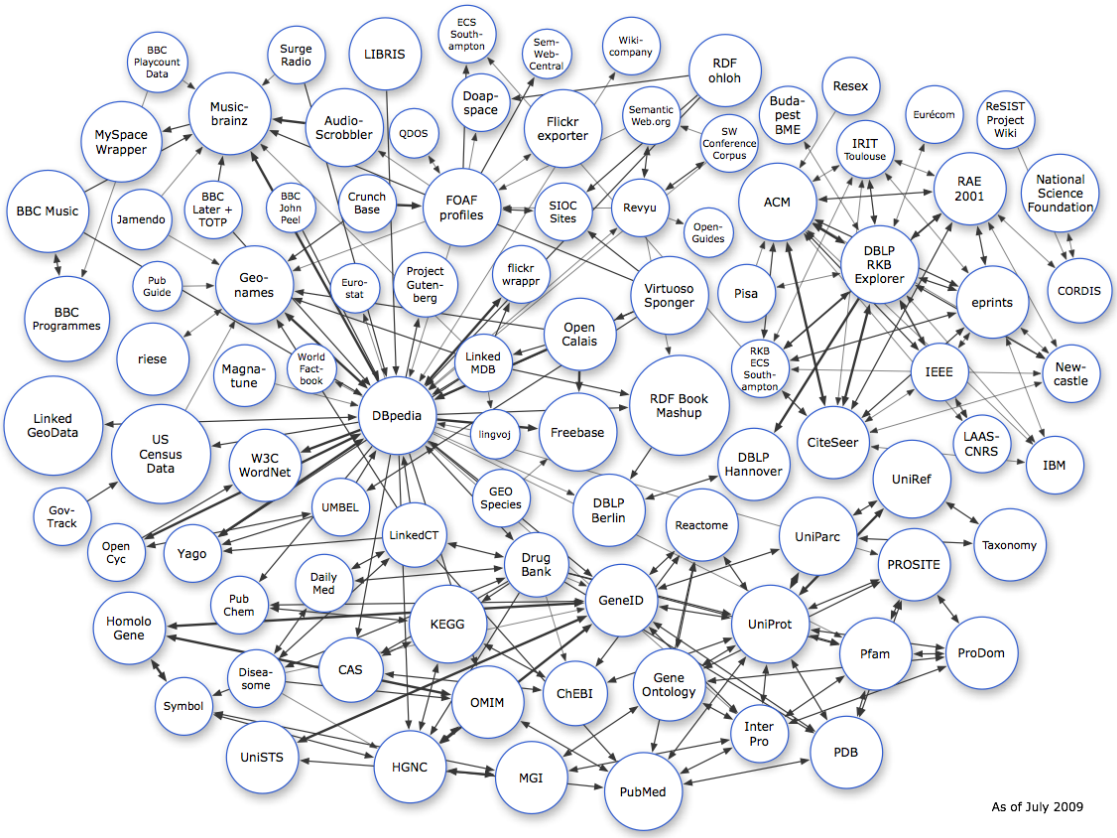

Figure 2.6. Linking Open Data dataset cloud

computer programs. Links between different data sources are stored in standard formats such as RDF-XML. These links may be created automatically or manually. For example, the DBpedia and OpenCyc links to WordNet are created manually. Figure 2.6 shows the linked data sets as of July 2009.

2.3.3. Ontology Matching

In order to improve the tagging quality of Semantic TagPrint, several knowledge bases and ontologies are integrated using ontology matching techniques. Ontology matching approaches can be classified into two in terms of granularity and input interpretation [55]: Element-level and Structure-level.

- Element-level matching approaches process entities without considering relations to other entities. String-based, Language-based and Linguistic resources are the widely used in element-level matching techniques.

i. String-based techniques measure similarity of strings by matching the letters within them.

ii. Language-based techniques measure the similarity of entities by considering entity names as words and process them with NLP techniques.

iii. Linguistic resources use linguistic relations between entities that are used to calculate the similarities between them. Dictionaries and thesauri are examples of linguistic resources [55].

- Structure-level matching approaches consider relations to other entities, which processing. Data analysis, statistical analysis, graph-based and taxonomy-based algorithms are the widely used in structure-level matching techniques.

  i. Data analysis and statistical techniques measure similarity considering the statistical distribution of entities. Learning algorithms are fall into this category.

  ii. Graph-based techniques represent entities as nodes and process given ontologies using graph matching algorithms. Taxonomy-based techniques are also graph-based algorithms but they consider only the parent child relations [55].

2.3.4. Related Work in Reconciling Ontological Knowledge Bases for Semantic Tagging

There are several systems that have attempted to semantify and/or reconcile information from various knowledge bases including WordNet and Wikipedia [56–58].

YAGO [56] uses Wikipedia as a source of information and extracts 14 relationship types, such as *subClassOf*, *familyNameOf*, and *locatedIn* from Wikipedia categories and redirection information. Unlike DBpedia, YAGO does not use the full Wikipedia category hierarchy when extracting *subClassOf* relationships, instead it maps leaf categories to WordNet concepts with a hybrid method based on heuristic rules [50]. YAGO parses Wikipedia categories into tokens and maps the premodifier and the head compound to a WordNet concept, if it is defined. For example, the category *American people in United States* becomes a subclass of WordNet concept *person/human*. In WordNet, a concept usually has multiple meanings. However, YAGO does not perform any Word Sense Disambiguation, that is, it does not map a word to an appropriate meaning according to its context. Instead, it directly maps a term to a concept that has the highest rank among

all of the meanings of the term.

KYLIN [57] utilizes Wikipedia infoboxes, redirects and categories for information retrieval, and it populates missing infoboxes using machine learning algorithms. Unlike YAGO, which extracts only a set of predefined relationship types, KYLIN can learn to extract values for any attribute [57].

KOG [58] is another system that maps Wikipedia infobox classes and attributes extracted from KYLIN to WordNet. KOG uses the Stanford parser [59], a NLP application, to locate the WordNet concept that matches the longest substring of the infobox class name. For example, an infobox class *beach volleyball player* is mapped to *volleyball player* in WordNet, instead of *player*. KOG selects the highest ranked sense like YAGO does. It also uses the manually generated links between DBpedia and WordNet as training data. For each mapping, the learner algorithm generates scores that are used to align precision and recall [58].

2.4. Search Technologies

Web search engines help locating desired information on the Web. The keyword based search engines Google [60] and Yahoo [61] are the most widely used ones.

The performance of a search engine is measured based on two factors: the precision and the recall. Precision and Recall formulas are defined as follows:

$$Precision = \frac{\#\text{ relevant documents}}{\#\text{ documents retrieved}} \tag{2.1}$$

$$Recall = \frac{\#\text{ relevant documents}}{\#\text{ all relevant resources}} \tag{2.2}$$

The precision is defined as the ratio between numbers of documents relevant to user's interest over the number of documents retrieved by the search engine. The recall value is defined as the ratio between numbers of documents relevant to user's interest over all relevant resources that exist. For example, a web user wants to find documents about the programming language *java*. If 80 of returned 100 results from a search engine are about *java* programming language and remaining ones are related with other meanings of the term *java*, precision of this search engine for this search result is 80%.

Modifying queries to increase one of the factors generally cause the other factor to decrease. There are two main approaches to improve the quality of search results [62]. One approach is to use multiple search engines and combine the search results by filtering out the duplicate documents. These kinds of tools are called meta-search engines e.g. AskJeeves [63], MetaCrawler [64] and Clusty [65].

Another approach is to analyze Web content using NLP techniques such as WSD. This approach has two phases. The first phase is query extension, where queries are extended with synonyms of the query term. Then, the meaning of the query term is requested from the user or is extracted using WSD techniques. The first phase also supports natural language queries such as *What is the capital of Turkey*. These kinds of queries are converted into machine readable queries using NLP algorithms. The second phase improves the quality of the response using text analysis. WSD algorithms may be applied to the result set, where the documents that have different contexts than the requested one can be filtered. Hakia [66] and Powerset [67] are examples of NLP based semantic search engines.

In this study, SKMT is proposed as an NLP based semantic search engine to demonstrate potential benefits of Semantic TagPrint.

3. MODEL

This chapter proposes a model for semantic tagging. The model has been realized through the design and implementation of three major sub-systems, which are loosely coupled: an ontology framework (UNIpedia) to serve as a knowledge resource for semantic tagging system (Semantic TagPrint), and a semantic knowledge management platform (SKMT) (Figure 3.1).

- UNIpedia is a platform to address the coverage problem of semantic tagging systems. A term within content can be semantically tagged and retrieved, if it is defined in an ontology. In order to improve the tagging performance of Semantic TagPrint, an ontology framework that combines an ontology with knowledge bases is proposed.
- Semantic TagPrint automatically maps noun phrases of English text documents to ontological entities defined in UNIpedia. Ontological entities are ranked with weights of significance, which are used for recommending as semantic tags.
- SKMT is a user interface to Semantic TagPrint. This tool is an application enables users access to the results of the semantic tagging approach proposed in this thesis.

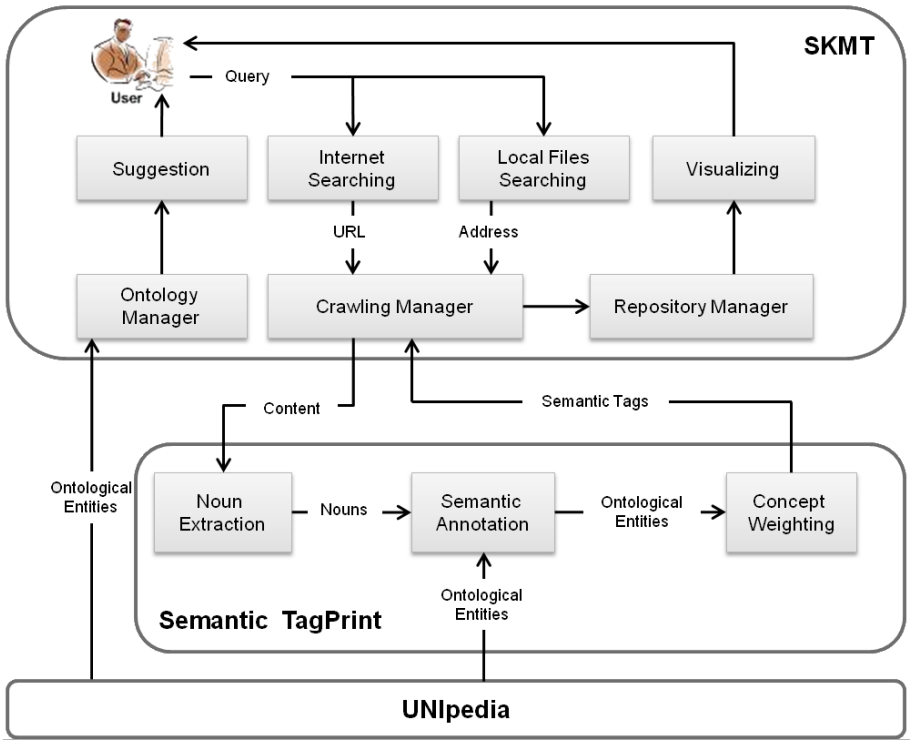

Figure 3.1. Model Architecture

3.1. UNIpedia: A Unified Ontological Knowledge Platform for Semantic Content Tagging and Search

WordNet and Wikipedia as domain-independent knowledge bases are widely used for this purpose. Wordnet is a lexical knowledge base that covers most of the concepts defined in the English language. However, it does not contain most named instances of the concepts it covers – such as of people, organizations, geographic locations, books and songs. Furthermore, many new (contemporary) concepts are not covered, as well. Therefore, many current documents contain terms that are not in Wordnet.

Wikipedia does cover up-to-date content. Thanks to user contributions, its content rapidly evolves and remains up to date, which is very desirable when dealing with current documents, but it lacks formally defined hierarchical relationships among its categories. These relationships are not appropriate for consideration as a taxonomy.

The success of semantic processing clearly depends on the term coverage of the ontologies utilized. In order to effectively process domain-independent documents, a comprehensive, up-to-date, and evolving ontology is required. For this purpose, an ontology framework that combines knowledge bases is proposed. UNIpedia aims to serve as a high quality, comprehensive, up-to-date, domain independent and easily re-usable resource for semantic applications. UNIpedia uses WordNet as its backbone ontology, and maps instances from other knowledge bases to WordNet concepts by introducing an *isA* relationship between them. UNIpedia reconciles the following knowledge bases:

- *WordNet* serves as the backbone ontology to cover concepts defined in the English language. WordNet is chosen because it is reliable and its concepts have single parents forming directed acyclic concept hierarchies, which are easy to process in Semantic Web applications.
- *Wikipedia* is used to instantiate WordNet concepts with named instances. Wikipedia is chosen because it contains reliable, up-to-date and extensive content.
- *OpenCyc* is another knowledge base mapped to WordNet. It is chosen as a knowledge source, because of its comprehensiveness, publicly availability, and reliable content.

3.1.1. System Architecture

UNIpedia has two major modules: Unifier and Indexer (Figure 3.2).

- The *Unifier* is the main module that generates mappings between WordNet and knowledge bases. Currently, Wikipedia and OpenCyc are integrated.
- The *Indexer* module generates the UNIpedia based on the mappings between WordNet and other knowledge bases.

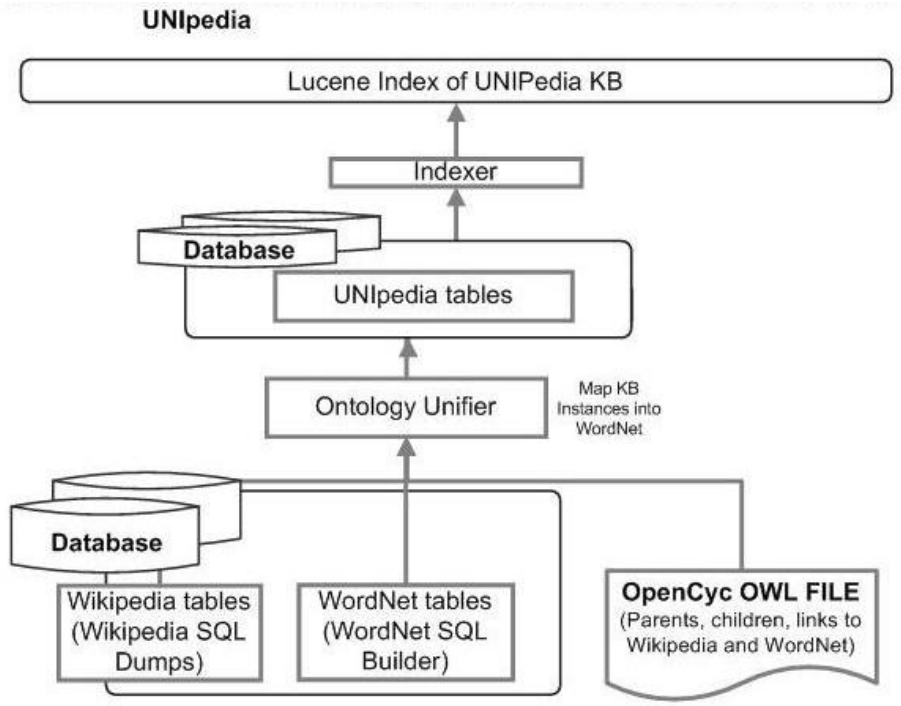

Figure 3.2. UNIpedia System Architecture

3.1.2. Unifier

Unifier takes knowledge bases as its sources, and maps instances from source knowledge bases to WordNet concepts. The mapping process has four steps: *Extraction*, *Alignment*, *Filtering* and *Selection*.

3.1.2.1. Extraction. In this step, Unifier extracts metadata from the source knowledge bases. The extracted metadata is used to instantiate WordNet concepts in the following steps. The *Extraction* step is knowledge base specific and requires specific implementation

for each source knowledge base.

To handle Wikipedia, Unifier takes every Wikipedia article as a named instance, and extracts metadata from its article title, content, categories, infoboxes and redirection links. An instance gets its name from the main article title and its synonyms from redirection links in the article. To properly capture the instance name and context, Unifier represents each instance with its own unique title name in order to handle disambiguation. For example, *Java* has three instances, each of which has a unique name: *Java(coffee)*, *Java (programming language)*, and *Java (island)*. Phrases between parentheses are filtered and will be considered as a candidate parent concept for the instance when naming the instance.

Unifier uses Wikipedia infoboxes (attribute value pairs describing subject of article) as a metadata source for extracting candidate parent concepts for instances. However, not all Wikipedia articles have infobox information. Furthermore, many infobox entities are erroneous with improperly defined abbreviations and spelling errors introduced by users [58]. Thus, Unifier also uses *categories* defined as part of the article as a metadata source. In most cases, categories introduce noise in terms of information extraction. In Wikipedia, there are relational categories that cause noise, which need to be filtered out. Examples include *1879 births* and *1955 deaths*, etc. In the extraction step, Unifier ignores categories named with words from a black list of words: *death, birth, award, people, member, family, figure, year, candidate, alumnus, nominee, winner, alumni, history, issue and disambiguation*. These black list of words are selected for the Wikipedia extraction step based on our justifications after analysing the Wikipedia pages.

A Wikipedia article generally starts with its short definition. Typically the first sentence of an article has a structure like *X is a Y*, where *X* denotes the page title. The first noun phrase *Y* after *is a* is extracted. The first sentence of an article is also used as an instance description in UNIpedia.

OpenCyc has a set of relationships defined between its concepts. Among the relationships, *label*, *aliases*, and *a type of* relationships are utilized in metadata extraction. Unifier uses the *label* relationship to extract names for UNIpedia instances, and extract

synonyms of named instances from the *aliases* relationships. *A type of* relationship is similar to Wikipedia categories and serves as metadata sources for the mapping process. Similarly, to properly handle information from the *a type of* relationships, a blacklist of words are filtered out: *function, topic, slot, type, relation, predicate, microtheory, collection, pit, context, concept, and synset.* These black list of words are selected for the OpenCyc extraction step based on our justifications after analysing the OpenCyc classes.

3.1.2.2. Alignment. In the *Extraction* step, Unifier extracts metadata as candidate parent concepts for instances. *Alignment* uses rule based heuristics to convert these phrases to WordNet concepts.

The extracted metadata are free-form English phrases. The system extracts nouns or noun phrases using prepositions and pronouns as delimiters. For instance, the phrase *woman basketball players in Europe* contains the preposition *in*. The system extracts the words within a phrase that come before the preposition. The phrase is transformed to *woman basketball players*. Unifier has a predefined set of blacklist words for this purpose: *by, who, from, with, for, about, in, at, on, stubs, to, orders, types, parts of, involving.*

After removing the blacklist words and those subsequent to them, Unifier searches for the normalized phrases in WordNet. If a phrase is not defined in WordNet, the system recursively ignores the first word of a phrase and attempts to find the longest matching phrase in WordNet. For example, the phrase *woman basketball players* is not defined in WordNet. Therefore, the system disregards the first word, *woman*, and checks whether the phrase *basketball players* exists. Unifier also checks the singular form of a phrase. Thus, *woman basketball players in Europe* is aligned to the term *basketball player* in WordNet.

Due to the fact that WordNet terms may have several meanings, Unifier must align the terms to the proper WordNet concepts. Unifier tries to exploit existing manually created links. DBPedia and OpenCyc contain manually created links to WordNet concepts. However these links are sparse and out-dated. Approximately %1 of OpenCyc and DBpedia entities are linked to WordNet. Furthermore, *isA* property of instances may change in time. Instead of using these links directly, Unifier uses these existing links between

WordNet ontology and Wikipedia and OpenCyc instances to find the appropriate meaning of a term. To do so, the system combines all manual mappings to create a pool for links. The pool contains a total of 444,910 links and 6,622 distinct WordNet concepts. The mapping module checks whether the mapping pool contains the searched term. If it could not locate the term, it maps the term to the concept that has a higher rank than others. The concept ranks in WordNet are defined based on frequencies of encountering the senses in the semantic concordance texts used in the construction of the database [68]. Pseudo code of the Alignment function is defined in Figure 3.3.

S is a list of concepts derived from the WordNet database
P is list of phrases extracted from knowledge bases
$getRank(i)$ returns sense rank of a phrase i, if it is defined in the links pool else return the highest rank
$concept(i, r)$ returns the corresponding concept for the phrase i with sense rank r
$remove(P, W)$ removes the word from the list P
$P1 \leftarrow normalizePhrase(P)$
for all $W \in P1$ **do**
  if $P1 \in S$ **then**
    Return $concept(i, getRank(i))$
  else
    $P1 \leftarrow remove(P1, W)$
  end if
end for
Return null

Figure 3.3. UNIpedia Alignment algorithm

3.1.2.3. Filtering. The *Alignment* step identifies candidate parent concepts for the instances. These concepts could be syntactically and semantically related. In the *Filtering* step, concepts that are parents of other candidate concepts are filtered. For example, suppose that for the named instance *hamburger* if two candidate parent concepts: *food* and *sandwich* are extracted. In WordNet, the *sandwich* concept is a child of the *food* concept and the *sandwich* concept provides more information about *hamburger* than the *food* concept. Therefore, the *food* concept is filtered from the candidate concept list. Con-

cepts whose names subsume another concept's name are also filtered out. For example, the term *basketball player* subsumes the term *player*. As a consequence, Unifier filters the *player* concept if it exists in a list of candidate parent concepts.

3.1.2.4. Selection. In the *Selection* step, Unifier selects the most appropriate WordNet concept for an instance from within the collection of concepts generated in the previous steps. Then it creates an *isA* relationship between the instance and the chosen WordNet concept. Both statistical and ontological properties of candidate concepts are used to make this selection. Properties such as frequency, Google Popularity (GP) and Normalized Google Popularity (NGP) are statistical. Properties such as Information Content (IC) and Depth (DP) are ontological properties. All of these properties are taken into consideration in the *selection* step.

GP [69] is a statistical value that represents the occurrence probability of an instance name and concept name in the collection of Web pages indexed by The Google search engine. Higher values indicate that two concepts are frequently used together in Web pages and they are more related to each other.

NGP [69] considers occurrence probabilities of instances and concepts separately. NGP is computed with the formula *ngp* (Formula 3.3).

$$gp(i) = \frac{logGHits(i)}{logM} \tag{3.1}$$

$$ngd(i,c) = \frac{\max(logGHits(i), logGHits(c) - logGHits(i,c))}{logM - \min(logGHits(i), logGHits(c))} \tag{3.2}$$

$$ngp(i,c) = 1 - ngd(i,c) \tag{3.3}$$

Where *GHits(i)* is the number of Google hits for the instance name i, *GHits(i,c)* is the number of Google hits for the instance name i and concept name c. M is the total number of Web pages indexed by Google [70].

Information Content values of concepts are derived using the taxonomic structure of WordNet formed by hypernym relations. It measures how much information a concept expresses. Information Content is calculated according to Formula 3.4.

$$ic(c) = 1 - \frac{\log(child(c)+1)}{\log(max_{ont})} \tag{3.4}$$

Where function *child(c)* returns the number of concepts that have common ancestor concepts c. max_{ont} represents the number of concepts that exist in an ontology [71]. The formula returns values between [0-1]: leaf node concepts earn IC values of 1 and the root concept has IC value of 0.

The *depth* of a concept in an ontology taxonomy is another semantic feature. Taxonomy of an ontology is formed by directed edges that have a link from the parent concept to the child. The depth of a concept c is the number of directed edges from the root concept to this concept. Since WordNet concepts have single parents, there is only one path from each WordNet concept to the root concept. The root concept is at depth zero. The expressive power of a concept increases with the depth of definition in the taxonomy.

$$dp(c) = distance(root) \tag{3.5}$$

In the previous steps, metadata was extracted from various sources. The mapping process may map phrases to the same concepts. Thus, a concept may occur several times in the list of candidate parent concept. The frequency of a concept in a candidate list is an important indicator. The algorithm favors concepts with higher frequencies on the assumption that they are more relevant.

The following selection heuristics are based on the statistical and ontological properties of the source:

- Information Content Based Heuristic - *ICH*
- Depth Based Heuristic - *DPH*
- Google Popularity Based Heuristic - *GPH*
- Normalized Google Popularity Based Heuristic - *NGPH*
- Voting between all above - *VOTEH*

ICH selects a candidate concept for a mapping, based on the highest score of candidate concepts using Formula 3.6. Similarly, *DPH* uses the *DP*, *GPH* uses the *GP*, *NGPH* uses the *NGP* scores of candidate concepts multiplied by their frequency of occurrences in the extracted metadata. *VOTEH* votes between the heuristic selections with varying weights. The weights are determined based on their performance during experiments conducted.

$$ich(c) = ic(c) \times \log(frequency(c) + 1) \tag{3.6}$$

Several heuristics have been defined, where each may favor a different candidate concept. Table 3.1 shows the mappings of the top ten Wikipedia pages in 2009 [72]. It can be observed that selection heuristics offer different candidate concepts for the Wikipedia pages.

Table 3.1. Mapping of Top 10 Wikipedia pages using UNIpedia framework

Article Title	**ICH**	**DPH**	**GPH**	**NGPH**	**VOTEH**
The Beatles	musical group	artist	musical group	musical group	musical group
Michael Jackson	soprano	soprano	soprano	soprano	soprano
YouTube	website	website	implementation	website	website
Wikipedia	encyclopedia	encyclopedia	website	website	website
Barack Obama	president	president	nobel laureate	nobel laureate	nobel laureate
United States	republic	country	country	country	country
Facebook	website	website	website	website	website
Swine influenza	infection	infection	infection	infection	infection
Eminem	rapper	rapper	rapper	rapper	rapper
Lost (TV series)	series	series	series	series	series

3.1.3. Indexer

The Indexer integrates the WordNet and knowledge base mappings, and generates the UNIpedia ontology in Lucene [73] index format. UNIpedia is composed of a large number of concepts and instances. To improve the performance of semantic Web tagging and search, the ontology is kept in Lucene inverted index files. Every concept and instance is represented as a Lucene document and the properties of the concept form the fields of the document.

One of the important features of the Indexer is that it can properly handle conflicts caused by duplicate mappings. It is very common that same instance is defined in several

data sources. For example, the instance *physics* is defined in the sources: WordNet, Wikipedia and OpenCyc. The Indexer can filter duplicate instances based on their senses. The Indexer decides whether instances are the same or have different senses upon whether they share the same parent concept. If an instance is mapped to different concepts, all the mappings are included even if they are the same.

Another important feature of the Indexer is that it maintains the sense ranks for concepts. Concepts may have different senses. WordNet maintains sense rank for each concept. UNIpedia maintains the sense ranks for concepts such that WordNet concepts preserve their sense ranks, while Wikipedia and OpenCyc term ranks are expanded to the WordNet concept ranks at a lower level for the shared term. WordNet is given priority because of its greater accuracy.

3.2. Semantic TagPrint - Tagging and Indexing Content for Semantic Search and Content Management

By adopting an ontology to cover most named concepts and instances, Semantic TagPrint is conceived. It maps a text document to semantic tags that are defined as entities in an ontology. It uses a linear time lexical chaining WSD algorithm for the mapping process. The lexical chaining algorithm disambiguates terms based on several semantic features. After the mapping process, statistical and semantic features are utilized for weighting the significance of the semantic tags. Semantic tags are sorted and recommended to users by their significance.

Figure 3.4 shows an example of how a piece of textual content is mapped into semantic tags defined in UNIpedia [74]. The tree representation of the semantic are generated using the Semantic TagPrint system. The underlined words in the textual content are the identified terms by Semantic TagPrint which would be annotated with UNIpedia ontological entities and recommended as semantic tags. The semantic tags are represented with the black nodes. The numbers next to the semantic tag names are assigned weights by Semantic TagPrint, which states their significance in the content. Ontological concept classes are represented as circles, while concept instances are shown as rectangles. White nodes are ancestor concepts of the semantic tags.

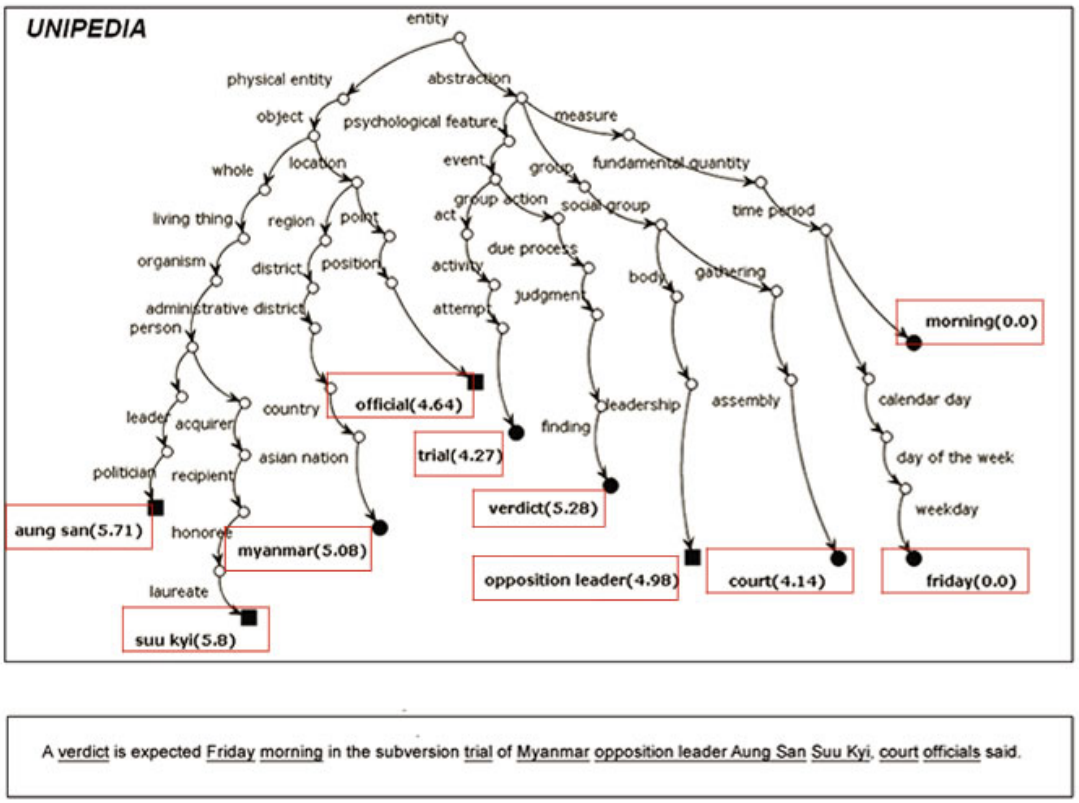

Figure 3.4. Semantic tagging of textual content from an CNN article

3.2.1. System Architecture

Semantic TagPrint is composed of three major modules: UNIpedia, Semantic Tagging Engine and User Interface (Figure 3.5). The Semantic Tagging Engine analyzes a given text document and assigns weighted ontological entities. The User Interface is used for examining the tags and tuning the parameters and visualizations of weighted semantic tags.

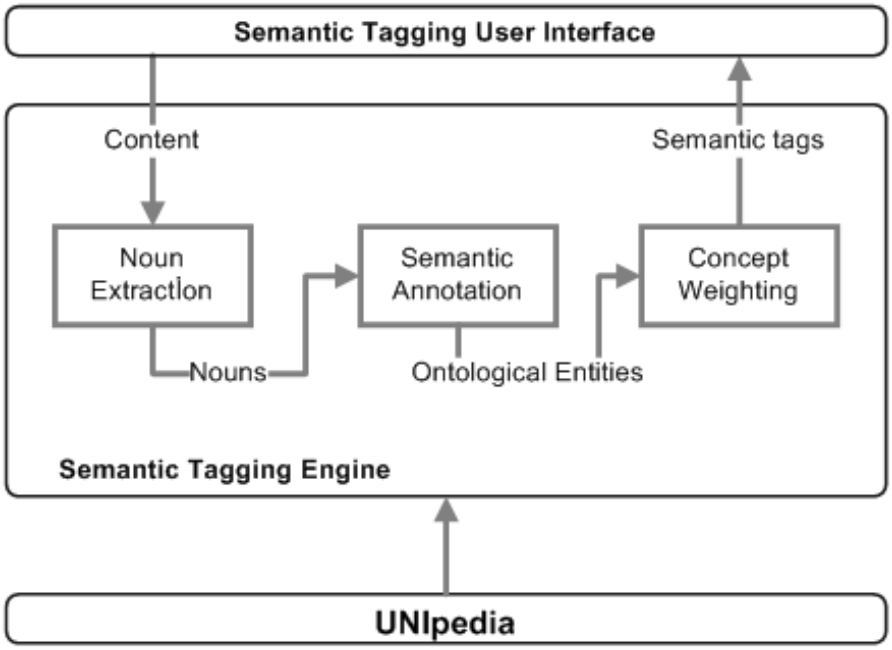

Figure 3.5. Semantic TagPrint System Architecture

With the interface shown in Figure 3.6, input parameters can be modified and the generated output can be viewed in 2D or 3D semantically connected graphs. Unlike other supervised or unsupervised WSD algorithms, knowledge based WSD algorithms have the advantage that Semantic TagPrint users can see why a sense is selected in a given context through such an interface and use this information to debug the algorithms and tune the parameters.

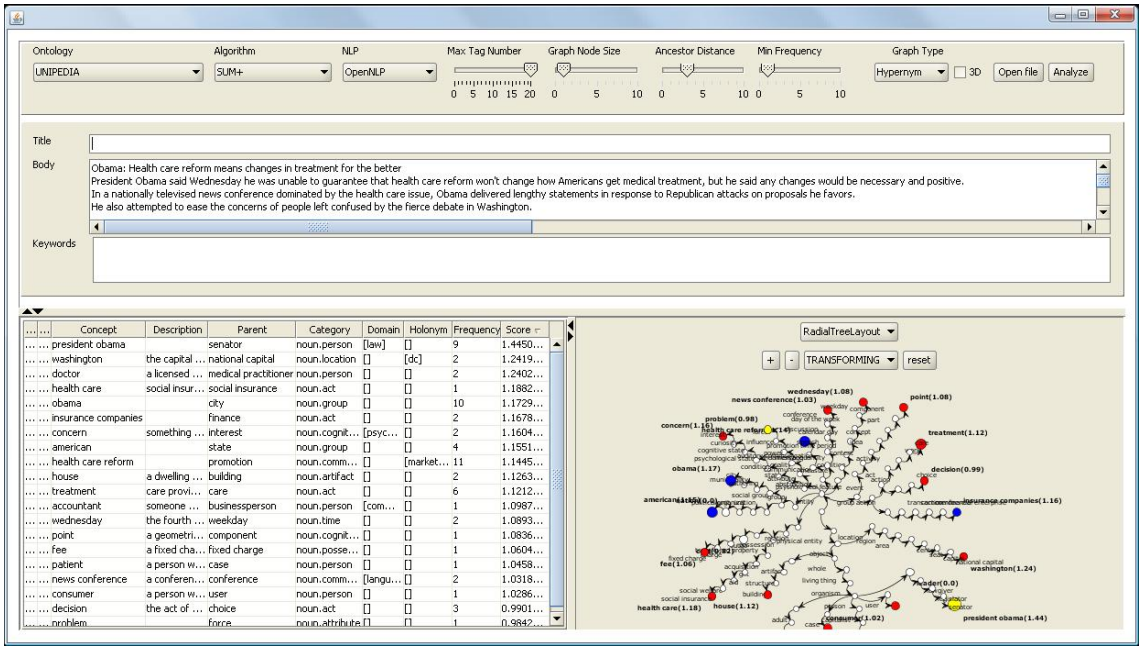

Figure 3.6. Semantic TagPrint User Interface

The semantic tagging engine has three modules.

- *Noun phrase extraction* module parses the raw input text and extracts noun phrases.
- *Semantic Annotation* module maps noun phrases to UNIpedia concepts including both concept classes and their instances.
- *Concept Weighting* module uses statistical and semantic features to weigh the significance of ontological entities and recommend these entities as semantic tags.

In the following sections, the semantic tagging engine's modules will be analyzed comprehensively.

3.2.2. Noun Phrase Extraction

Content of a document is usually best described by noun phrases. Noun phrase extraction is a key step and it directly affects the performance of a semantic tagging system. Semantic TagPrint uses an NLP program called OpenNLP [75] to extract consecutive noun tokens to form noun phrases sorted by their occurrence frequency. Pseudo code of the noun extraction function is defined in Figure 3.7.

```
C: text
P: list
L: list
C ← preProcess(C)
S ← sentenceDetection(C)
for i = 0 to length(S) do
  W ← tokenizing(S_i)
  T ← tagging(W)
  for j = 0 to length(W) do
    if T_j =Noun or W_j − 1 =The then
      append(W_j, L)
    else
      append(L, P)
      clear(L)
    end if
  end for
end for
ReturnP
```

Figure 3.7. Semantic TagPrint Noun Phrase Extraction algorithm

Sentence detection, tokenizing and tagging are functions provided in OpenNLP. Functionalities of these functions are explained below in detail:

- *Sentence detection* function splits the given content into sentences. Sentence detection is harder than it may appear. While sentences end with symbols like the period and the question mark, these symbols do not necessarily terminate sentences. The presence of abbreviations and numbers that include such characters complicates sentence detection. For example, consider the following sentence: *Youtube.com is a video sharing website.* The website name includes a period which does not end the sentence.
- *Tokenizing* function splits a sentence into tokens (words). Tokenizing cannot be simply handled by detection of the space character. A tokenizer is required to split words that consist of contractions (i.e. *doesn't*).
- *Tagging* function labels the tokens with the parts of speech such as noun, verb,

adverb, etc. Consecutive noun tokens, which form noun phrases, are extracted with their occurrence frequency.

3.2.3. Semantic Annotation

The Semantic Annotation process has two steps: Phrase mapping and Sense mapping.

3.2.3.1. Phrase Mapping. Noun phrases are mapped to UNIpedia to retrieve their corresponding UNIpedia terms. As described in Figure 3.8, if there is an exact match between the noun phrases and a UNIpedia term, the term is returned as the result. Otherwise if the noun phrase is composed of multiple tokens of words, the system generates two sub-phrases by removing the first and last word from the noun phrase. These two noun phrases replace the original noun phrase and repeat the phrase mapping process until there is a match or there are no more tokens left. The algorithm favors the longer and the right side phrase in case there is a tie. Consider the phrase *semantic web tool*. It is not defined in UNIpedia, thus it is divided into two sub-phrases which are *web tool* (right side) and *semantic web* (left side). Semantic Web is defined unlike Web tool. Therefore, the phrase mapping algorithm maps *semantic web tool* phrase to *semantic web* phrase.

```
P: list
if queryOntology(P) > 0 then
  Return P
else if length(P) = 1 then
  Return NULL
else
  P_r ← phraseMapping(removeFirstToken(P))
  P_l ← phraseMapping(removeLastToken(P))
  if length(P_r) ≥ length(P_l) then
    Return P_r
  else
    Return P_l
  end if
end if
```

Figure 3.8. Semantic TagPrint Phrase Mapping algorithm

In certain cases, plurality causes mapping problems. For instance, *Siemens* is a company and it would be a mistake if it is stemmed into *Siemen*. To address this issue, Phrase Mapping algorithm queries both the original and the stemmed phrase. If both are defined in UNIpedia, the original one is selected.

3.2.3.2. Sense Mapping. A UNIpedia term may have several senses, each of which is a UNIpedia concept. These polysemous terms get different meanings according to the context. UNIpedia terms from the previous step can't be mapped to a UNIpedia concept directly, if the terms are polysemous. To map them to the right UNIpedia concepts, a Lexical chaining algorithm is developed with different semantic features to overcome the WSD problem. Moreover, a thorough analysis is conducted on the algorithms, and new algorithms are proposed to improve the mapping accuracy.

A lexical chain is a set of noun phrases connected by semantic relations between each

other. Each sense of a noun phrase may connect to a different lexical chain. The sense of a noun phrase, which is connected to the longest lexical chain would be the selected meaning for the noun phrase.

Forming the lexical chains by comparing each sense between each other to works in a polynomial time. This thesis aims to tag documents in a real time, therefore a lexical chaining algorithm that works in a linear time O(n) is proposed. The proposed algorithm connects each sense to their parent concepts. In this way, senses are connected to each other through common ancestors.

The lexical chaining algorithms exploit these semantic properties: hypernym, domain, category and holonym. For each property, the lexical chaining algorithm generates a set of lexical chains. With four different sets of lexical chains, how each property contributes to the final WSD could be observed and explored for further improvement.

Hypernym: The Hypernym algorithm takes a set of noun phrases and corresponding UNIpedia senses as an input, and generates lexical chains. The algorithm has two phases. Firstly, each sense has an initial score of zero. Once it is connected, it increases scores of their ancestor senses until reaching a specified threshold of ancestor distance. Senses also increase their scores to contribute scores of their child senses. Senses are scored using the following formula in the first phase:

$$hypernym_{phase1}(c) = \sum_{n=1}^{N} \frac{1}{distance(c, child_{n_c}) + 1} \tag{3.7}$$

Figure 3.9 shows the lexical chains and calculated scores using the hypernym algorithm with maximum ancestor distance 3 and *programming language*, *java* and *prolog* as term input from WordNet. The term *java* has three meanings, *prolog* and *programming language* each has one corresponding meanings.

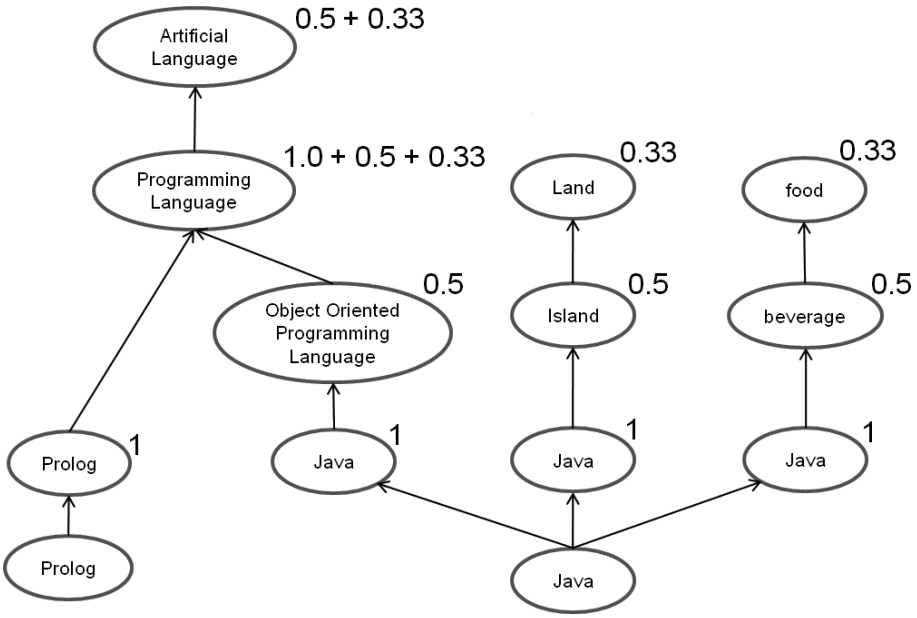

Figure 3.9. Hypernym based lexical chain for the terms *java* and *prolog* after the first phase

In the second phase, senses increment their scores by adding scores of their ancestor senses. In this way, senses that have more common ancestor with other senses get higher scores. For example the senses *java* and *prolog* have a common ancestor sense *programming language*. Both senses increments score of the sense *programming language* in the first phase and then they both benefit the added scores in the second phase.

$$hypernym_{phase2}(c) = \sum_{n=1}^{M} w_n - w_{nc} + synonym(c) \tag{3.8}$$

Where $hypernym_{phase2}(c)$ calculates the weight of a given concept c, N is the number of child senses passing through the context of the concept c, $hypernym(c)$ is the hypernym score for a given concept c, w_{nc} is the hypernym score of the nth ancestor of the concept

c, w_{nc} is the score added by c and concepts sharing the same name to the nth ancestor, M equals to the maximum ancestor distance constant and $synonym(c)$ is c's synonym number.

Considering that senses derived from the same phrase may have different parent senses and these parent senses share common ancestor, lexical chains may end up being composed of senses derived from the same phrase, although they aren't supposed to appear in the same lexical chain. To solve this issue, scores added by sense itself and senses that have common sense names are subtracted. In addition, one sense can be derived from multiple phrases. In these cases, a sense earns an extra score of number of its synonyms.

Senses which have a higher number of connected senses in the lexical chains, get higher score and these senses form the context. Therefore, the Hypernym algorithm maps a phrase to the sense which gets the highest score among the phrase's senses. When two senses of a phrase get equal scores, the algorithm can not disambiguate and maps the phrase to one of the senses randomly.

Figure 3.10 shows the sense selection for the term *prolog* after the second phase. The sense *prolog* gets score 1.83. Since the term *prolog* has one meaning, the sense *prolog* would be selected even, it gets a zero score from the Hypernym algorithm.

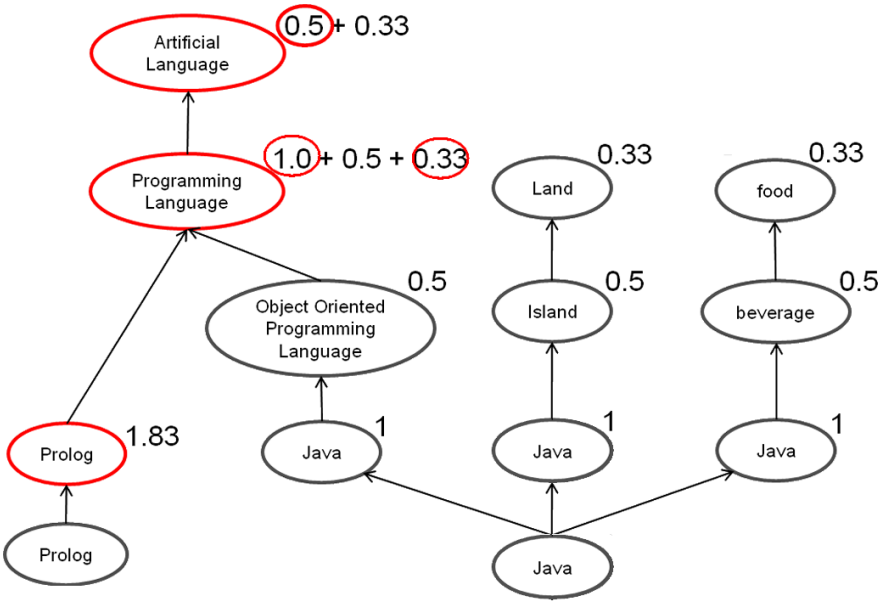

Figure 3.10. Hypernym based lexical chain sense selection for the term *prolog* after the second phase

Figure 3.10 shows the sense selection for the term *java* after the second phase. The sense *java (object oriented programming language)* has a common ancestor with the sense *prolog* and gets score 1.50. The other meanings of the term *java* do not have a common ancestor therefore, they get zero scores. Since, the sense *java (object oriented programming language)* gets higher score than the other meanings of the term *java*, it is selected in the sense mapping step.

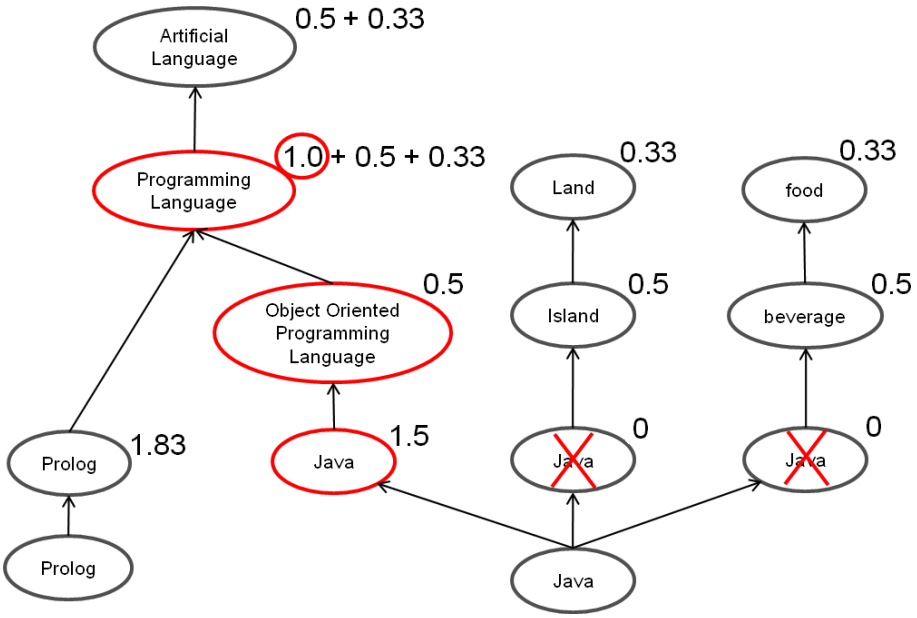

Figure 3.11. Hypernym based lexical chain sense selection for the term *java* after the second phase

Domain: Domain algorithm creates lexical chains using domain property of senses. In this algorithm, senses are represented with their domain senses. Domain senses are connected with each other and scored according to the hypernym property. Senses get the score of its domain sense. If a sense itself is a domain concept, then it gets its own score. The Domain algorithm maps a noun phrase to a sense that gets the highest score among all senses of the noun phrase. For example, both the third meaning of *Java* and the first meaning of *Ontology* terms have domain values the concept *Computer Science*. Thus, they are connected to the same lexical chain. Therefore, the Domain algorithm maps *Java* to its third and *Ontology* to its first meanings.

Holonym: In holonym based lexical chaining approach, senses are connected if holonym relationship exists between them. Senses are also connected if they have a common holonym property. For example, both *car* and *bus* concepts have the concept *wheel* as a part. Thus, *bus* and *car* are connected to the same lexical chain. The Holonym algorithm only considers the first degree holonym relationships. For example, the concept *molecule* has parts of the concept *atom* and *atom* has parts of the concept *nucleus*. *Nucleus* and *molecule* concepts would not be connected in a lexical chain because they don't have a direct holonym relationship. Senses in a lexical chain get scores of the number of

senses in the chain.

Category: Every concept has a category in UNIpedia. Such categories are food, action, person, etc. Senses from the same category gets connected and forms a lexical chain.

3.2.3.3. Improved Sense Mapping algorithms. Each lexical chaining algorithm has its advantages and disadvantages. To improve the sense mapping results, seven different algorithms are implemented for mapping a noun phrase to one of its meanings. These algorithms use the scores calculated during different lexical chaining processes and normalize these scores before the mapping process. In addition to previously described four algorithms, three additional algorithms are implemented: Sense Rank, SUM and SUM+.

Rank: Senses are ranked in WordNet. The rank algorithm simply maps phrases to the highest ranked sense from all its meanings. Also, senses are scored using the following formula:

$$rank(c) = \frac{1}{r_c} \tag{3.9}$$

Where the function $rank(c)$ is the rank score for a given concept c and r_c is the sense rank of c.

SUM: SUM algorithm is the weighted summation of normalized scores from four algorithms: Hypernym, Domain, Holonym and Category. Weights (coefficients) are calculated based on the performance of each algorithm. The formula of the SUM algorithm is defined below:

$$sum(c) = W_{hyper} \times hypernym(c) + W_d \times domain(c) + W_{holo} \times holonym(c) + W_c \times category(c) \quad (3.10)$$

SUM+: SUM+ algorithm uses statistical data of the sense rank in addition to the ontological properties. As defined in the below formula, it is the summation of normalized scores from Holonym, Domain, Hypernym, Category and Sense Rank algorithms with different weights.

$$sumplus(c) = sum(c) + W_{rank} \times rank(c) \quad (3.11)$$

3.2.4. Concept Weighting

Semantic TagPrint system uses statistical and ontological features to weigh the significance of ontological entities and recommends these entities as semantic tags. These ontological features include lexical chaining scores derived from the WSD phrase, Depth value, and Information Content (IC) values of a concept. Statistical features are Term Frequency (TF) and Inverse Google Popularity (IGP) values. By weighting and combing score values from these features, Semantic TagPrint weighs the significance of a concept using the below formula:

$$weight(c) = w_1 \times hypernym(c) + w_2 \times domain(c) + w_3 \times holonym(c) + w_4 \times depth(c) + w_5 \times ic(c) + w_6 \times tf(c) + w_7 \times igp(c) \quad (3.12)$$

IGP, Depth and IC values of concepts are used to favor specific concepts, meanwhile TF statistical information is used to favor frequently used concept names. Note that the scores derived from Category lexical chaining are not used, because category concepts are too general concepts and they would not give too much information about the term significance. Instead, this feature is used to filter specific tags of certain categories. Semantic tags are filtered, which are defined under the *attribute*, *time* and *quantity* categories.

3.3. SKMT - Semantic Knowledge Management Tool for searching, analyzing, and managing content

Semantic Knowledge Management Tool (SKMT) is proposed as a platform to search, analyze and manage content. SKMT provides a user accessible platform for Semantic TagPrint. SKMT uses Semantic TagPrint to generate metadata for given documents. Documents are indexed with their metadata, which enables both semantic and conventional keyword based search. SKMT supports indexing and searching on both local and web resources.

SKMT shows search results visually in a connected graph. Documents are represented as nodes in the graph and nodes are connected to each other with search terms. Documents, which contain the same search keywords, are clustered together. This feature enhances content findability and gives valuable information about the documents' relations between each other.

Tag cloud, a visual illustration of tags, is another feature supported in SKMT. Significant terms of a document can be determined and indexed in SKMT. These significant terms are shown as a tag cloud for a document or for a collection of documents.

3.3.1. Semantic Search

Semantic search works based on the meanings of search and document terms. In order to realize semantic search, both search terms' and document terms' meanings are

required. SKMT takes search terms' meanings directly from the system users using an auto complete search field. When a user starts typing into the auto complete field, the suggestion module of SKMT suggests phrases that start with a given input from the vocabulary of a selected ontology. Suggestions are ontological entities, which compose of entities' names with their parents' names. Then a system user selects one of the term's meanings to start semantic search process. Figure 3.12 shows list defined meanings of the term *capital* in UNIpedia.

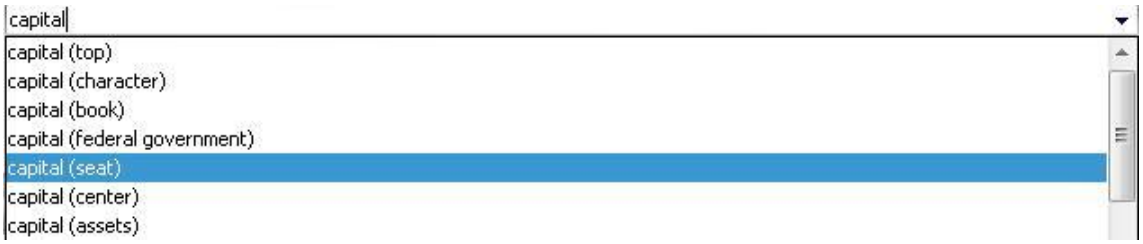

Figure 3.12. SKMT auto complete field lists defined meanings of the term *capital* in UNIpedia

Meanings of terms within a document are extracted using Semantic TagPrint in SKMT. Semantic TagPrint applies WSD algorithms to extract meanings of terms.

Semantic search improves keyword search by removing keywords dependency by handling synonymous (having same meaning) and polysemous (having different meanings) terms and supporting the search utilizing semantic relations between terms.

Synonyms of a search term would be mapped to the same ontological entities in UNIpedia by Semantic TagPrint, thus documents containing synonyms of a search term would be returned in the semantic search result. Figure 3.13 shows the search results of a semantic search and keyword based search of *Barack Obama* in the CNN test data set.

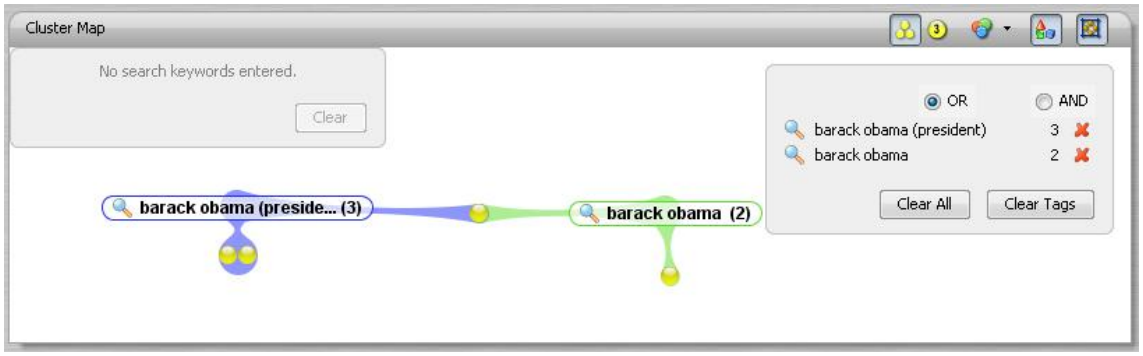

Figure 3.13. Semantic and keyword based search of the word *Barack Obama* in the CNN data set using SKMT

The semantic search recall score is higher than the score of the keyword based search for the search results in Figure 3.13. There are three documents returned for the semantic search and two for the keyword based search. There are three document clusters in the graph. The left cluster contains two documents and these documents contain in their content only synonyms of *Barack Obama* such as *President Obama* and *Obama*. The middle cluster contains one document and this document contains the term *Barack Obama* and it is mapped to the *president* meaning. Therefore, it is a common document for both search queries. The right cluster contains one document and this document contains the term *Barack Obama*, but it is mapped to the *senator* meaning.

Users would be interested in only a specific meaning of a term when they are searching for it. Documents that contain terms that have other meanings than this intended meaning would be filtered in a semantic search. To demonstrate this feature of the semantic search, *apple* and *Apple Inc.* pages of English Wikipedia are indexed using SKMT. Then *apple (edible fruit)*, *apple (company)*, and *apple* are searched in the indexed documents. Figure 3.14 shows the search results.

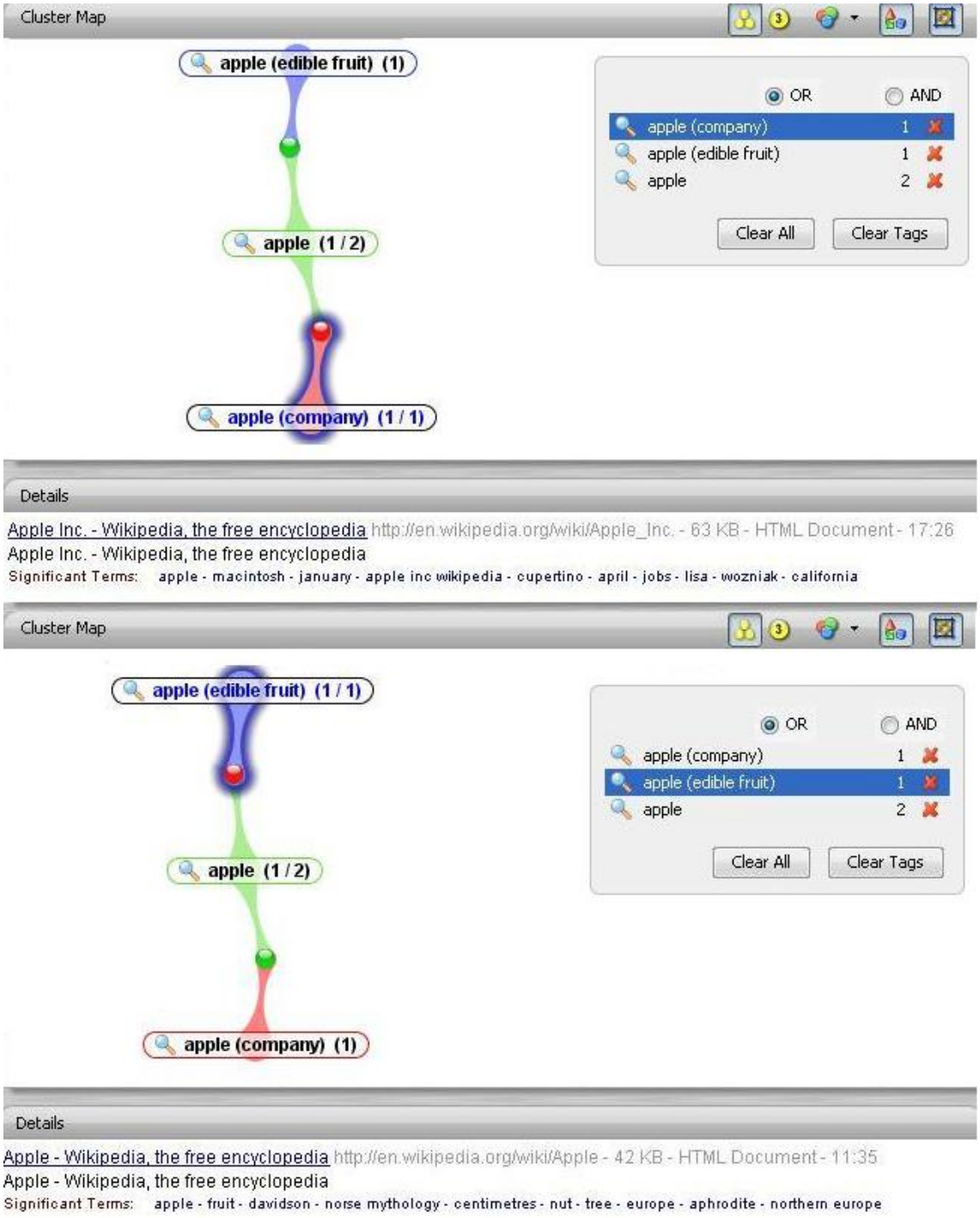

Figure 3.14. Semantic and keyword based search of the word *apple* in Wikipedia using SKMT

The semantic search precision score is higher than the keyword based search's score for these search results. There are two documents in the graph in Figure 3.14. One document contains *Apple (edible fruit)* and the other one contains *apple (company)* in the semantic search results. All documents contain the term *apple* in the keyword based

search result. When a user is interested in only one of the term's meanings, irrelevant documents would be returned in a keyword based search.

Semantic search also provides searching capability using parent and child relations between semantic tags. For example, a user can search news articles related to capital cities of countries. Documents, which contain instances of capital cities would be returned in the semantic search results, even if the searched term doesn't appear in their content. Figure 3.14 shows the search results for the semantic and keyword based search of *capital* in the CNN news data.

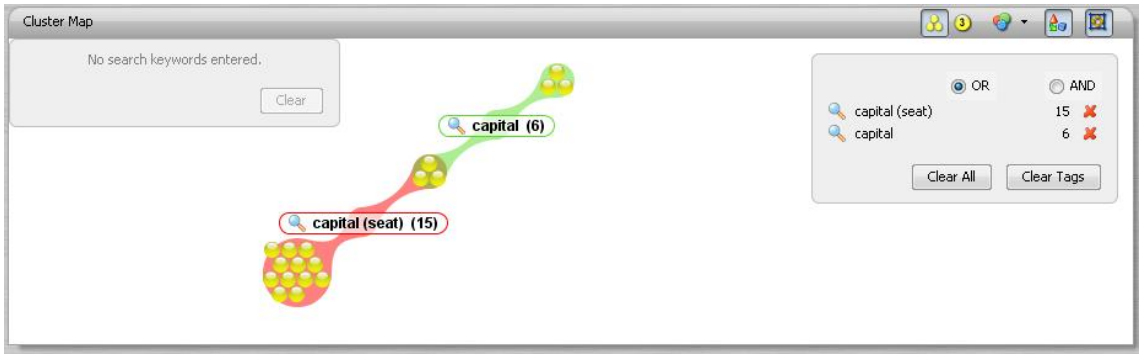

Figure 3.15. Semantic search using parent and child semantic properties of the concept *capital(seat)* using SKMT

The semantic search returned fifteen documents. In contrast, keyword search returned six documents, clearly, the score of the semantic search is significantly better than the keyword search for the search results. Moreover, the documents returned by the keyword search contains the term capital. However, the term is used in its *finance* meaning. Therefore, precision score of the semantic search is also higher than the keyword search.

3.3.2. Semantic Listing in a Taxonomy

Semantic tags are ontological entities and they are defined in certain places in a taxonomy of an ontology. These kind of tags can be represented in a tree format or in a list based on their occurrence frequencies. Showing in a tree gives extra information about terms (ancestor concepts and synonyms). However, locating a specific concept is harder than locating the same concept in a list, because a user has to know the exact position of the concept in the ontology's taxonomy.

SKMT uses the taxonomy of WordNet ontology. This taxonomy is complicated for an ordinary user. Therefore, a user friendly approach is required in SKMT to list semantic tags. Figure 3.16 shows significant terms and semantic tags for the CNN news data. The right side of Semantic Keywords shows semantic tags for some selected documents.

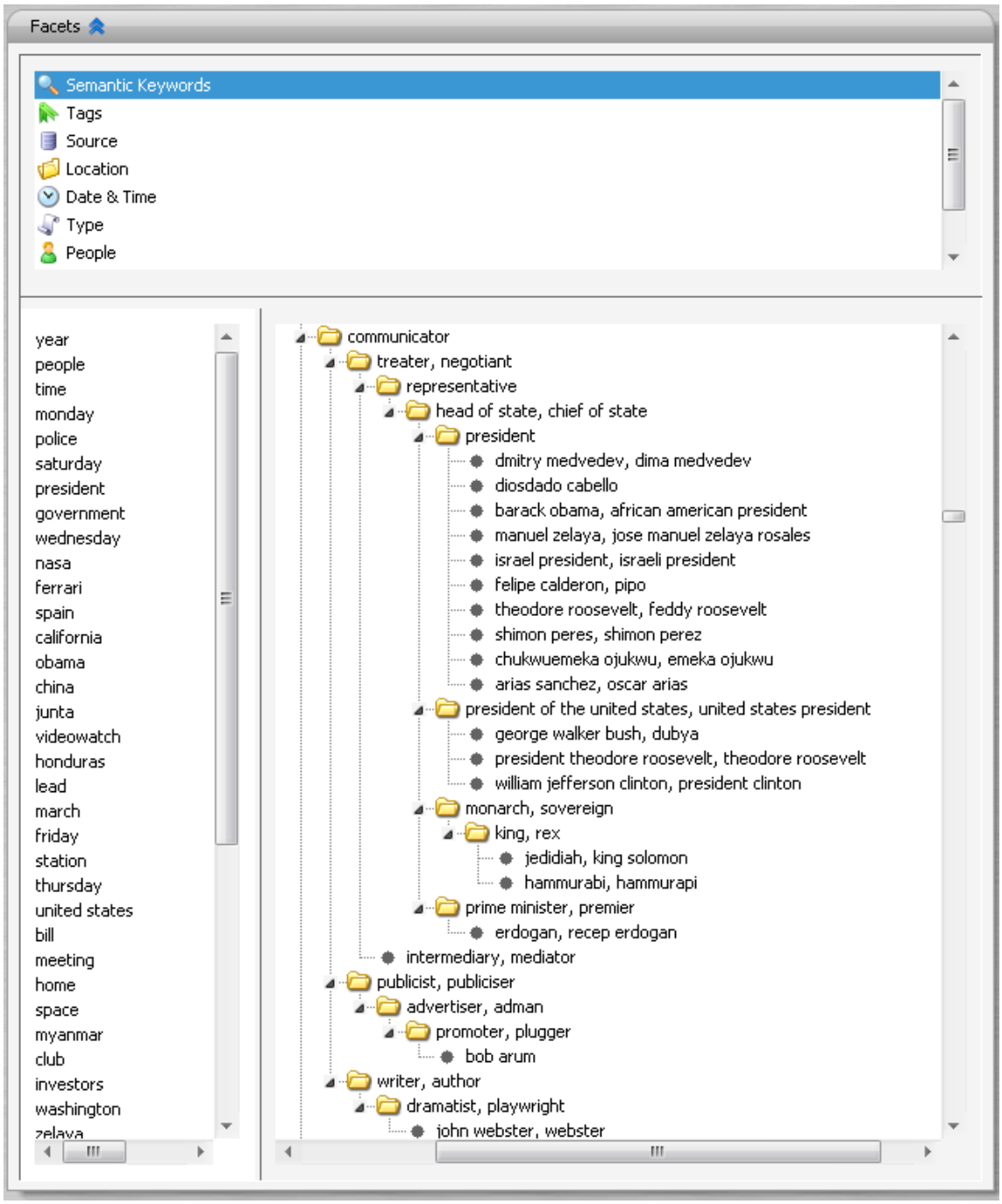

Figure 3.16. Tree representation of semantic tags in SKMT

3.3.3. Semantic Tag Cloud

The semantic tag cloud feature of SKMT shows frequently used top N semantic tags of indexed documents extracted by Semantic TagPrint as a tag cloud. A semantic tag cloud shows valuable information about the content of the documents and the relations between them. Users can easily get brief information about the documents without reading them. Figure 3.17 shows the semantic tag cloud for the documents which are tagged with the concept *Barack Obama (president)* in the CNN test data set.

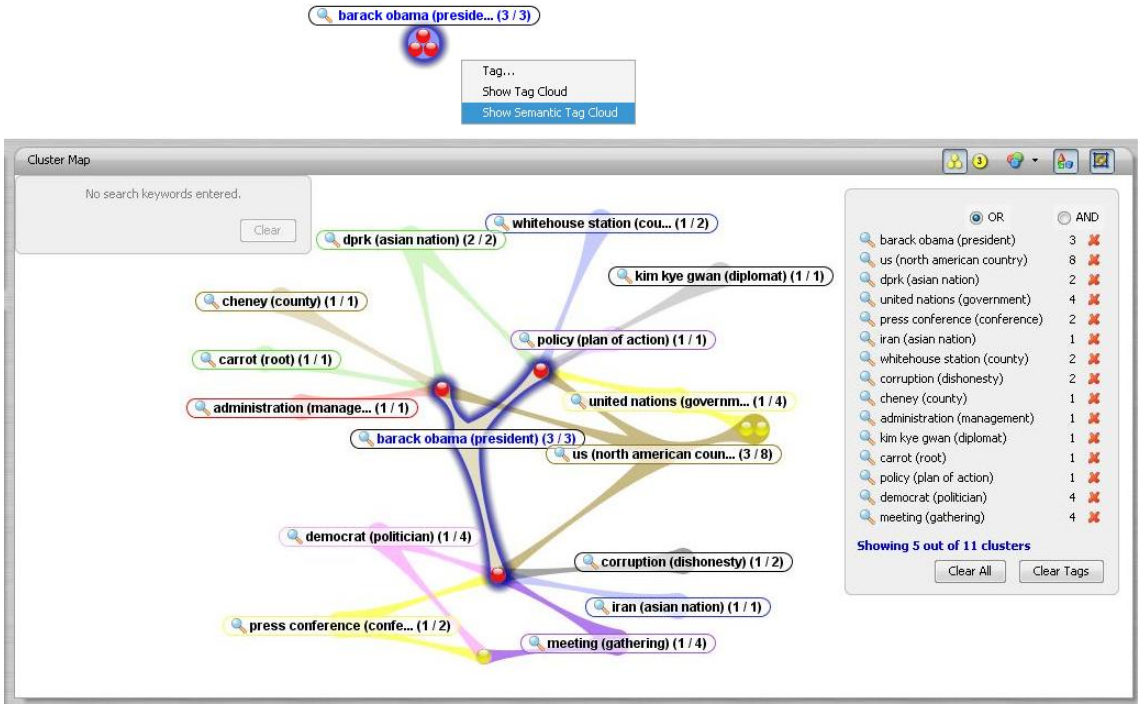

Figure 3.17. Semantic tag cloud of documents tagged with the concept *Barack Obama (president)*

4. IMPLEMENTATION

This chapter first lists the technologies used for the proposed model in chapter 3. Then implementation details of the three main components (UNIpedia, Semantic TagPrint, and SKMT) of the model are presented.

4.1. Technology

Several publicly available tools and technologies are used to implement the semantic tagging model. The model proposed in this study is implemented using the Java programming language. The presented approach uses NetBeans [76] integrated development environment (IDE) to design the graphical user interface (GUI) of Semantic TagPrint, Eclipse [77] as the software development tool for implementing the algorithms of the model, PostgreSQL [78] to store the knowledge bases, Apache Lucene [73] to index and search UNIpedia and documents, Google as a search engine to retrieve statistics about terms (see formulas 3.1 and 3.3) and enable web search in SKMT. Finally, OpenNLP is used as an NLP program to extract consecutive noun tokens to form noun phrases in Semantic TagPrint.

4.2. UNIpedia Implementation

UNIpedia has two major modules: Unifier and Indexer.

The Unifier is the main module that maintains the information knowledge base sources and generates mappings between WordNet and the knowledge bases. UNIpedia is currently integrated with the information sources WordNet, Wikipedia and OpenCyc.

- Database version of WordNet 3.0 is used as the WordNet source in UNIpedia. This version of WordNet is provided in the Web site [79].
- OpenCyc is distributed as an OWL file. The Unifier module of UNIpedia parses the OWL verison of OpenCyc using the Jena API [80] to extract the desired metadata.
- Wikipedia is distributed as SQL and XML dumps. UNIpedia used the SQL dumps

of Wikipedia from the website [81]. The SQL dumps are large SQL files with approximately 2,800,000 articles. The Unifier module of UNIpedia reads the SQL files and inserts them into the database. In order to simplify maintainance of UNIpedia, only Wikipedia pages table is stored in the database. Other information related with Wikipedia such as category and infobox are queried in real time using the Wikipedia Web services [82].

After the data preparetion step, Unifier generates mappings and stores them in the database. PostgreSQL [78] is used as a database in UNIpedia.

Once the database is ready, the Indexer module integrates the WordNet and the knowledge base mappings, and generates the UNIpedia ontology in Lucene index format. Firstly, WordNet nouns are indexed in Lucene format by querying the WordNet tables. Afterwards, the knowledge base mappings are queried and added to the WordNet index. Next, the UNIpedia ontology is created.

UNIpedia currently composes of 2,520,622 word sense pairs, each represented as a Lucene document. Semantic and statistical properties of UNIpedia entities are represented as fields in a structure of the Lucene document. There are totally 14 fields defined for a document. These features can be seen in Figure 4.1 [83].

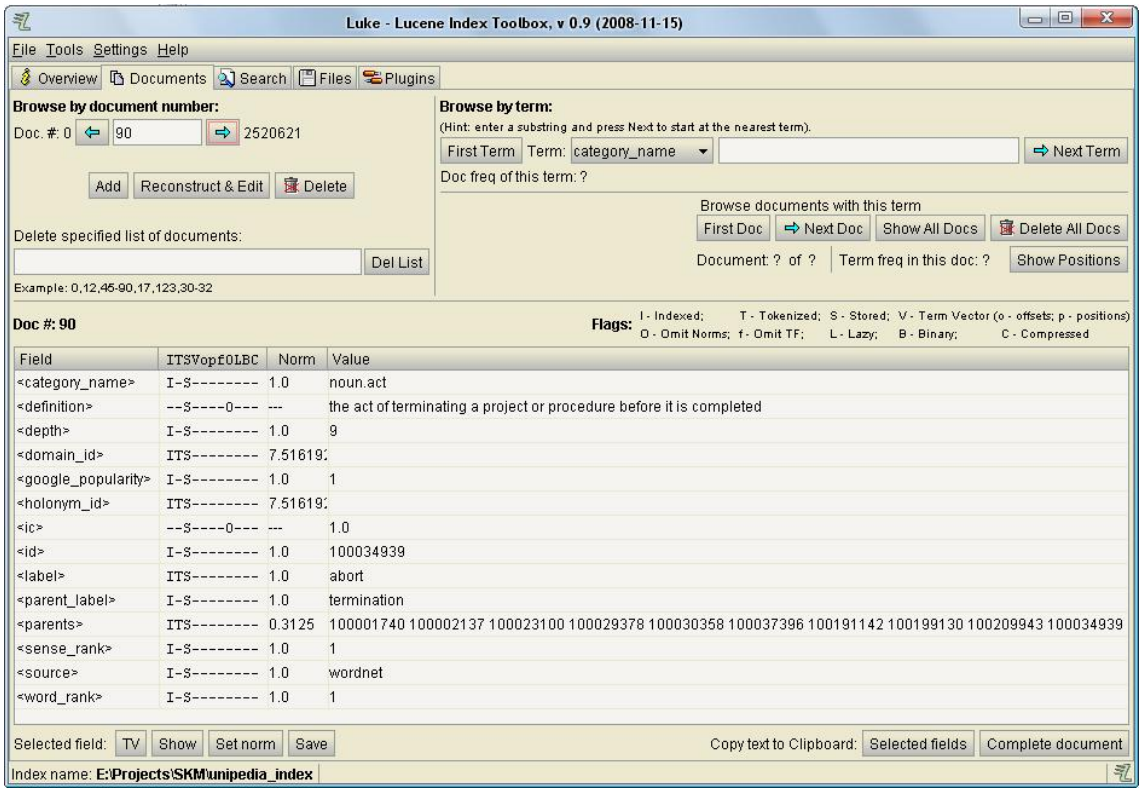

Figure 4.1. Luke (a tool for querying Lucene index files) shows the UNIpedia Lucene index

4.3. Semantic TagPrint Implementation

Semantic TagPrint is composed of three major modules: UNIpedia, Semantic Tagging Engine and User Interface. The Semantic Tagging Engine module is implemented using standard data structures such as hashmaps, hashsets, arrays, etc. Implementation details of the user interface is provided in the following section.

The user interface is designed and implemented for debugging the algorithms and tuning the parameters. Figures 3.6 and 4.2 show the user interface that visualizes a semantic tree of tags for a given text in 2D and 3D.

The middle zone is the text input zone with three text boxes: title, body, and keywords. These three fields form the content input for semantic tagging. Phrases from the title or the keywords have privileges over the text body. These phrases are not subject to the minimum occurrence limitation.

The top zone is the configuration zone that enables various parameter settings. To achieve better tagging results, different algorithms can be tried with different parameter settings. As shown in Figure 3.6, alternating ontologies, mapping algorithms or NLP methods can be chosen. In addition, keyphrase number, maximum ancestor distance, minimum occurrence frequency, node size, graph layouts, and graph dimension can be configured.

- Ontologies: Semantic TagPrint system is currently integrated with two ontologies: WordNet and UNIpedia. The system can be extended with new ontologies after converting ontologies into our knowledge representation model in Lucene index files.
- Algorithm: Algorithm selection determines which algorithms would be used in the sense mapping phase. As described in the previous sections, currently there are seven implemented algorithms: Category, Hypernym, Holonym, Domain, Sense Rank, Sum and Sum Plus.
- NLP: To extract noun phrases from the given text, OpenNLP and Minipar NLP programs are integrated into Semantic TagPrint.
- Max. Tag Number: Keyphrase number specifies the number of extracted semantic tags from the input text. The semantic tags are delimited to effectively analyze the generated semantic graph.
- Ancestor Distance: Maximum ancestor distance determines the maximum ancestor distance between two senses to form a chain between them.
- Min. Frequency: Minimum occurrence frequency is used to filter phrases, which are encountered below a specified value for a given content. Minimum occurrence frequency can be calculated automatically in Semantic TagPrint using the below formula,

$$minf = \lfloor \frac{phraseNum * \log 10(phraseNum)}{uniquePhraseNum} - 1 \rfloor \tag{4.1}$$

where the function MinF returns the minimum occurrence frequency for a given text document. PhraseNum is the number of extracted nouns and UniquePhraseNum is the number of unique nouns extracted.

The bottom zone is the output zone with a semantic table and a graph. The table contains detailed information about the semantic tags such as definition, parent concept, frequency, score, etc. The table can be sorted according to each defined field. The graph visualizes the semantic tags as nodes and relationships between tags as linkages in a graph. Semantic tags derived from WordNet, OpenCyc, and Wikipedia are labeled in different colors in the graph: red, blue and yellow, respectively. Parents of the semantic tags which don't occur in the text are labeled white. Node sizes represent the frequency of the concept occurrences. Graph type specifies the source of the graph, which can be the hypernym or domain properties of concepts in the ontology. It can serve as a semantic tag cloud for the end user, after simplification, or it can be saved as the semantic tag print of the text input for semantic search or data integration purposes. Figure 4.2 shows an example 3D JUNG graph.

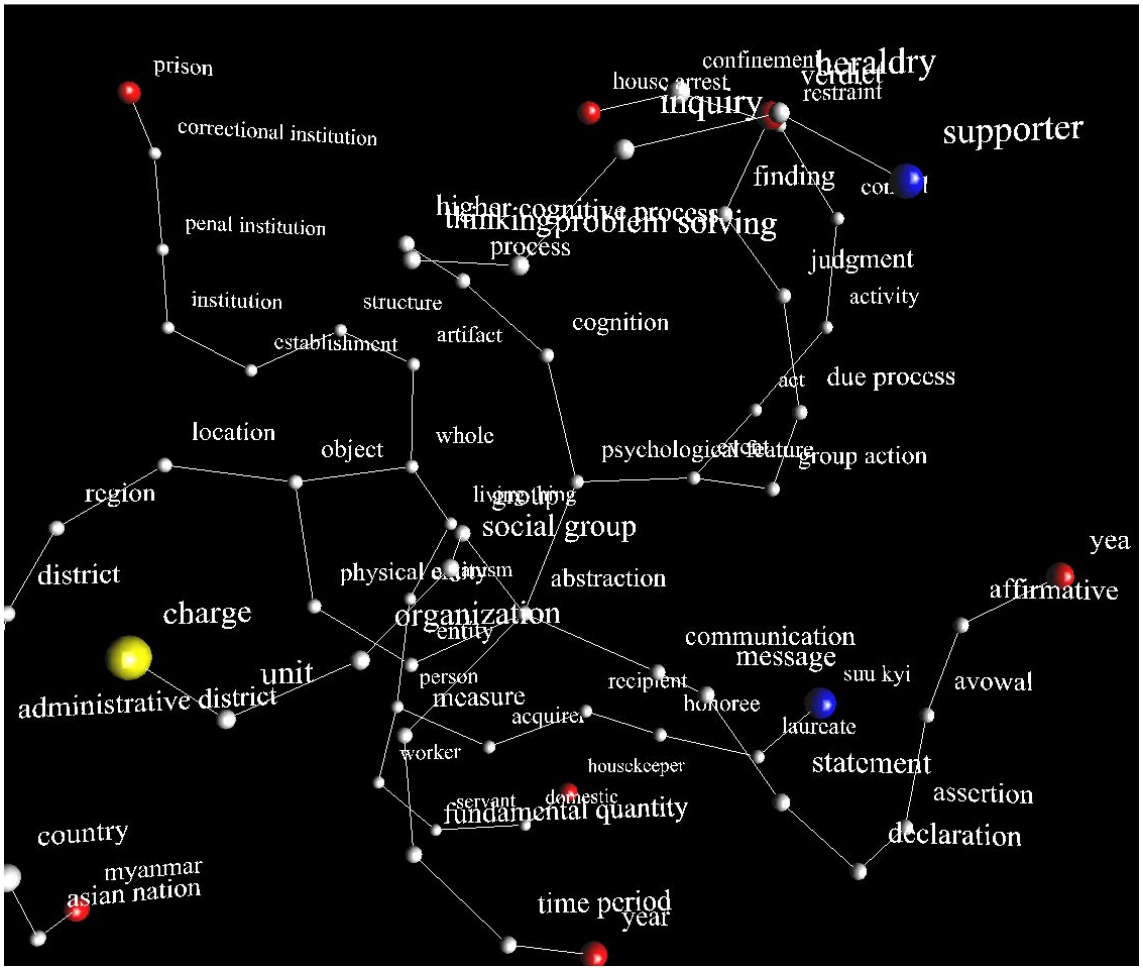

Figure 4.2. JUNG 3D graph example generated in Semantic TagPrint

4.4. SKMT Implementation

SKMT is developed by the integration of Aduna Autofocus [84] and Semantic TagPrint systems. New features are also added to the system such as auto complete, tag cloud, semantic tree, internet search, and semantic search.

Figure 4.3 shows the system architecture of the SKMT framework. In the following sections, information about functionalities of SKMT's modules will be provided.

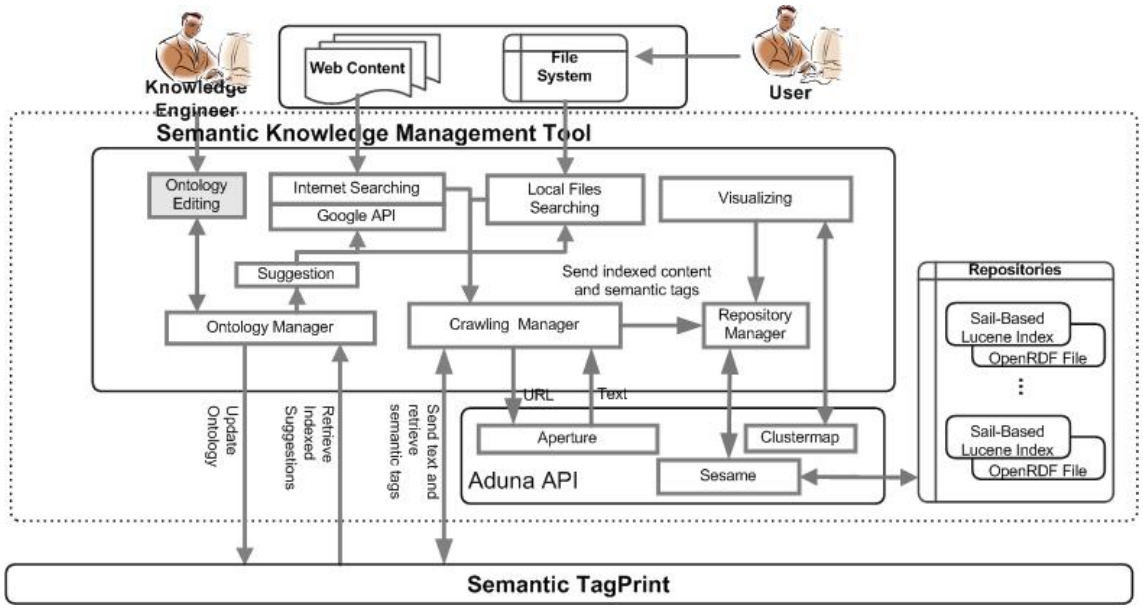

Figure 4.3. SKMT System Architecture

SKMT User Interface composes of the following four main parts: Search, Facets, Cluster Map and Details Panels. These panels are located in Figure 4.4; upper left, lower left, upper right, and lower right parts, respectively.

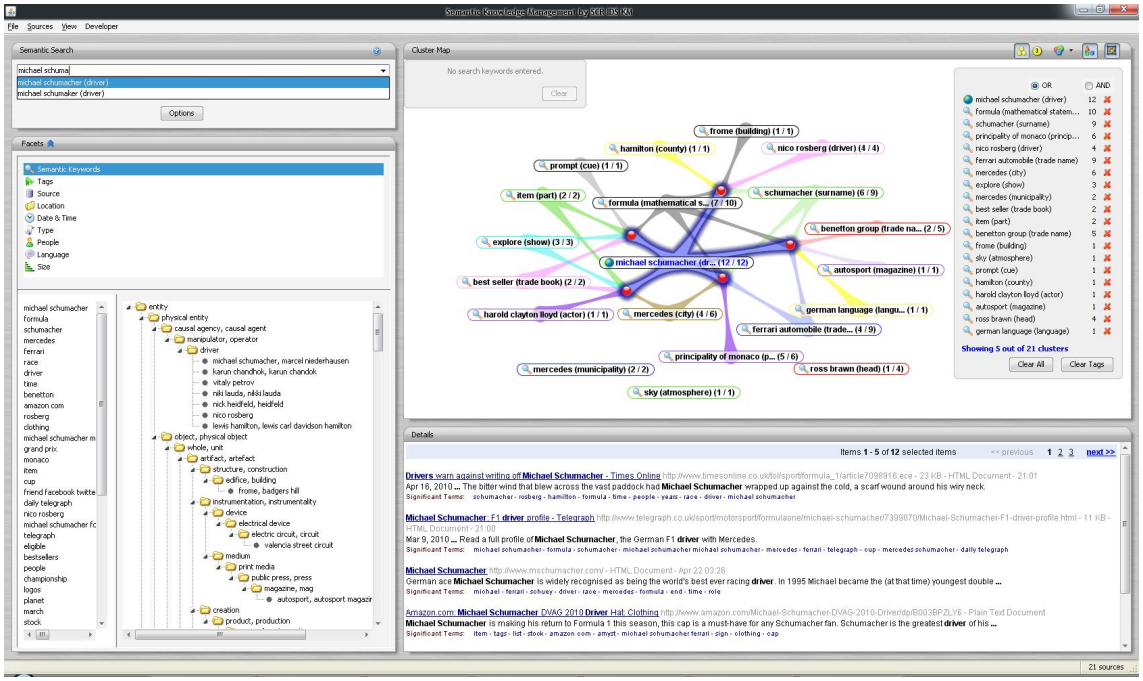

Figure 4.4. SKMT User Interface

4.4.1. Search Panel

Search Panel lets users to build and execute search queries. SKMT supports keyword search, semantic search, and their combinations. When a user starts typing into the auto complete field, the suggestion module of SKMT suggests phrases that start with a given input from the vocabulary of a selected ontology. Suggestions are ontological entities, which are composed of entities' names with their parents' names. Users can select one of the suggestions or enter their own keywords. If a user uses his or her own keywords, then a keyword based search is executed. Otherwise semantic search is done. Users are also able to generate queries that include both semantic phrases and keywords. For instance, a user can search documents about *Siemens company* and *health* constructing the query shown in Figure 4.5.

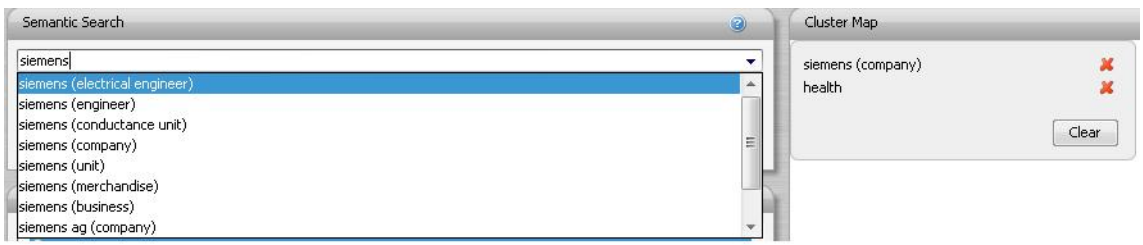

Figure 4.5. Combination of a keyword based and semantic search in SKMT

SKMT supports searching on both local and web resources. Google Search engine is used to enable web searches. Firstly, a constructed query by an SKMT user is converted into a search phrase. Semantic keywords are converted into a phrase with their parent concept names. For example, corresponding search phrase for the query in Figure 4.5 is *Siemens company + health*. Then the search phrase is queried in Google using their API. Addresses of returned N web documents are sent to the Crawling Manager module of SKMT and they are indexed and semantically tagged.

Documents are indexed with unique ids of semantic tags, which are concepts and instances in an ontology to enable semantic search. When a user selects a search phrase from the auto complete field, its unique id is searched over the indexed documents. In this way, the synonym words problem of the keyword based search is solved. Furthermore, documents are also indexed with semantic tags' ancestor concepts to enable searching based on the semantic property hypernym. For instance, when a user searches the concept *president*, documents which are indexed with child concepts and instances of *president* are also retrieved in the results.

Indexed documents are stored as an OpenRDF [85] repository and Lucene index files. The repositories can be exported in various formats such as N3 [86]. Figure 4.6 shows a part of a repository file exported in N3 format.

```
<file:/E:/Projects/SKM/cnn/0.txt> <http://aperture.sourceforge.net/2007/07/19/mad#dateAsNu
    a nfo:FileDataObject , nfo:FileDataObject , nie:DataObject , nfo:PlainTextDocument ;
    nfo:fileLastModified "2009-08-04T17:50:47"^^<http://www.w3.org/2001/XMLSchema#dateTime
    nfo:fileName "0.txt" ;
    nfo:belongsToContainer <file:/E:/Projects/SKM/cnn/> ;
    ads:date "1249397447000"^^<http://www.w3.org/2001/XMLSchema#long> ;
    nfo:fileSize "1770"^^<http://www.w3.org/2001/XMLSchema#long> ;
    nie:mimeType "text/plain" ;
    ads:size "1770"^^<http://www.w3.org/2001/XMLSchema#long> ;
    ads:initialText "F1 team chief Brawn caught speeding" ;
    nie:language "en" ;
    ads:significantTerm "magistrate" , "ferrari" , "benetton" , "ross brawn" , "jenson" ,
    <http://scr.siemens.com/semtag> <http://scr.siemens.com/100001740 100001930 100002684
```

Figure 4.6. Part of a sample repository file in N3 format generated in SKMT

4.4.2. Facets Panel

Facets Panel is used to search indexed documents under various metadata types (faceted search) and shows assigned tags for the clustered documents. Documents can be clustered based on their extraction source, creation time, type, language, and size. Figure 4.7 shows the Facets Panel for the extraction source metadata type.

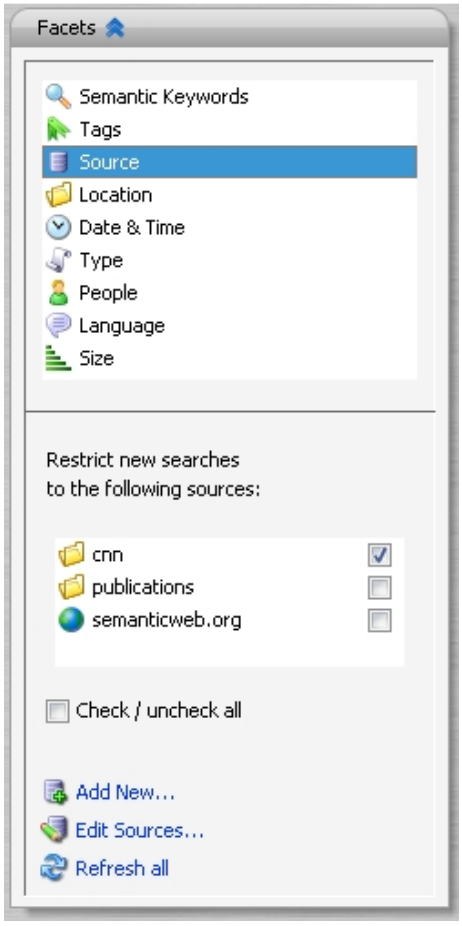

Figure 4.7. SKMT Facets Panel

Semantic Keywords part of Facets Panel shows significant terms and semantic tags for selected documents. Figure 4.8 shows significant terms and semantic tags for the indexed documents, which are tagged with the instance *Michael Schumacher* (Formula One [87] (F1) racing driver) [88].

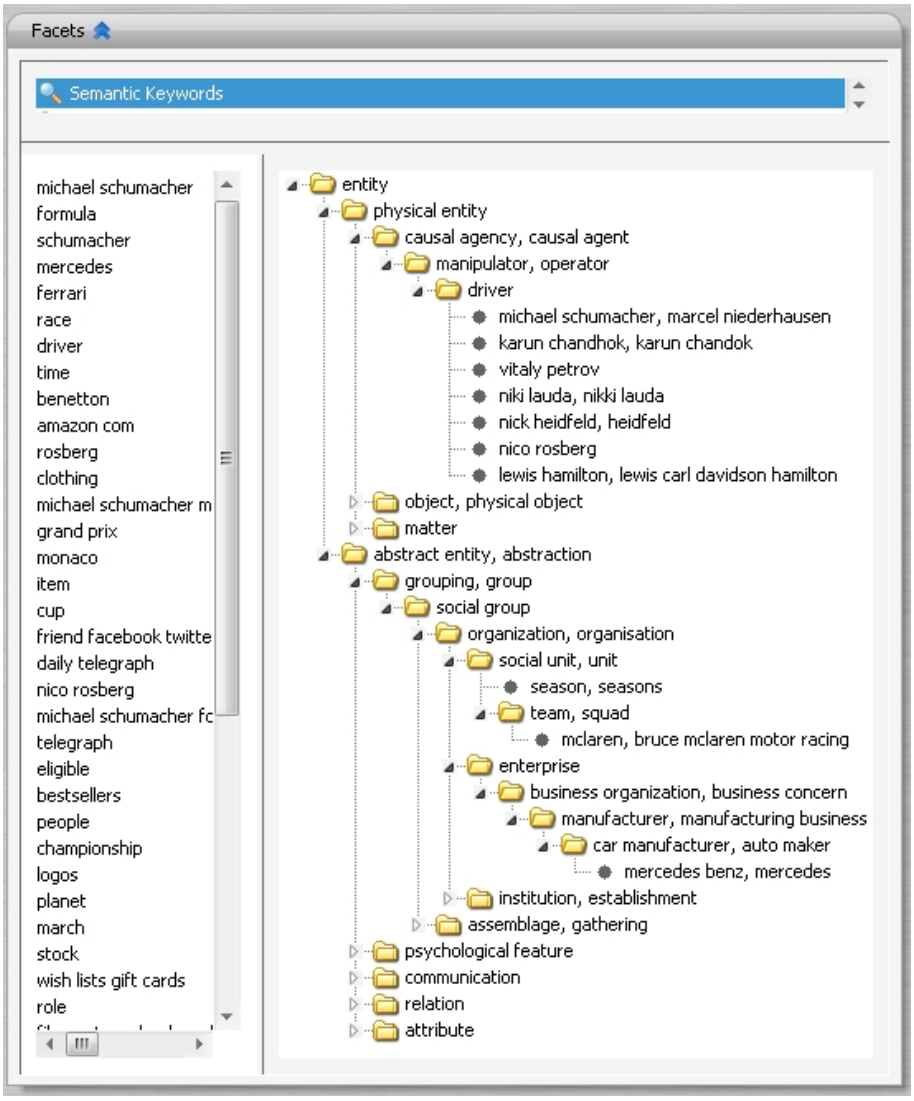

Figure 4.8. List of the semantic tags for the indexed documents, which are tagged with the instance *Michael Schumacher* in SKMT

The left side of Semantic Keywords shows the significant terms in a list. Frequently used top N noun phrases are indexed and shown as significant terms. This feature of Aduna AutoFocus is also modified. Previously, the list was generated with the frequently passed words instead of the noun phrases.

The right side of Semantic Keywords shows semantic tags for selected documents. The tree representation of semantic tags is constructed based on the taxonomy of WordNet ontology. It is one of the new features added to Aduna Autofocus.

4.4.3. Cluster Map Panel

Cluster Map Panel shows search results and indexed documents visually in a graph. Documents are represented as nodes and nodes are connected with search keywords. Documents which contain the same search keywords are clustered. When a user selects one of the clusters, search keywords or documents, it affects the content of both Semantic Keywords Panel and Details Panel. These panels' contents are populated based on the selected items. Figure 4.9 shows the Cluster Map Panel of the SKMT framework.

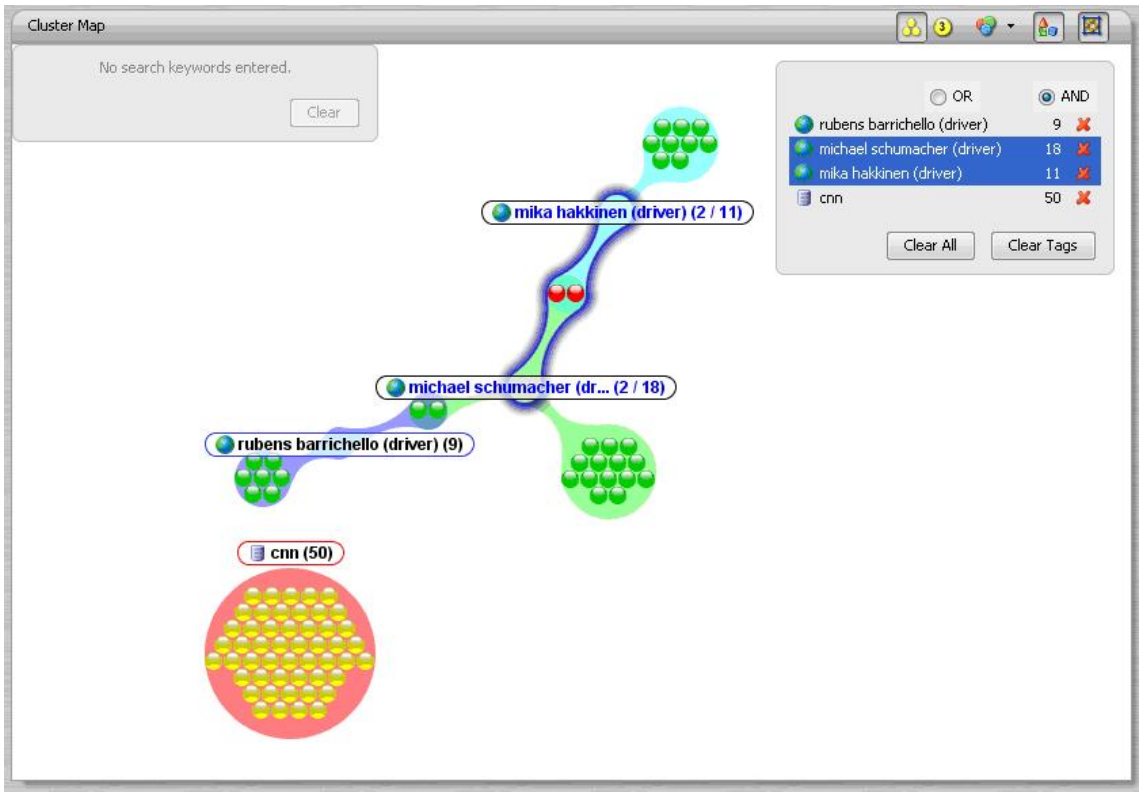

Figure 4.9. SKMT Cluster Map Panel showing indexed documents in a graph

Documents in Cluster Map are represented as nodes in different colors; selected ones are red and other ones are in yellow. The search queries and sources are represented with their names as a cluster consisting of collections of documents. These types of graph elements also include two numbers next to their names; the number of documents in the cluster which includes the selected search queries and the total number of documents in the cluster, respectively.

The right side of the Cluster Map Panel is the list of search queries and document

sources. For example in Figure 4.9, first three items in the list are examples of web search sources and the last item is a local source. Users can select multiple items in the list and see documents fall into the selection in the Details Panel.

The graph in Figure 4.9 consists of three web search queries, one local source and six document clusters. The cluster in the center of the graph consists of a collection of documents, which contains both the semantic tags *Michael Schumacher* and *Mika Hakkinen* (F1 racing driver) [89]. This is one of the benefits of showing the search results visually in a graph. In this way, users can see the documents that contain common search terms.

Cluster Map Panel functionality comes with Aduna AutoFocus. Additionally, the tag cloud feature is added. SKMT supports two kinds of tag clouds: keyword and semantic. Keyword based tag cloud shows the N most frequently used significant terms of the selected clusters as a tag cloud. In contrast, Semantic tag cloud shows the N most frequently used semantic tags of the selected clusters as a tag cloud. Figure 4.10 is an example of a keyword based tag cloud and Figure 4.11 is an example of a semantic tag cloud.

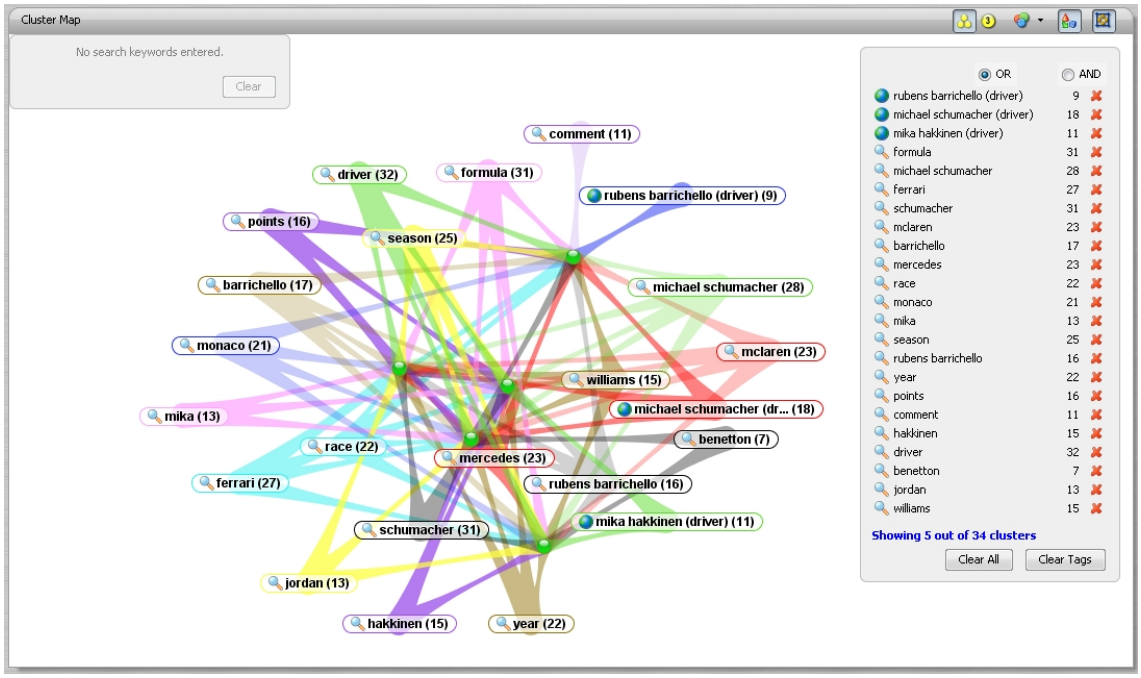

Figure 4.10. Keyword based Tag Cloud for the term *Michael Schumacher* in SKMT

In the tag clouds, related concepts are positioned closer to each other. This feature

gives information about the concepts and the relations between them. In Figure 4.11, *driver*, *formula* and *principality of Monaco* are common concepts between the searched terms *Michael Schumacher*, *Mika Hakkinen*, and *Rubens Barrichello* (F1 racing driver) [90]. *Ferrari* (F1 team) [91] and *Ross Brawn* (F1 team principal) [92], which are in closer positions in the graph to *Michael Schumacher*, are mostly related concepts in the graph as in real life. Similarly, *Mika Hakkinen* is positioned closer to *Republic of Finland*, where he was born [89].

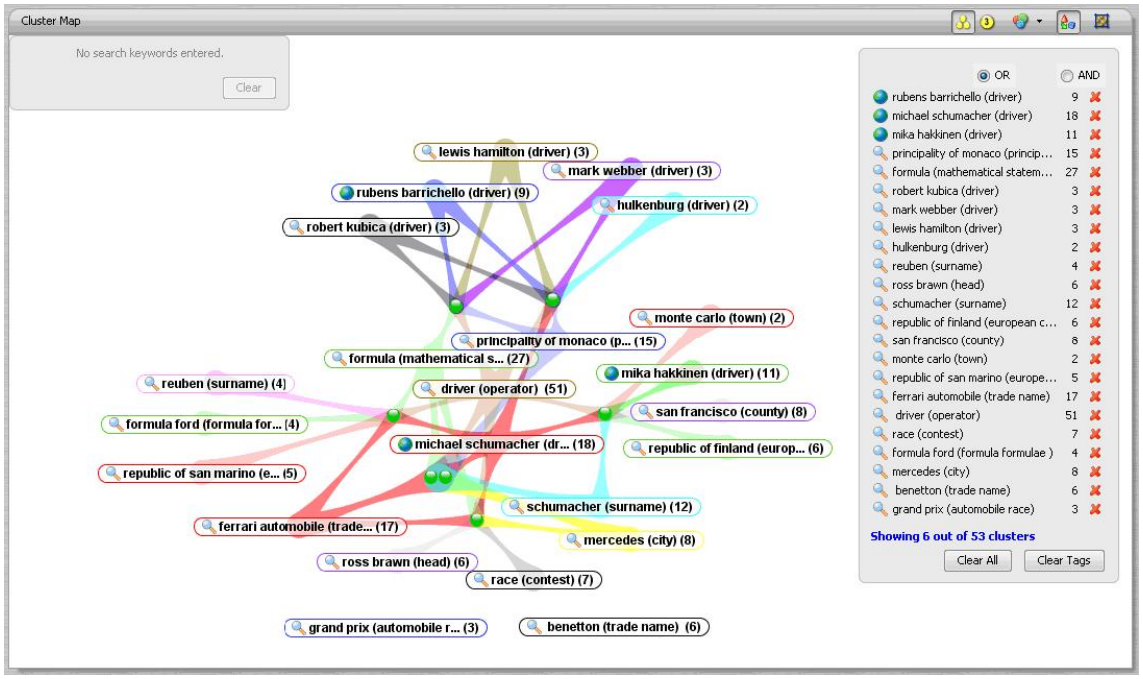

Figure 4.11. Semantic Tag Cloud for the concept *Michael Schumacher* in SKMT

4.4.4. Details Panel

Details Panel shows details of the documents, which are selected in the Cluster Map Panel. Appearance of the panel is similar to Google search results. Each document is represented in three rows. The first row composes of document title, address, size, type, creation time, respectively. The second row includes the initial text of the document if it is a local resource, or returned Google search content, if it is a web content. Third row shows significant terms extracted from the Semantic TagPrint. The bottom part of Figure 4.12 shows details of the documents which are items of a selected cluster.

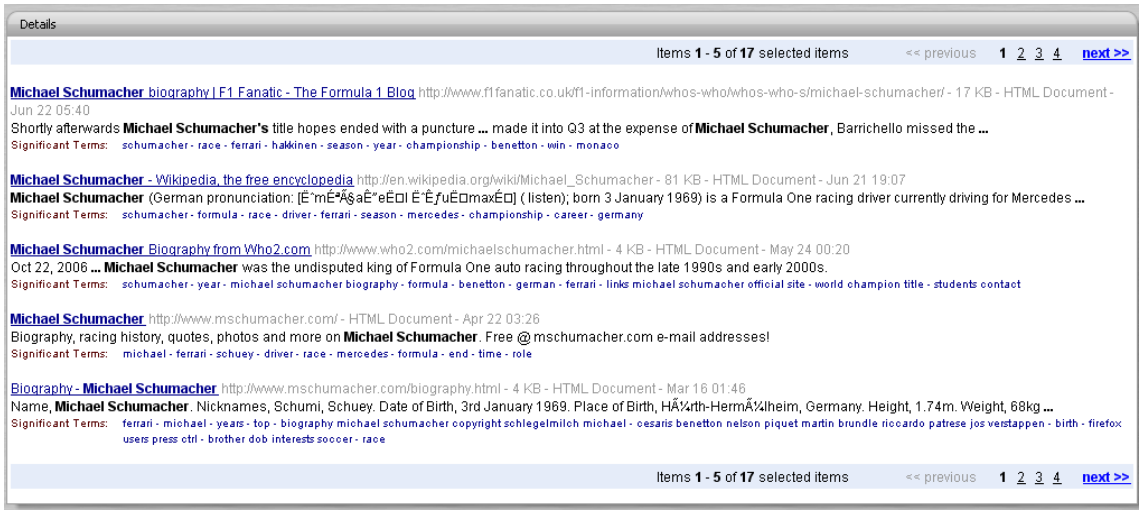

Figure 4.12. SKMT Details Panel showing the search results

5. EVALUATIONS AND EXPERIMENTS

This Chapter includes evaluations of the UNIpedia mappings, automated semantic annotation (WSD) in Semantic TagPrint and the semantic tag recommendation algorithms.

5.1. Evaluation of the UNIpedia Mapping Algorithms

In this section, the data sets used for the experiments, the evaluation metrics, the experiment results, and the performance evaluation of the mapping algorithms between WordNet and Wikipedia are presented.

5.1.1. Data Sets

Test data is required for evaluating performances of the concept selection and sense mapping algorithms of UNIpedia. Performance evaluation can be conducted in two ways: manual evaluation and automated evaluation. Automated evaluation relies on a computer processable test data set. For this, the manually created links between DBPedia and WordNet would be the test dataset. However, as discussed before (see section 2.3.2), these links are incomplete and out-of-date.

Since no appropriate computer processable test data was available, human volunteers were used. The manual evaluation of the WordNet and Wikipedia mappings is a challenging and error prone task that may require human expertise. The volunteers must be familiar with the Wikipedia instances they annotate. They must also be familiar with WordNet concepts in order to appropriately define the *isA* relationships. To mitigate this issue, the number of annotators was kept small and they were requested to annotate carefully considering all the meanings of WordNet concepts. Randomly selected 100 Wikipedia pages were annotated by five volunteer annotators. They were provided with extracted WordNet concepts for the Wikipedia pages and requested to select the most appropriate WordNet concept for a page. When none of the suggested WordNet concepts appropriately defined the Wikipedia pages, they were told to introduce a concept of their

own or annotate as *none*.

Based on the experiment results, coefficients were defined for the voting algorithm and metadata sources to be utilized in the selection step.

5.1.2. Evaluation Metrics

In the *Alignment* step, extracted metadata is aligned to WordNet concepts. This step includes processing metadata to find corresponding WordNet terms and disambiguation of the corresponding terms. Only the performance of the disambiguation (sense mapping) algorithm was evaluated. The accuracy of the sense mapping algorithm is calculated as follows:

$$Acc_{SenseMapping} = \frac{\text{True sense mappings \#}}{\text{Sense Mappings \#}} \tag{5.1}$$

Calculating the performance of the concept selection is challenging. For one thing, a Wikipedia article could be an instance of multiple concepts in WordNet. Consequently, annotators may assign the article to several concepts, where each assignment is correct. For instance, *Michael Jackson* is both an *artist* and *singer*. It is perfectly acceptable that annotators may identify *Michael Jackson* as instances of both *artist* and *singer*.

Since Wikipedia pages are annotated by multiple annotators, the match between the annotator and heuristic selections are weighted using the following formula:

$$weight(p) = \frac{manualAnnotation(unipediaRecommendation(p))}{maxManualAnnotation(p)} \tag{5.2}$$

Where *maxManualAnnotation(p)* is the selection number of the most selected Word-

Net concept for a given Wikipedia page p by the annotators. This formula increases a heuristic's score by one, if the heuristic recommends the most selected WordNet concept for a given Wikipedia page p by the annotators.

Total performances of the semantic annotation algorithms are calculated using the widely used measures Precision (P), Recall (R) and F-measure (F).

Precision gives information about the correctness of given answers by the system. Precision is defined as follows:

$$Precision = \frac{\#\text{ correct answers provided}}{\#\text{ answers provided}} \tag{5.3}$$

Recall is another important measure that gives information about the coverage and the performance of a system. Recall is defined as follows:

$$Recall = \frac{\#\text{ correct answers provided}}{\#\text{ all mappings}} \tag{5.4}$$

F-measure is helpful to evaluate systems which don't answer all given input values that have coverage lower than 100%. A system can achieve 100% precision without answering any queries. To asses overall performance of a system, F-measure is widely used. F-measure values can be calculated giving varying weights to precision and recall. The traditional F-measure (F1-measure) which gives equal weighs to precision and recall (harmonic mean) is used in this study. F1-measure is defined as follows:

$$F1 - measure = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

5.1.3. Results and Discussion

In Figure 5.1, *Description* and *Infobox* values show the reliability of the metadata sources on randomly selected pages. The precision of the extracted metadata from the descriptions of Wikipedia pages is significantly better than the extracted metadata from the infobox template names of the pages.

In the first round of performance evaluation, metadata extracted from the infobox templates were ignored and the weights of concepts extracted from the description of Wikipedia pages were increased.

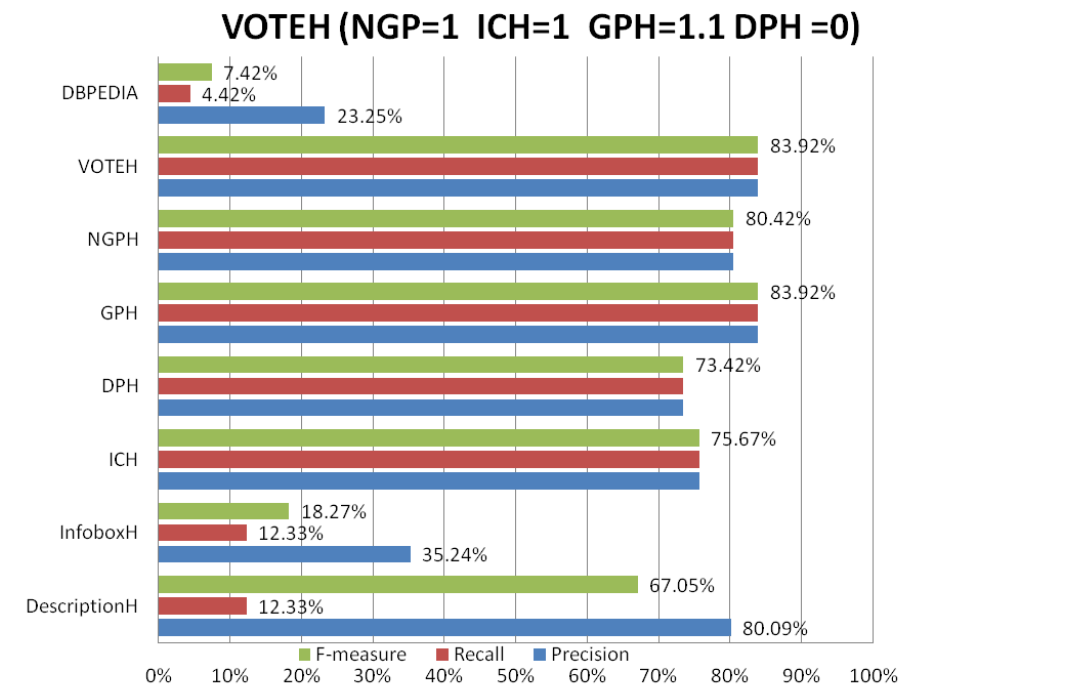

Figure 5.1. Comparison performance values of the concept selection algorithms over the randomly selected 100 pages

ICH, *DPH*, *GPH* and *NGPH* values show the performance of the ontological and statistical properties as selection heuristics.

After evaluating the experiment results, a voting algorithm was decided to be used in order to increase the performance of the system with varying weights between the selection heuristics. Since *GPH* performed better than other heuristics, it got the highest

weight. *DPH* performed worst; therefore it received zero weight in the voting algorithm.

The accuracy of the sense mapping algorithm was evaluated manually, based on the concepts selected by *VOTEH*. As stated before, existing manually created links derived from DBpedia and OpenCyc were used to create a mapping pool. The sense mapping algorithm maps a phrase to one of its senses if one of the senses is defined in the pool. Otherwise, the phase is mapped to the highest ranked sense of the phrase. When there is more than one corresponding sense for the phrase, the sense with the higher frequency in the pool is selected. The performance of the sense mapping algorithm scored better (90.00% accuracy) in comparison to directly mapping to highest sense (80.00% accuracy).

After evaluating the algorithms, the VOTEH algorithm with the links pool composes of DBpedia and OpenCyc links was selected for construction of UNIpedia. The size of UNIpedia is shown in Table 5.1. Wikipedia has the highest contribution to UNIpedia. Each Wikipedia page corresponds to an instance in UNIpedia and 1,491,902 of 2,722,401 Wikipedia pages are successfully mapped to WordNet. The other pages could not be mapped due to inadequate metadata or because they were filtered out by the system on the account that they are category pages. OpenCyc also has considerable contribution to UNIpedia. UNIpedia is composed of 2,242,446 unique words (phrases) that may well cover most of the daily used terms.

Table 5.1. Size of UNIpedia

	Word Sense Pairs	**Words**	**Concepts**	**Instances**
WordNet	146,312	117,798	74,390	7,725
Wikipedia	2,305,689	2,144,837	0	1,442,694
OpenCyc	68,621	65,097	0	41,483
UNIpedia	2,520,622	2,242,446	74,390	1,491,902

The current DBpedia WordNet mappings were used as a baseline to compare UNIpedia results (Figure 5.1). UNIpedia performed significantly better (84% precision) and recall values compared to manually created links of DBpedia. YAGO is another system that links WordNet and Wikipedia. YAGO is also evaluated by human annotators and it has

94.54% type (isA) relationship accuracy. In contrast, UNIpedia achieved 83.92% linking accuracy. YAGO has a higher accuracy, since it does not perform any selection over metadata sources. YAGO defines all of the aligned Wikipedia categories with links to WordNet. Moreover, YAGO does not perform any Word Sense Disambiguation, and directly maps a term to a concept that has the highest rank among all meanings of the term. There were experiments where UNIpedia had higher accuracy in comparison to the highest sense algorithm.

5.2. Evaluation of the Semantic TagPrint Semantic Annotation Algorithms

This section presents comparative evaluations of the Semantic TagPrint sense mapping algorithms. First let us introduce the test data sets and evaluation metrics.

5.2.1. Data Sets

Semantic TagPrint currently supports two ontologies: WordNet and UNIpedia. Since there is no appropriate computer processable test data that is annotated with UNIpedia entities, the sense mapping algorithms are evaluated on the SemCor data set (see section 2.2.1).

5.2.2. Evaluation Metrics

Precision, Recall and F-measure are used to evaluate the Semantic TagPrint semantic annotation algorithms as in the case of section 5.1.2.

5.2.3. Results and Discussion

Hypernym algorithm depends on the maximum ancestor distance constant. Firstly, Hypernym algorithm is evaluated with varying maximum ancestor distances on the SemCor data set. Figure 5.2 shows the performance values. The Highest performance is observed with maximum ancestor distance value 3. The performance is decreasing after that point, because most of the concepts gets connected with common ancestors even they are not much related.

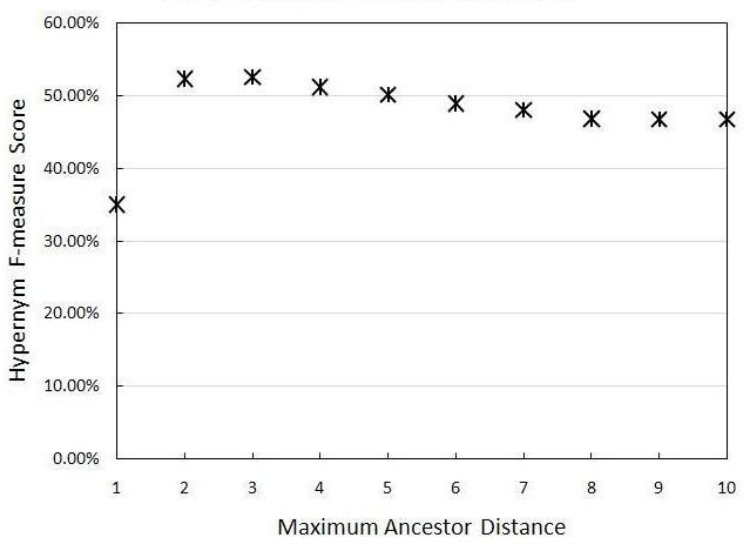

Figure 5.2. Evaluation of Hyprenym algorithm with varying max ancestor distances

Table 5.2 shows the precision, recall, and F-measure of Hypernym, Domain, Holonym, Category, Sense Rank, SUM and SUM+ algorithms based on the SemCor data set.

Table 5.2. Comparison of Precision, Recall and F-measure values for the WSD

Algorithm	Category	Domain	Hypernym	Holonym	SUM	RANK	SUM+
Precision	0.437	0.402	0.542	0.769	0.559	0.797	0.795
Recall	0.280	0.257	0.511	0.308	0.536	0.797	0.795
Fmeasure	0.342	0.313	0.526	0.440	0.547	0.797	0.795

Category and Domain algorithms performed worse than Hypernym and Holonym algorithms, because categories are high level senses which can't give enough information to disambiguate terms. Corresponding senses of a word can get the same category value that decreases the recall of the algorithm. Domain algorithm suffers from the low coverage (recall), only %8 of WordNet concepts has defined domain relation. Domain and category relationships of WordNet can be concluded to be limited for disambiguating words in a context.

Hypernym based lexical chaining algorithm performed better than other chaining algorithms in terms of recall and f-measure, since hypernym relations of senses are well

defined in WordNet.

The performances of the above algorithms give clues about their coefficients in SUM and SUM+ algorithms. Holonym algorithm has the highest precision value compared to the other algorithms. Thus, it has the highest coefficient in the SUM and SUM+ algorithms. Table 5.3 shows the variable coefficients that maximizes performance of the SUM+ algorithms. Different coefficient combinations (0,1,2) are experimented to maximize the performance of the SUM algorithm. Then, SUM+ algorithm is evaluated using lexical chain coefficients that maximize the SUM algorithm with varying sense rank coefficient.

Table 5.3. Coefficients of the SUM and SUM+ lexical chaining algorithms

Sense Rank	**Hypernym**	**Holonym**	**Category**	**Domain**
1	1	2	0	0

SUM+ and Sense Rank utilize the sense ranks and perform better than the other algorithms. It is often hard to beat the Sense Rank algorithm especially using the knowledge based approaches, because the ranks of the senses are calculated based on the occurrence frequency of senses in the SemCor data set.

Based on observations, SUM+ is preferred in the sense mapping task since it performs better than the other lexical chaining algorithms. SUM+ gives priority to WordNet concepts using the sense rank values. When there are defined instances for a given term only in UNIpedia, SUM+ algorithm behaves like the SUM algorithm because instances are ranked equally in UNIpedia.

5.3. Evaluation of the Semantic TagPrint Concept Weighting Algorithms

This section presents comparative evaluations of Semantic TagPrint concept weighting and tag recommendation algorithms. First let's introduce the experiment data sets and evaluation metrics.

5.3.1. Data Sets

To evaluate Semantic TagPrint tag recommendation algorithms, a CNN news data set is created by collecting 50 CNN news articles covering a variety of domains such as politics, sports, technology, business, etc. News articles from the CNN Web site were preferred since these articles are tagged by their authors.

Since Tagging is a subjective task, author assigned tags are not always good enough to represent content of a document [93]. Therefore, human volunteers were used to tag the CNN news data set. Each article is tagged by five volunteers. They are asked to provide as many tags as they feel necessary to represent the content of the article. As shown in Table 5.4, The result CNN data set consists of totally 192 author-assigned and 973 user- and reader-assigned unique tags per article.

Table 5.4. CNN test data characteristics

	Author Tags	**Author & Reader Tags**
Assigned tags	192	973
Tags passes in content	138	805
Tags defined in Unipedia	149	680
Tags defined in Wordnet	62	464

5.3.2. Evaluation Metrics

Semantic TagPrint is evaluated by counting the exact matches with these manually assigned tags. Since the documents are tagged by multiple annotators, significance is given to commonly used tags. Therefore, exact matches are weighted using the following formula [93]:

$$w(t) = 1 + \ln\left(freq(t)\right) \tag{5.6}$$

Where *freq(t)* is the number of t assigned as a tag by the annotators to a given

document. Semantic TagPrint system offers ontological entities as tags. A semantic tag is counted as an exact match if one of its synonyms or ancestor's synonyms is an exact match to a manually assigned tag.

5.3.3. Results and Discussion

Statistical and ontological features of UNIpedia are utilized for concept weighting. First, a weight formula to summarize the features of the coefficients is derived from a training data set using linear regression. Table 5.5 shows the variable coefficients in the Semantic TagPrint weighting formula.

Table 5.5. Coefficients of the tag weighting algorithm

TF	**Hypernym**	**Holonym**	**IGP**	**Domain**	**IC**	**Depth**
1.9	0.0	0.9	1.3	0.4	1.2	1.6

IGP, Depth, and IC favor specific terms and get the higher coefficients in the formula since users usually select named instances or specific terms as tags. Holonym also get a higher weight because it provides valuable information about the related terms within a content. In contrast, Domain property suffered from its low coverage. Hypernym property didn't perform well because general terms also have strong parent child relations.

There are many automated tagging and keyword extraction systems in the literature, but most of them are not freely available. Kea [94] is a publicly available keyphrase extraction system that is widely used for the experimenting purposes. However, Kea uses a machine learning algorithm that requires a domain specific training data [95]. Therefore, Semantic TagPrint is compared to context-based (tags documents based on their context) and publicly available tagging systems Zemanta [13] and Yahoo Term Extraction [14].

Zemanta is an automatic semantic tagging system that suggests content from various sources such as Wikipedia, YouTube, Flickr, and Facebook. Zemanta disambiguates terms and maps them to the Common Tag ontology [96]. Common Tag is a tagging scheme in RDF, a format that consists of properties such as URI value and login date. URI value is

a pointer to a document identifying the meaning of tags.

Yahoo Term Extraction is a free Web service provided by Yahoo, which analyzes the given content and offers a list of significant words. Additionally, Semantic TagPrint is compared to Frequently Passing Noun Phrases algorithm that suggests frequently used noun phrases within a document. Table 5.6 and Table 5.7 show the performance of the tagging systems on the CNN data set with suggestion number 8 tags per document.

Table 5.6. Performances of the tagging systems for author assigned tags

Algorithm	**Assigned Tag**	**Unweighted Match**	**Weighted Match**	**P (%)**	**R (%)**	**F (%)**
Semantic TagPrint	3.70	1.40	1.40	17.5	42.3	24.0
Zementa	3.70	1.87	1.87	23.3	53.1	31.4
Yahoo Term Extraction	3.70	0.43	0.43	5.4	12.8	7.2
Frequently Occuring Noun Phrases	3.70	0.77	0.77	9.6	23.3	13.2

Table 5.7. Performances of the tagging systems for author and volunteers assigned tags

Algorithm	**Assigned Tag**	**Unweighted Match**	**Weighted Match**	**P (%)**	**R (%)**	**F (%)**
Semantic TagPrint	19.57	4.20	7.37	52.5	20.6	29.2
Zementa	19.57	3.37	6.12	42.1	17.9	24.8
Yahoo Term Extraction	19.57	1.80	3.10	22.5	9.4	13.2
Frequently Passing Noun Phrases	19.57	3.23	5.68	40.4	16.9	23.5

Semantic TagPrint and Zemanta performed significantly better than the other tagging systems. Zemanta performed better for the author assigned tags. In contrast, Semantic TagPrint performed better for the reader and author assigned tags. This is because

Wikipedia and Author assigned tags mostly contain instances of concepts *people* and *location* such as Barack Obama and *America*. Also, Zemanta suggests terms from Wikipedia. In contrast, our system offers both concepts and instances that define the context of a document rather than the specific named instances.

Furthermore, Semantic TagPrint can offer semantic tags of a keyword's synonyms or its parent concept's synonyms. For example, Semantic TagPrint tags a CNN article with *Barack Obama* and only *President Obama* is mentioned in the article. Semantic TagPrint can offer such a tag because *President Obama* and *Barack Obama* are synonyms of a specific instance in UNIpedia. Another example is an article that is tagged with the tag *golf player*. Semantic TagPrint tags the article with *Matt Bettencourt*, which is a child of the concept *golf player* in UNIpedia. Similarly, the system can derive *golf player* information via *Matt Bettencourt*.

6. CONCLUSIONS AND FUTURE WORK

This thesis presents an automated semantic tagging system to capture machine understandable meaning of a given text content with its semantic tags. Word Sense Disambiguation and knowledge base coverage are key challenges in such a system.

To address the knowledge base coverage problem, an ontology framework model and implementation which is applicable to semantical document processing has been developed. Statistical and ontological properties of concepts and instances were utilized in rule based heuristic algorithms. Wikipedia and OpenCyc were chosen for the implementation of the framework since they are known to contain up to date instances. Useful metadata resources within Wikipedia and OpenCyc were identified. UNIpedia is a high quality and extendable ontological knowledge base. The proposed approach supports working with other knowledge bases, except for the metadata extraction step.

Results of the experiments show that our mapping algorithm between WordNet and Wikipedia is fairly accurate. It is observed that existing links between WordNet and DBpedia are sparse and out-dated, and experiments demonstrated that the links generated by UNIpedia were significantly better. The manually created links were used in the disambiguation task which increased the system performance.

The WordNet ontology coverage increased approximately 20 times after integration with Wikipedia and OpenCyc. Such an extensive and high quality ontology would be a useful resource in knowledge based AI applications, especially in NLP applications that process free form text.

To address the Word Sense Disambiguation problem, a linear time lexical chaining Word Sense Disambiguation algorithm is proposed for automated semantic annotation. Moreover, a series of experiments is conducted to evaluate the performance of the system. The mapping algorithm is compared on the SemCor sense annotated dataset and the experiments show that our mapping algorithm is fairly accurate.

Tag recommendation performance of Semantic TagPrint is compared with those generated from Yahoo Term Extraction, Zemanta, and Frequently Passing Noun Phrases algorithm. Evaluations show that the tag recommendation used in this study performs better than other systems and algorithms.

The potential benefits of Semantic TagPrint are demonstrated by the design and implementation of the Semantic Knowledge Management Tool (SKMT). SKMT provides a user accessible platform for Semantic TagPrint to semantically tag documents, and performs semantic searches. Advantages of semantic tags are demonstrated by comparing semantic search with keyword based search on sample search scenarios.

The experiment results and our evaluations show that the designed and implemented semantic tagging system would be a useful system to manage documents and locate desired information.

As future work, the proposed semantic tagging system could be improved in several directions.

- UNIpedia can be improved in several directions. Unification with other reliable ontological knowledge bases to improve the size and coverage of UNIpedia is planned. WordNet, as the backbone ontology, has limited domain and holonymy relationships. Domain and holonymy relationships can be extracted from other knowledge bases to build a richer ontology.
- Semantic TagPrint tagging performance can be improved by using statistical metrics such as word co-occurrence.
- SKMT can be customized with a domain-specific ontology and search applications.
- A Social Network can be formed in SKMT. Users can be represented based on their shared resources. Social network analysis could be performed using the tool and users can be related with each other.
- The proposed model currently supports only English language. Other languages may be supported by constructing a multi-language ontology.

APPENDIX A: Test Data Sets

A.1. The UNIpedia Test Data

This section provides the details of the test data used in evaluating UNIpedia. Table A.1 shows the randomly selected Wikipedia article titles, the UNIpedia mappings, and the manual mappings respectively. UNIpedia always maps a single concept. Frequencies of the manual annotations are provided in parentheses following the annotations.

Table A.1: The UNIpedia mappings and manual annotations of 100 randomly selected Wikipedia pages by 5 annotators.

Wikipedia Page Titles	**UNIpedia Mapping**	**Manual annotations**
Anaplastology	medicine	medicine(5)
Argenton-sur-Creuse	commune	commune(5)
Ariosophy	names	names(4), ideology(1)
Averna	queen	goddess(2), mythical goddess(1), queen(1), roman mythology(1)
Ayyalur	town	town(5)
Barruecopardo	village	village(5)
Bsmoen	village	village(5)
Bentheim-Lingen	county	county(4), country(1)
Bokajan	town	town(5)
Boragh	armoured personnel carrier	armoured personnel carrier(4), weapon(1)
Brcourt	bunker	bunker(5)
Bufoides	endangered species	endangered species(5)

Table A.1: The UNIpedia mappings and manual annotations of 100 randomly selected Wikipedia pages by 5 annotators continue.

Wikipedia Page Titles	**UNIpedia Mapping**	**Manual annotations**
Bunyip	legendary creature	legendary creature(2), mythology(1), folklore(1), mythical creature(1)
Burland	village	village(5)
Cadaverine	molecule	molecule(5)
Capsazepine	analogue	analogue(3), pharmacology(1), none(1)
Cayke	character	character(4), none(1)
Chandurthi	village	village(5)
Chanunpa	pipe	pipe(5)
Cheesecake	cake	cake(3), dessert(2)
Chhatari	town	town(5)
CIP-Tool	software	software(4), software program(1)
Colotenango	city	city(5)
Concentration	measure	measure(3), chemical proportion(1), chemical property(1)
Cornette	piece	headgear(4), headwear(1)
Crayke	village	village(5)
Cryptoclidus	genus	genus(4), plesiosaur(1)
Delgany	village	village(5)
Diffwys	mountain	mountain(5)
Digswell	village	village(5)
Dogbowl	guitarist	guitarist(2), artist(2), singer(1)
Elf-Arrows	people	weapon(2), none(2), folklore(1)
Ethenzamide	analgesic	anti-inflammatory drug(3), analgesic(1), drug(1)
Exloo	city	village(2), settlement(2), city(1)
Friesach	town	town(5)

Table A.1: The UNIpedia mappings and manual annotations of 100 randomly selected Wikipedia pages by 5 annotators continue.

Wikipedia Page Titles	**UNIpedia Mapping**	**Manual annotations**
Gastrinoma	tumor	tumor(5)
Giuiria	genus	genus(5)
Goldin	surname	surname(5)
Gramalote	city	city(3), municipality(1), town(1)
Guitiriz	municipality	municipality(4), city(1)
Hagstrm	company	company(4), instrument(1)
Hakapik	tool	club(4), tool(1)
Heage	village	village(5)
Holycross	village	village(5)
Kakka	god	god(3), deity(2)
Kapia	family	surname(5)
Kjerkeberget	mountain	mountain(4), hill(1)
Limours	commune	commune(4), town(1)
Loliondo	district	village(3), settlement(1), district(1)
Martinov	village	village(4), municipality(1)
Melanocorypha	genus	genus(5)
Moed	order	order(4), religious order(1)
Molazzana	town	town(3), commune(1), city(1)
Musketeer	type	warrior(2), infantry(2), soldier(1)
Nayah	singer	singer(5)
Nisterberg	municipality	municipality(4), location(1)
Noetus	native	clergy(4), christian(1)
Norg	city	village(4), settlement(1)
Okayplayer	collective	website(5)
Orientamenti	book	book(5)
Passade	municipality	municipality(4), manicipality(1)

Table A.1: The UNIpedia mappings and manual annotations of 100 randomly selected Wikipedia pages by 5 annotators continue.

Wikipedia Page Titles	**UNIpedia Mapping**	**Manual annotations**
PCMark	series	benchmark(3), computer software(1), benchmark tool(1)
Pecan	species	species(3), carya(2)
Pealoln	commune	commune(5)
Piprawa	village	village(5)
Pishtane	village	village(5)
Piton	spike	spike(2), equipment(2), none(1)
Placemaking	landscape architecture	terminology(3), landscape architecture(1), term(1)
Pleaseeasaur	act	act(3), musical act(1), musical group(1)
Potzbergturm	tower	tower(5)
Renix	engine	engine(5)
Rdtligen-Alchenflh	municipality	municipality(5)
Sarcocornia	genus	genus(4), amaranthaceae(1)
Satalkheri	town	town(5)
Scagliola	technique	technique(2), archicture(1), building material(1), achitectural technique(1)
Schallstadt	location	town(4), city(1)
Schillings	surname	surname(5)
Schirmitz	municipality	municipality(5)
Scido	geography	town(3), commune(1), city(1)
SIDPERS	database	database(5)
Skancke	family	family(4), surname(1)
Soultone	amplifier	company(4), manufacturer(1)

Table A.1: The UNIpedia mappings and manual annotations of 100 randomly selected Wikipedia pages by 5 annotators continue.

Wikipedia Page Titles	**UNIpedia Mapping**	**Manual annotations**
Stanford-on-Avon	hamlet	hamlet(4), village(1)
Stretched	singles	singles(4), song(1)
Supraorganization	organization	organization(5)
Svileuva	village	village(5)
Thoracoscopy	medical procedure	medical procedure(5)
Tinkerbelle	sailing vessel	sailing vessel(4), sailboat(1)
Uhrhaa	island	island(5)
Vadavannur	village	village(5)
Vazhakulam	town	town(5)
Veges	surname	surname(4), none(1)
Vennaskond	rock group	rock band(4), rock group(1)
Vestervig	settlement	settlement(3), town(2)
Villeneuve-la-Garenne	commune	commune(5)
Villingendorf	district	town(5)
Walong	cantonment	town(5)
WDVH-FM	radio station	radio station(5)
XDrawChem	software program	software program(5)
XET-AM	radio station	radio station(5)

A.2. The Semantic TagPrint Test Data

This section provides the details of the test data used in evaluating Semantic TagPrint. Table A.2 shows the addresses, author assigned tags, and reader assigned tags of the collected 50 CNN news articles respectively.

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data.

1	http://edition.cnn.com/2009/SPORT/08/04/f1.ross.brawn.button.speeding/index.html
	ross brawn, jenson button, rubens barrichello, formula one racing, motorsport
	ross brawn(7), speeding(6), formula one(4), f1(3), court(2), england(1), driver(1), mercede(1), traffic ticket(1)
2	http://edition.cnn.com/2009/SPORT/football/08/04/lionel.messi.barcelona.argentina/index.html
	lionel messi, zlatan ibrahimovic, fc barcelona, real madrid cf, argenti, european football
	barcelona(5), lionel messi(4), contract(3), messi(2), deal(2), football(2), soccer(2), transfer(1), spanish football league(1), barcelona football team(1), spain(1)
3	http://edition.cnn.com/2009/WORLD/asiapcf/08/04/yettaw.hospitalized/index.html
	myanmar, aung san suu kyi
	suu kyi(6), myanmar(5), house arrest(3), nobel peace prize(2), arrest(2), john william yettaw(2), yettaw(2), junta(2), john willian yettaw(1), myanmar government(1), aung san suu kyi(1), immigration law(1), convulsion(1), election(1), prison(1), american(1), ciolation(1), trial(1), hospital(1)
4	http://edition.cnn.com/2009/WORLD/asiapcf/08/03/australia.terror.raids/index.html
	australia, terrorism, somalia, al qaeda

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	al qaeda(5), al shabaab(4), australia(4), somali(3), somalia(2), terrorist act(2), terrorist plot(2), warrant(1), al shabbab(1), terrorist attack(1), melbourne(1), failed(1), the somali militia(1), terror plot(1), police(1), investigation(1), islamic militia(1), terrorist(1)
5	http://edition.cnn.com/2009/WORLD/asiapcf/08/04/nkorea.clinton/index.html
	north korea, nuclear weapon, kim jong il, euna lee, laura ling
	bill clinton(6), north korea(5), kim jong il(4), american journalist(4), prison(3), euna lee(2), kim jong(2), kcna(2), negotiation(2), laura ling(2), release(2), north korean(1), pyonyang(1), nuclear program(1), united state(1), private mission(1), visit(1), pyongyang(1), nuclear(1), chi(1), white house(1), korean central news agency(1), journalist(1), obama(1)
6	http://edition.cnn.com/2009/SHOWBIZ/TV/08/03/kara.dioguardi.idol/index.html
	american idol, adam lambert, paula abdul, ryan seacrest
	american idol(6), kara dioguardi(5), fox(4), paula abdul(3), songwriter(1), announcement(1), dioguardi(1), tom jones.(1), contract(1), singing competition(1), mike darnell(1), theory of a deadman(1), kelly clarkson(1), judging(1), jessie jame(1), pink(1), simon fuller(1), jury(1)
7	http://edition.cnn.com/2009/POLITICS/08/02/geithner.economy/index.html
	u.s. national economy, timothy geithner, federal budget
	timothy geithner(4), economy(4), alan greenspan(3), deficit(2), america(2), geithner(2), mike pence(2), value added tax(2), abc(1), this week(1), discussion(1), unemployment(1), american economy(1), treasury secretary(1), stimulus package(1), u.s. economy(1), dept(1), healthcare(1), united state(1), job(1), greenspan(1), barack obama(1), recovery(1), tax(1), john mccain(1)
8	http://edition.cnn.com/2008/TECH/space/11/16/shuttle.endeavour.docking/index.html?eref=edition_space

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	space shuttle endeavour, nasa, international space station
	nasa(6), endeavour(6), astronaut(3), space shuttle(2), shane kimbrough(1), shuttle(1), space(1), leroy cain(1), mike fincke(1), dock(1), living space(1), installation(1), the international space station(1), international space station(1), thanksgiving dinner(1), don petit(1), heidemarie stefanyshyn piper(1), steve bowen(1), greg chamitoff(1), space shuttle columbia(1), eric boe(1), sandra magnus(1), nasa tv(1), makeover(1), solar alpha rotary joint(1), high resolution picture(1)
9	`http://edition.cnn.com/2008/TECH/space/06/11/google.founder.space/index.html?eref=edition\_space`
	manned space flight, sergey brin, international space station
	space trip(6), google(6), sergey brin(6), fsa(2), russian federal space agency(2), space adventure(2), soyuz spacecraft(2), owen garriot(2), richard garriot(1), nasa(1), eric anderson(1), russian soyuz(1), 5 million(1), russia(1), space advantures ltd(1), spacecraft(1), russian space acency(1), stacey tearne(1), soyuz(1), founder(1)
10	`http://edition.cnn.com/2008/TECH/space/07/31/nasa.mars/index.html?eref=edition\_space`
	nasa, mars exploration, phoenix mars lander, Algorithm
	mar(6), phoenix lander(5), water(4), nasa(4), ice(3), life(2), bill boynton(2), martian water(2), life on mar(1), university of arizona(1), mars odyssey(1), thermal and evolved gas analyzer(1), michael meyer(1), experiment(1), ice sample(1), phoenix(1)
11	`http://edition.cnn.com/2008/TECH/space/05/20/ceac.wheeler/index.html?eref=edition\_space`
	nasa, russian federal space agency, european space agency, international space station, kennedy space center, south pole, mars exploration

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	nasa(3), space farming(2), ceac(2), mar(2), controlled environment agriculture center(1), moon(1), neil armstrong(1), raymond wheeler(1), the international space station(1), life(1), hydroponic system(1), space farm(1), sustainability(1), space habitation(1), plant growing(1), kennedy space center(1), space greenhouse(1), bio regenerative(1), life on moon(1), life support system(1), space shuttle(1), moon landing(1), space life(1), lunar expedition(1), first human mission(1), president bush(1), space colony(1)
12	`http://edition.cnn.com/2009/SHOWBIZ/Music/07/23/brad.paisley.white.house/index.html?eref=edition\_entertainment`
	the white house, brad paisley, country music
	brad paisley(6), white house(6), reception(2), performance(2), michelle obama(2), president(1), country music(1), music performance(1), slavery(1), obama(1), president obama(1), jim robinson(1)
13	`http://edition.cnn.com/2009/SHOWBIZ/Movies/05/29/travolta.mourning/index.html?eref=edition\_entertainment`
	john travolta, jett travolta, denzel washington
	john travolta(6), death(4), denzel washington(4), son(4), bahama(1), the taking of pelham 123(1), distraught(1), death of jett travolta(1), struggle(1), jett travolta(1)
14	`http://edition.cnn.com/2009/US/05/27/mainstreet.taylor.tailor/index.html?eref=edition\_entertainment`
	national economy, clothing, al pacino, jay leno
	ryan taylor(4), taylor the tailor(3), celebrity(2), ryan tailor(1), death(1), son(1), business(1), drobe(1), johnny gill(1), custom tailored client(1), struggle(1), tailor(1), custom tailor(1), bernie mac(1), john travolta(1)
15	`http://edition.cnn.com/2009/SHOWBIZ/Movies/05/28/up.pixar/index.html?eref=edition\_entertainment`
	pixar animation studio, ed asner, cannes film festival

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	up(5), pixar(5), animation(3), john lasseter(2), car(1), paul newman(1), film(1), pixar making(1), animation movie(1), movie(1), house(1), balloon(1), wall e(1), record(1), pixar's success(1), toy story(1), tom hank(1), ed asner(1), success(1), finding nemo(1), pete docter(1)
16	http://edition.cnn.com/2009/SHOWBIZ/Movies/05/25/night.museum.cast/index.html?eref=edition_entertainment
	ben stiller, owen wilson, robin william, smithsonian institution, movie comedy
	owen wilson(5), movie(5), robin william(5), ben stiller(5), night at the museum(4), film(3), comedy(1), night at the museum: battle of the smithsonian(1), smithsonian(1), museum(1), plot(1), filming(1), battle of the smithsonian(1), levy(1), shawn levy(1)
17	http://edition.cnn.com/2009/SPORT/08/02/tiger.woods.buick.lead.letzig/index.html?eref=edition_sport
	tiger wood, john senden, john daly, pga tour, golf
	buick open(5), golf(5), tiger wood(5), tournament(3), michael letzig(3), pga tour(3), warwick hill(3), michigan(1), golf player(1), tiger(1)
18	http://edition.cnn.com/2009/SPORT/08/02/rally.finland.hirvonen.loeb.winner/index.html?eref=edition_sport
	mikko hirvonen, sebastien loeb, kimi raikkonen, finland, motorsport
	mikko hirvonen(6), sebastien loeb(4), victory(3), motorsport(3), rally of finland(2), world rally(2), world championship(1), finland(1), jari matti latvala(1), race(1), championship(1), rally(1), kimi raikkonen(1), finland rally(1), ford focus(1), finnish rally(1), formula one(1)
19	http://edition.cnn.com/2009/WORLD/americas/07/09/costa.rica.zelaya.honduras/index.html?eref=edition_world
	jose manuel zelaya, roberto micheletti, hondura, costa rica

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	hondura(6), jose manuel zelaya(5), roberto micheletti(5), political crisis(4), oscar aria(2), dialogue(2), mediation(2), crisis(2), honduran(1), hugo chavez(1), president oscar aria(1), resolution(1), politics(1), re election(1), jose manuel zelay(1), costa rica(1), military coup(1), negotiation(1)
20	`http://edition.cnn.com/2009/SPORT/08/01/swimming.phelps.cavic.world.championships/index.html?eref=edition\_sport`
	michael phelp, milorad cavic, competitive swimming, sport
	michael phelp(6), swimming(5), world record(4), milorad cavic(3), rome(3), 100 meter butterfly(2), world swimming championship(1), olympic champion(1), 100 meter butterfly record(1), world championship(1), liam tancock(1), cesar cielo(1), championship(1), record(1), swimming championship(1), serbian(1)
21	`http://edition.cnn.com/2009/SPORT/08/01/golf.women.british.ochoa.matthew/index.html?eref=edition\_sport`
	catriona matthew, christina kim, michelle wie, lorena ochoa, lpga tour, women's golf
	catriona matthew(6), royal lytham(3), golf(3), women's british open(3), tournament(2), british open(2), scottish(2), win(2), golfer(2), giulia serga(2), final(1), jiyai shin(1), scottish golfer(1), british open golf(1)
22	`http://edition.cnn.com/2009/SPORT/football/08/01/arsenal.patrick.vieira.inter.milan/index.html?eref=edition\_sport`
	patrick vieira, arsene wenger, arsenal fc, fc internazionale milano, european football
	arsenal(6), arsene wenger(6), patrick vieira(5), football(4), inter milan(3), soccer(3), swoop(2), english premier league(2), premier league(1), mulling(1), vieira(1)
23	`http://edition.cnn.com/2009/SPORT/07/31/f1.ferrari.massa.recovery.brazil.schumacher/index.html?eref=edition\_sport`
	felipe massa, rubens barrichello, ferrari spa, motorsport, formula one racing

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	felipe massa(6), accident(5), hungarian grand prix(3), ferrari(3), recovery(3), brazil(2), motorsport(2), formula one(2), budapest(1), return(1), grand prix(1), surgery(1), rubens barrichello(1), lewis hamilton(1), f1(1), crash(1)
24	http://edition.cnn.com/2009/SPORT/07/21/usain.bolt.powell.gay.athletics/index.html?eref=edition_sport
	usain bolt, tyson gay, asafa powell, jamaica, track and field, olympic game, sport
	usain bolt(6), athletics(5), championship(4), olympic champion(3), london(3), berlin(2), olympic(1), champion(1), track athletic(1), world championship(1), grand prix(1), fine tuning(1), london grand prix(1), world record(1), fastest man(1)
25	http://edition.cnn.com/2009/SPORT/07/21/cricket.sri.lanka.pakistan.colombo/index.html?eref=edition_sport
	danish kaneria, pakistan, sri lanka, cricket
	pakistan(6), cricket(6), sri lanka(4), colombo test(3), danish kaneria(3), khurram manzoor(2), spinner(1), colombo(1), kaneria(1), danish(1), test in colombo(1)
26	http://edition.cnn.com/2009/SPORT/07/21/pacquiao.boxing.fight.cotto/index.html?eref=edition_sport
	manny pacquiao, miguel cotto, philippine, boxing
	miguel cotto(5), manny pacquiao(5), world title(3), boxing(3), fighter(3), wbo(3), boxer(2), mgm grand garden arena(1), wbo fight(1), philipinne(1), mannpacquiao miguelcotto(1), las vega(1), pacquiao(1), juan manuel marquez(1), box(1)
27	http://edition.cnn.com/2009/CRIME/07/29/bernie.madoff/index.html?eref=edition_business
	bernie madoff, bernard l. madoff investment securities llc, charles ponzi
	bernie madoff(5), sec(2), lawyer(2), guilty(1), fraudulent(1), ponzi scheme(1), cnn(1), madoff(1), remorseful(1), autograph(1), investor fraud(1), felon(1), securities and exchange commission(1), butner federal correctional institution(1)
28	http://edition.cnn.com/2009/BUSINESS/07/29/china.ipo/index.html?eref=edition_business

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	economic crisis, financial market, business
	chi(4), china stock(2), economy(2), construction company(1), ipo market(1), construction market(1), ipo(1), composite index(1), economic bubble(1), chinese bours(1), chine state construction engineering(1), stock exchange(1), market(1), state construction(1), fall(1), finance(1), bbmg(1), shanghai(1), jittery(1)
29	http://edition.cnn.com/2009/POLITICS/07/28/states.budget.crunch/index.html?eref=edition_business
	national economy, sales tax, tax policy, marijuana
	marijuana(6), strip club(4), budget(3), tax(3), state(2), gambling(2), legal(1), wisconsin(1), sports lottery(1), financial problem(1), oakland(1), virginia(1), shortfall(1), bingo game(1), united state(1), funding(1), california(1), economy(1), delaware(1), tom ammiano(1), strip(1), new york(1), budget gap(1), utah(1), financial wo(1), bankruptcy(1), state budget(1), finance(1)
30	http://edition.cnn.com/2009/WORLD/americas/07/26/cubal.tough.times/index.html?eref=edition_business
	raul castro, cuba
	cuba(6), raul castro(4), finance(3), revolution(2), castro(2), economic crisis(2), cuban revolution(2), economy embargo(1), president(1), economy(1), global economic crisis(1), austerity(1), economic downturn(1), cuban revolution day(1)
31	http://edition.cnn.com/2009/TECH/07/24/luxury.cellphones/index.html?eref=edition_business
	samsung cell phone, motorola cell phone, nokia cell phone, lg group
	cell phone(4), motorola(3), luxury phone(3), prada(2), luxury(2), recession(1), conspicuous consumption(1), expensive(1), samsung(1), vertu(1), trouble(1), market(1), luxe(1), nuovo(1), market slump(1), lg(1), luxury cell phone(1), emporio armani(1)
32	http://edition.cnn.com/2009/POLITICS/07/22/obama.health.care/index.html?eref=edition_business
	barack obama, health care policy

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	health care(4), obama(4), reform(2), medical treatment(2), health care reform(2), insurance(1), videowatch(1), health insurance(1), american health care policy(1), american government(1), barack obama(1), president obama(1), patient(1), coverage(1), debate(1), democrat(1)
33	http://edition.cnn.com/2009/WORLD/asiapcf/07/22/indonesia.convoy.attack/index.html?eref=edition_business
	jakarta, indonesia, papua province
	indonesia(5), attack(3), mining(2), mining company(2), u.s.(2), fire(2), indonesian(1), fired on(1), fired pt(1), antara(1), papua(1), us(1), bus convoy(1), freeport bus(1), pt freeport(1), employee(1), shooting(1)
34	http://edition.cnn.com/2009/TRAVEL/07/10/pets.fly.airlines/index.html?eref=edition_business
	air travel, midwest air group inc, southwest airlines inc, pet
	pet(5), midwest airline(3), airline(3), cabin(2), cat(2), flight(2), dog(2), pet travel(1), furry(1), tony hoard(1), pet airway(1), passenger(1), immunology(1), seat(1), asthma(1), cabin area(1), allergy(1)
35	http://edition.cnn.com/2009/WORLD/europe/07/10/g8.summit/index.html?eref=edition_business
	barack obama, pope benedict xvi, africa
	g 8(5), obama(5), economy(4), security(2), world hunger(2), africa(2), videowatch(1), president(1), iran(1), united state(1), emission(1), health care(1), legislation(1), kenya(1), global economy(1), climate change(1), barack obama(1), pope benedict(1), environment(1), g 8 talk(1), international security(1), gathering(1), climate(1)
36	http://edition.cnn.com/2009/WORLD/europe/07/05/russia.oil.magnate/index.html?eref=edition_business
	mikhail khodorkovsky, dmitry medvedev

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	mikhail khodorkovsky(5), oil(4), dmitry medvedev(3), medvedev(2), pardon(1), yuko(1), tax evasion(1), russia(1), yukos oil company(1), tax(1), corruption(1), kremlin(1), guilt(1), trial(1), khodorkovsky(1), former oil magnate(1)
37	http://edition.cnn.com/2009/WORLD/africa/08/01/south.africa.suicide/index.html?eref=edition_world
	nelson mandela foundation, nelson mandela, suicide
	south africa(5), nelson mandela(5), suicide(5), policeman(2), police(2), president(1), mandela(1), south african defence force(1), kill(1), prisoner(1), south african policeman(1), guard(1), foundation(1), sergent(1)
38	http://edition.cnn.com/2009/WORLD/americas/08/01/venezuela.radio.stations/index.html?eref=edition_world
	venezuela, hugo chavez
	venezuela(6), radio station(4), hugo chavez(3), globovision(3), diosdado cabello(2), cabello(1), shut down(1), station(1), media crime(1), radio station closed(1), politics(1), radio(1), closure(1), chavez(1), zuloago(1), closing(1)
39	http://edition.cnn.com/2009/WORLD/meast/07/31/iraq.babylon.damage/index.html?eref=edition_world
	babylon, nebuchadnezzar, united nation
	babylon(4), damage(3), iraq(3), u.s. military(3), ancient(3), hanging gardens of babylon(2), us(2), ishtar gate(2), ancient wonder(2), damaged(1), u.s. troops iraq(1), isthar gate(1), hanging garden(1), wonders of the ancient world(1), historical site(1), un(1), unesco(1), military(1), troop(1), history(1), processional way(1), rehabilitate(1)
40	http://edition.cnn.com/2009/WORLD/americas/07/30/ecuador.farc.ties/index.html?eref=edition_world
	farc, rafael correa, raul reye, hugo chavez, ecuador, colombia
	ecuador(7), guerrilla(6), farc(6), diary(4), rebel(4), allegation(3), suarez video(2), raul reye(2), rafael correa(2), drug(2), correa(1), reye(1), revelation(1), colombian(1), colombia(1), money(1), farc guerrilla(1), finance(1)

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

41	http://edition.cnn.com/2009/WORLD/asiapcf/07/30/taliban.code.conduct/index.html?eref=edition_world
	the taliban, afghanistan, pakistan
	taliban(6), suicide attack(5), afghanistan(4), military(2), civilian(2), mujahideen(2), suicide mission(1), rule book(1), code of conduct(1), us(1), iraq(1), u.s. military(1), troop(1), booklet(1), battle(1), islam(1), militant(1), azimi(1), afghan(1), afghan military(1)
42	http://edition.cnn.com/2009/WORLD/americas/07/29/mexico.arrests/index.html?eref=edition_world
	familia michoacan, mexico, drug trafficking, crime
	mexico(5), drug gang(4), la familia michoacana(4), violence(1), bookkeeper(1), cartel(1), armando quintero guerra(1), mexican drug gang(1), war(1), drug cartel(1), arrested(1), michoacan(1), police(1), arrest(1), servando gomez(1), gang(1), killing(1), drug(1)
43	http://edition.cnn.com/2009/WORLD/africa/07/29/nigeria.violence/index.html
	nigeria, islam, christianity
	islamic militant(6), nigeria(4), maiduguri(3), muslim(2), nigerian(1), christian(1), violence(1), fight(1), red cross(1), riot(1), government force(1), arrested(1), fled home(1), battle(1), human rights organization(1)
44	http://edition.cnn.com/2009/WORLD/americas/07/28/honduras.turmoil/index.html?eref=edition_world
	hondura, nicaragua, jose manuel zelaya, roberto micheletti
	hondura(4), san jose accord(3), honduran congress(3), zelaya(2), jose manuel zelaya(2), accord(2), military(2), crisis(2), oscar aria(1), national election(1), president(1), congress(1), political(1), zelava(1), coup(1), amnesty(1), micheletti(1), court(1), election(1), san jose(1), reinstatement(1), roberto micheletti(1), costa rica(1), debate(1)

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

45	http://edition.cnn.com/2009/WORLD/asiapcf/07/28/suukyi.trial/index.html?eref=edition_world
	myanmar, aung san suu kyi
	verdict(6), myanmar(5), trial(4), suu kyi(4), house arrest(3), john william yettaw(3), aung san suu kyi(2), violation(1), subversion trial(1), military(1), election(1), prison(1), yettaw(1), lawyer(1), junta(1), incident(1)
46	http://edition.cnn.com/2009/WORLD/asiapcf/07/25/afghanistan.wrap/index.html?eref=edition_world
	afghanistan, the taliban, international security assistance force, helmand province
	afghanistan(4), taliban insurgent(3), suicide(2), afghan facility(1), taliban(1), ansf(1), terror(1), police building(1), suicide attacker(1), security(1), house arrest(1), military(1), jalalabad(1), strike(1), militant(1), azimi(1), suicide vest(1), military hospital(1), verdict(1), government target(1), bank(1), suu kyi(1), insurgent(1), isaf(1), taliban militant(1), nato(1), british troop(1), myanmar(1), khost(1), suicide attack(1), yettaw(1), u.s. marine(1), medical treatment(1)
47	http://edition.cnn.com/2009/WORLD/europe/04/13/turkey.arrest.television.plot/index.html
	turkey, recep tayyip erdogan
	ergenekon(6), recep tayyip erdogan(5), turkan saylan(5), turkey(5), turkish academic(3), police(3), anatolian agency(2), mehmet haberal(2), akp(2), raid(1), turkish police(1), coup(1), rector(1), academic(1), politics(1), islam(1), arrest(1), baskent university(1), overthrow(1), government(1), university rector(1), prime minister(1)
48	http://edition.cnn.com/2009/WORLD/europe/01/30/davos.erdogan.peres/index.html
	recep tayyip erdogan, shimon pere, gaza, turkey, davo, israel

Table A.2: The 50 CNN news articles with authors and readers assigned tags used as the test data continue.

	davo(6), israel(5), recep tayyip erdogan(4), shimon pere(4), turkey(3), hama(2), erdogan(2), pere(2), world economic forum(2), talk(2), muslim(1), spat(1), palestinian(1), palestine(1), economy(1), military(1), gaza(1), peace(1), polite exchange(1), guilty psychology(1), killing(1), amicable(1)
49	`http://edition.cnn.com/2009/POLITICS/05/29/george.bush.speech/index.html`
	george w. bush, khalid shaikh mohammed, dick cheney, barack obama, michigan
	dick cheney(4), michigan(4), bush(3), interrogation(2), terrorism(2), obama(2), president(1), waterboarding(1), economy(1), george bush(1), assertion(1), socialism(1), speech(1), khalid sheikh mohammed(1), george w. bush(1), book(1), investigation(1), defend(1), administration(1)
50	`http://edition.cnn.com/2009/WORLD/asiapcf/07/10/china.quake/index.html?eref=edition\_world`
	chi, natural disaster, earthquake
	chi(5), yunnan(4), earthquake(4), quake(2), death(1), damage(1), southwestern chi(1), mild(1), xinhua(1), relief(1)

REFERENCES

1. Netcraft, "April 2009 Web Server Survey", 2009, http://news.netcraft.com/.

2. "Metadata — Wikipedia, The Free Encyclopedia", 2010, http://en.wikipedia.org/w/index.php?title=Metadata&oldid=356923186, [Online; accessed 23-April-2010].

3. Kim, H. L., A. Passant, J. G. Breslin, S. Scerri, and S. Decker, "Review and Alignment of Tag Ontologies for Semantically-Linked Data in Collaborative Tagging Spaces", *ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pp. 315–322, IEEE Computer Society, Washington, DC, USA, 2008.

4. "Delicious - Social bookmarking website", http://delicious.com/, [Online; accessed 23-April-2010].

5. "Facebook - Social networking website", http://www.facebook.com/, [Online; accessed 23-April-2010].

6. "Flickr - Photo sharing website", http://www.flickr.com/, [Online; accessed 23-April-2010].

7. "Youtube - Video sharing website", http://www.youtube.com/, [Online; accessed 23-April-2010].

8. Wikipedia, "Delicious (website) — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Delicious_%28website%29&oldid=363874111, [Online; accessed 29-May-2010].

9. Gruber, T., "Ontology (Computer Science) - definition in Encyclopedia of Database Systems", 2008.

10. Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: an On-line Lexical Database*", *Int J Lexicography*, Vol. 3, No. 4, pp.

235–244, January 1990.

11. "Wikipedia, The Free Encyclopedia", 2010, http://en.wikipedia.org/.

12. Miller, G. A., C. Leacock, R. Tengi, and R. T. Bunker, "A semantic concordance", *HLT '93: Proceedings of the workshop on Human Language Technology*, pp. 303–308, Association for Computational Linguistics, Morristown, NJ, USA, 1993.

13. "Zemanta - Content suggestion engine", http://www.zemanta.com/, [Online; accessed 17-May-2010].

14. Yahoo, "Term Extraction Web Service - YDN", http://developer.yahoo.com/search/content/V1/termExtraction.html/, [Online; accessed 02-June-2010].

15. "W3C Semantic Web Activity", http://www.w3.org/2001/sw/, [Online; accessed 1-May-2010].

16. Mathes, A., "Folksonomies - cooperative classification and communication through shared metadata", 2004.

17. Obitko, M., "Semantic Web Architecture", 2007, http://www.obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html, [Online; accessed 29-May-2010].

18. W3C, "RDF - Semantic Web Standards", http://www.w3.org/RDF/, [Online; accessed 02-June-2010].

19. Grigoris Antoniou, F. V. H., *A semantic Web primer*, 2004.

20. "OWL Web Ontology Language website", www.w3.org/TR/owl-features/, [Online; accessed 23-April-2010].

21. "Web Service Modeling Ontology (WSMO)", 2005, http://www.wsmo.org/TR/d2/v1.2/#ontologies/, [Online; accessed 02-June-2010].

22. Reeve, L. and H. Han, "Semantic Annotation for Semantic Social Networks Using Community Resources", *AIS SIGSEMIS Bulletin*, Vol. 2, No. 3-4, pp. 52–56, 2005.

23. Qi, X. and B. D. Davison, "Web page classification: Features and algorithms", *ACM Comput. Surv.*, Vol. 41, No. 2, pp. 1–31, 2009.

24. "Oxford English Dictionary website", 2010, `http://www.oed.com/`, [Online; accessed 06-June-2010].

25. Francis, W. N. and H. Kucera, "Brown corpus manual", Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.

26. "Sketch Engine - Corpus query system", `http://www.sketchengine.co.uk/`, [Online; accessed 1-May-2010].

27. "JustTheWord - Collocation resource", `193.133.140.102/JustTheWord/`, [Online; accessed 1-May-2010].

28. Lee, Y. K. and H. T. Ng, "An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation", *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pp. 41–48, Association for Computational Linguistics, Morristown, NJ, USA, 2002.

29. Black, E. W., "An experiment in computational discrimination of English word senses", *IBM J. Res. Dev.*, Vol. 32, No. 2, pp. 185–194, 1988.

30. Veronis, J. and N. M. Ide, "Word sense disambiguation with very large neural networks extracted from machine readable dictionaries", *Proceedings of the 13th conference on Computational linguistics*, pp. 389–394, Association for Computational Linguistics, Morristown, NJ, USA, 1990.

31. TufiŞ, D., R. Ion, and N. Ide, "Fine-grained word sense disambiguation based on parallel corpora, word alignment, word clustering and aligned wordnets", *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, p. 1312, Association for Computational Linguistics, Morristown, NJ, USA, 2004.

32. Pantel, P., "Inducing ontological co-occurrence vectors", *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 125–132, Association for Computational Linguistics, Morristown, NJ, USA, 2005.

33. Lesk, M., "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone", *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26, ACM, New York, NY, USA, 1986.

34. McCarthy, D. and J. Carroll, "Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences", *Comput. Linguist.*, Vol. 29, No. 4, pp. 639–654, 2003.

35. McCarthy, D., R. Koeling, and J. Weeds, "Ranking WordNet Senses Automatically", Technical report, 2004.

36. Galley, M. and K. McKeown, "Improving word sense disambiguation in lexical chaining", *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pp. 1486–1488, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

37. Shi, T., S. Jiao, J. Hou, and M. Li, "Improving Keyphrase Extraction Using Wikipedia Semantics", *IITA '08: Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application*, pp. 42–46, IEEE Computer Society, Washington, DC, USA, 2008.

38. Hu, X. and B. Wu, "Automatic Keyword Extraction Using Linguistic Features", *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pp. 19–23, IEEE Computer Society, Washington, DC, USA, 2006.

39. Matsuo, Y. and M. Ishizuka, "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information", *International Journal on Artificial Intelligence Tools*, Vol. 13, p. 2004, 2003.

40. Kumar, N. and K. Srinathan, "Automatic keyphrase extraction from scientific documents using N-gram filtration technique", *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, pp. 199–208, ACM, New York, NY, USA, 2008.

41. Li, X., X. Wu, X. Hu, F. Xie, and Z. Jiang, "Keyword Extraction Based on Lexical Chains and Word Co-occurrence for Chinese News Web Pages.", *ICDM Workshops*, pp. 744–751, IEEE Computer Society, 2008, `http://dblp.uni-trier.de/db/conf/icdm/icdmw2008.html\#LiWHXJ08`.

42. Turney, P. D., "Learning Algorithms for Keyphrase Extraction", *Inf. Retr.*, Vol. 2, No. 4, pp. 303–336, 2000.

43. Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction", *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence*, pp. 668–673, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

44. Wang, J. and H. Peng, "Keyphrases Extraction from Web Document by the Least Squares Support Vector Machine", *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 293–296, IEEE Computer Society, Washington, DC, USA, 2005.

45. Ercan, G. and I. Cicekli, "Using lexical chains for keyword extraction", *Inf. Process. Manage.*, Vol. 43, No. 6, pp. 1705–1714, 2007.

46. Li, X., X. Wu, X. Hu, F. Xie, and Z. Jiang, "Keyword Extraction Based on Lexical Chains and Word Co-occurrence for Chinese News Web Pages.", *ICDM Workshops*, pp. 744–751, IEEE Computer Society, 2008, `http://dblp.uni-trier.de/db/conf/icdm/icdmw2008.html\#LiWHXJ08`.

47. Turney, P., "Learning to Extract Keyphrases from Text", 1999.

48. "The NETEASE - Chinese news portal", `http://www.163.com`, [Online; accessed 17-May-2010].

49. Ponzetto, S. P. and M. Strube, "Deriving a large scale taxonomy from Wikipedia", *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pp. 1440–1445, AAAI Press, 2007.

50. Auer, S., C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data", *In 6th Intl Semantic Web Conference, Busan, Korea*, pp. 11–15, Springer, 2007.

51. Cyc, "OpenCyc for the Semantic Web", `http://sw.opencyc.org/`, [Online; accessed 02-June-2010].

52. Ruiz-casado, M., E. Alfonseca, and P. Castells, "Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets", *In: Proceedings of the Atlantic Web Intelligence Conference, AWIC-2005. Volume 3528 of Lecture Notes in Computer Science*, pp. 380–386, Springer Verlag, 2005.

53. "Help:Infobox — Wikipedia, The Free Encyclopedia", 2010, `http://en.wikipedia.org/wiki/Help:Infobox/`, [Online; accessed 23-April-2010].

54. Bizer, C., T. Heath, K. Idehen, and T. B. Lee, "Linked data on the web (LDOW2008)", *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 1265–1266, ACM, New York, NY, USA, 2008.

55. Shvaiko, P. and J. Euzenat, "A Survey of Schema-Based Matching Approaches", *Journal on Data Semantics*, Vol. 4, pp. 146–171, 2005.

56. Suchanek, F. M., G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge", *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, ACM Press, New York, NY, USA, 2007.

57. Wu, F. and D. S. Weld, "Autonomously semantifying wikipedia", *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 41–50, ACM, New York, NY, USA, 2007.

58. Wu, F. and D. S. Weld, "Automatically refining the wikipedia infobox ontology",

WWW '08: Proceeding of the 17th international conference on World Wide Web, pp. 635–644, ACM, New York, NY, USA, 2008.

59. Group, T. S. N. L. P., "The Stanford Parser: A statistical parser", 2010, `http://nlp.stanford.edu/downloads/lex-parser.shtml`.

60. "Google Search Engine", `http://www.google.com/`, [Online; accessed 17-May-2010].

61. "Yahoo Search Engine", `http://www.yahoo.com/`, [Online; accessed 17-May-2010].

62. Moldovan, D. I. and R. Mihalcea, "Using WordNet and Lexical Operators to Improve Internet Searches", *IEEE Internet Computing*, Vol. 4, No. 1, pp. 34–43, 2000.

63. "Ask.com Search Engine - Better Web Search", `http://www.ask.com/`, [Online; accessed 17-May-2010].

64. "MetaCrawler Web Search Engine", `http://www.metacrawler.com/`, [Online; accessed 17-May-2010].

65. "Clusty - Metasearch Engine", `http://clusty.com/`, [Online; accessed 17-May-2010].

66. "Hakia - Semantic search engine", `http://www.hakia.com/`, [Online; accessed 06-June-2010].

67. "Powerset - Semantic search engine", `http://www.powerset.com/`, [Online; accessed 06-June-2010].

68. Hawker, T. and M. Honnibal, "Improved Default Sense Selection for Word Sense Disambiguation", *Proceedings of the Australasian Technology Workshop, Sydney*, 2006.

69. Cilibrasi, R. L. and P. M. B. Vitanyi, "The Google Similarity Distance", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 19, No. 3, pp. 370–383, 2007, `http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=4072748`.

70. Gligorov, R., W. T. Kate, Z. Aleksovski, and F. van Harmelen, "Using Google dis-

tance to weight approximate ontology matches", *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pp. 767–776, ACM Press, New York, NY, USA, 2007.

71. Seco, N., T. Veale, and J. Hayes, "An Intrinsic Information Content Metric for Semantic Similarity in WordNet", *ECAI 2004*, 2004, http://eden.dei.uc.pt/~nseco/ecai2004b.pdf.

72. THEwikiStics, "Wikimedia statistics", 2009, http://wikistics.falsikon.de/2009/wikipedia/en/.

73. "The Apache Lucene website", http://lucene.apache.org/, [Online; accessed 02-June-2010].

74. "CNN.com a news article", 2010, http://edition.cnn.com/2009/WORLD/asiapcf/07/28/suukyi.trial/index.html?eref=edition_world/.

75. J. Baldridge, T. M. and G. Bierner, "The opennlp maximum entropy package", Technical report, 2002, http://opennlp.sourceforge.net/.

76. "Netbeans - Integrated development environment", http://www.netbeans.com/, [Online; accessed 1-May-2010].

77. "Eclipse - Integrated development environment", http://www.eclipse.org/, [Online; accessed 1-May-2010].

78. "PostgreSQL - Object-relational database", http://www.postgresql.org/, [Online; accessed 1-May-2010].

79. Bou, B., "The WordNet SQL Builder website", http://lucene.apache.org/, [Online; accessed 02-June-2010].

80. "The Jena Semantic Web Framework website", http://jena.sourceforge.net/, [Online; accessed 02-June-2010].

81. "Wikimedia Downloads — Wikipedia, The Free Encyclopedia", 2010, http://download.wikimedia.org/enwiki/, [Online; accessed 23-April-2010].

82. "MediaWiki API — Wikipedia, The Free Encyclopedia", 2010, http://en.wikipedia.org/w/api.php/, [Online; accessed 23-April-2010].

83. "Luke - Lucene Index Toolbox website", http://code.google.com/p/luke/, [Online; accessed 1-May-2010].

84. "Aduna Autofocus - Desktop search application", http://www.aduna-software.com/, [Online; accessed 17-May-2010].

85. Aduna, "OpenRDF - RDF repository", http://www.openrdf.org, [Online; accessed 17-May-2010].

86. Berners-Lee, T., "Notation3 (N3): A readable RDF syntax", http://www.w3.org/DesignIssues/Notation3/, [Online; accessed 17-May-2010].

87. "Formula One — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Formula_One&oldid=365406542, [Online; accessed 02-June-2010].

88. "Michael Schumacher — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Michael_Schumacher&oldid=365648398, [Online; accessed 02-June-2010].

89. "Mika Hakkinen — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Mika_H%C3%A4kkinen&oldid=363326572, [Online; accessed 02-June-2010].

90. "Rubens Barrichello — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Rubens_Barrichello&oldid=365251026, [Online; accessed 02-June-2010].

91. "Ferrari — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Ferrari&oldid=365256907, [Online; accessed 02-June-2010].

92. "Ross Brawn — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Ross_Brawn&oldid=363596570, [Online; accessed 02-June-2010].

93. Nguyen, T. D. and M.-Y. Kan, "Keyphrase Extraction in Scientific Publications.", Goh, D. H.-L., T. H. Cao, I. Slvberg, and E. M. Rasmussen (editors), *ICADL*, Vol. 4822 of *Lecture Notes in Computer Science*, pp. 317–326, Springer, 2007, http://dblp.uni-trier.de/db/conf/icadl/icadl2007.html\#NguyenK07.

94. Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-manning, and G. Inc, "Kea: Practical automatic keyphrase extraction", *In Proceedings of the 4th ACM conference on Digital Libraries*, pp. 254–255, 1998.

95. Al-Khalifa, H. S. and H. C. Davis, "Folksonomies versus Automatic Keyword Extraction: An Empirical Study", *IADIS INTERNATIONAL JOURNAL ON COMPUTER SCIENCE AND INFORMATION SYSTEMS (IJCSIS)*, Vol. 1, pp. 132–143, 2006.

96. "The Common Tag - Tagging format", http://www.commontag.org/, [Online; accessed 02-June-2010].