THE VALUE OF EVERYTHING: RANKING AND ASSOCIATION WITH ENCYCLOPEDIC KNOWLEDGE

Kino High Coursey, B.A, M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2009

APPROVED:

Rada F. Mihalcea, Major Professor
Paul E. Tarau, Committee Member
William E. Moen, Committee Member
Douglas B. Lenat, Committee Member
Ian Parberry, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Michael Monticino, Dean of the Robert B. Toulouse School of Graduate Studies

Coursey, Kino High. The Value of Everything: Ranking and Association with Encyclopedic Knowledge. Doctor of Philosophy (Computer Science and Engineering), December 2009, 178 pp., 27 tables, 43 figures, references, 117 titles.

This dissertation describes WikiRank, an unsupervised method of assigning relative values to elements of a broad coverage encyclopedic information source in order to identify those entries that may be relevant to a given piece of text. The valuation given to an entry is based not on textual similarity but instead on the links that associate entries, and an estimation of the expected frequency of visitation that would be given to each entry based on those associations in context. This estimation of relative frequency of visitation is embodied in modifications to the random walk interpretation of the PageRank algorithm.

WikiRank is an effective algorithm to support natural language processing applications. It is shown to exceed the performance of previous machine learning algorithms for the task of automatic topic identification, providing results comparable to that of human annotators. Second, WikiRank is found useful for the task of recognizing text-based paraphrases on a semantic level, by comparing the distribution of attention generated by two pieces of text using the encyclopedic resource as a common reference. Finally, WikiRank is shown to have the ability to use its base of encyclopedic knowledge to recognize terms from different ontologies as describing the

same thing, and thus allowing for the automatic generation of mapping links between ontologies.

The conclusion of this thesis is that the "knowledge access heuristic" is valuable and that a ranking process based on a large encyclopedic resource can form the basis for an extendable general purpose mechanism capable of identifying relevant concepts by association, which in turn can be effectively utilized for enumeration and comparison at a semantic level.

ACKNOWLEDGEMENTS

I want to thank Dr. Rada Mihalcea, for her dedication to my progress as a researcher and this work. Her constant encouragement and collaboration and passion make her both challenging and inspirational. Her ability to combine imagination with rigor has been the best possible influence imaginable. I also want to give my deepest thanks to Drs. Paul Tarau, Douglas Lenat, William Moen and Michael Witbrock for accepting the invitation to be on the committee. I especially wish to thank Dr. Tarau for inviting me to visit the UNT NLP group (which lead to this) and Dr. Lenat, whose work helped inspire in me a continuous interest in artificial intelligence (which lead to this).

I want to thank those who help shape my life. First, my parents—Claude High and Nadine Harper-Spencer; without them, I would not exist and whose love I have always felt. Next, the grandparents who raised me—T.C. and Lolee Coursey; they showed the importance of pursuing your true desire, and without them I would not be who I am now. My siblings—Claude III, Kim, Danielle & Marissa for being people I am always proud of. To Roger Bright and Art Benson giving me run of their labs and for showing their enthusiasm for science and technology, and Joe Kuban, who showed that you can teach evolution at a Catholic school. For Diane G. Emory, a well missed confidant who in years past was responsible for lots of after dinner debate, discussing philosophy and long chains of logic. And to the Pirzchalski family of Stan, Lisa, Lynda, Cameron and Jennie for adopting me. A special thanks to Karen Pirzchalski for showing all those who associated with her what angels are made of. I wish they all could be here.

Most Important Person Award: Susan Pirzchalski. For sitting next to me in logic class and staying for over twenty-five years, sharing dreams, being supportive in every way imaginable and showing super-human restraint (especially when editing this document). You show me every day that I made the right connection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

CONTEXT, MOTIVATION, AND GOALS

> What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.
>
> — *Herbert Simon, 1971*

An exponentially increasing amount of text and information in our world has created the need to find methods that can annotate, organize, and analyze these objects in meaningful ways. To aid in this task, a number of broad coverage knowledge sources like encyclopedias and other reference sources currently exist and more are constantly being generated. The most common use of these knowledge sources is to orient humans and provide navigation services to the complex semantic space they find themselves in. However, another possibility would be to reuse these same resources to aid computers in tasks requiring intelligence, especially for determining the most important associations of a given object and using that information for a particular task.

The primary focus of my research was to explore the use of a form of encyclopedic knowledge to aid automatic tasks. To do this I developed a method to implement a context sensitive simulation of a visitation process applied to a graph of encyclopedic knowledge. The result of the process is an estimate of the frequency of accessing each entry (and by extension each concept) in an encyclopedia. The relative visitation values assigned offer what I think of as a "knowledge access heuristic": that the more often a piece of knowledge or concept is accessed the more important it is to that context. Using such a visitation simulation one can suggest better allocation of processing resources, perform analysis and inferences based on the distribution of knowledge

access (allowing topic identification), and analyze the way the simulated knowledge access varied based on different stimuli (a form of semantic or topical similarity).

1.1 Motivation

To operate in a complex world, an intelligent system (human or computational) requires not only a great deal of knowledge about the world, but also a method for how to allocate its processing capacity across the knowledge it has access to. A great deal of effort has been expended to collect such knowledge, categorize, and index it in ways by both professionals and motivated groups of individuals to aid in not only accessing primary information but also related knowledge and materials. Without a method of both finding the material and estimating its relative importance in context we face the condition of potentially being positioned in an information space without orientation, landmarks, or navigation aids, under the assumption that the user will possess the background knowledge "to find their way." This is not necessarily a good assumption for either man or machine. Given that situation, is it possible to utilize the network of an encyclopedic knowledge source as a guide for what concepts are important and relevant to understanding a given text?

The ability to identify which topics are relevant to a particular text would have numerous applications. Full free text indexing is one method of indexing items with search engines commonly used on both the global Internet and local intranets. In the case of the Internet, indexing billions of documents and then using an ambiguous natural language query can return a very large, hard-to-absorb list. The other approach (examined here) is to develop a method that can identify which core and associated concepts are relevant in each document. The ability to do so accurately and automatically would provide some means of topic indexed search and navigation. A ranked list (similar to a "suggested reading list") can identify related topics that aid

understanding by either providing background or context, and enable semantic browsing by providing both meaningful tags and topical analysis.

In addition to human uses, computers can utilize such a list as a natural semantic dimension to index both knowledge and documents. Indeed, it can provide a simple way for applications to know what a text being analyzed is "about," and utilize this for comparison purposes. It would also allow recognition of the similarity and differences between texts at the topic level. This leads to the ability to recognize paraphrases. Word sense disambiguation could also be aided by having knowledge of the overall topic and context of a text. The same would also apply whenever additional context would be useful in the application of knowledge.

The method of estimating frequency of knowledge access is a modification of the random walk interpretation of the PageRank (Brin and Page, 1998) algorithm applied to a network of encyclopedic articles. The method, here called *WikiRank*[1], is able to identify the importance of entries not only directly mentioned by a text, but also those that have a strong association with the input material. Encouraging results were produced using this method in the areas of topic identification and selecting the annotations humans would use to describe a given text. A set of semantic similarity metrics extend WikiRank into a means of recognizing textual paraphrases. WikiRank is also explored as a method to map and link ontologies using the encyclopedia as a common base.

[1] Not to be confused nor associated with the Wikipedia statistics monitoring site http://www.wikirank.com

1.2 Questions and Hypothesis

The primary question of this research can be stated as: Is it possible to utilize the network of an encyclopedic knowledge source as a guide for what concepts are important and relevant to understanding a given text? This naturally led to my primary hypothesis: A process called WikiRank, which performs biased PageRanking applied to an encyclopedic knowledge source, can dynamically assign relevancy values to the elements of that source, and these values can be an automated source for human-like associations given a starting set of entries defining the input.

This led to several secondary application-level hypotheses:

When combined with a suitable tagging engine, biased PageRanking can be used to identify additional encyclopedic entries relevant to a particular text.

- *It will be possible to compare two objects on a semantic level defined by the encyclopedic reference by using the set of encyclopedic entries returned by the algorithm for each object.*

- *By combining the textual processing and semantic comparison capabilities it will be possible to recognize text that substantially imparts the same information at a semantic level.*

From these hypotheses several research questions presented themselves:

- *How can one judge how WikiRank performs? What ways exist to verify each hypothesis? What natural basis for comparison (if any) exists?*

  There are a number of naturally occurring test cases that can be found either in standardized test sets or associated within the data being studied. For instance, the text of an encyclopedic article can be used to test

processing if that article is first removed from the encyclopedic network, and the system suggestions can be compared to the original human-generated annotation.

- *How will the system react to the noise caused by ambiguity?*

  Some tasks offer direct linkage between inputs and the encyclopedic reference, while others require processing ambiguous natural language. Can the system use the bulk of its knowledge encoded as associations to identify the proper concepts (similar to Word Sense Disambiguation) even with potential noise? Can it operate with the ambiguity inherent in certain human generated input and knowledge sources?

- *Is WikiRank (when combined with sufficient knowledge) broad enough to perform different tasks?*

  I will be using the same basic algorithm for a number of purposes, across tasks that are both specific and broad. Does the algorithm return results when used with a sufficiently broad knowledge source across these multiple domains? Is the same set of similarity metrics sufficient?

- *Can a conceptually simple general-purpose mechanism for applying encyclopedic knowledge to associational tasks be competitive? Does a large quantity of broad knowledge coupled with such a general algorithm compete with methods tailored to a specific domain?*

  It will be shown that the algorithm (biased random walks over a knowledge graph) and associated methods of tagging, similarity comparison, and machine learning are all relatively well known. Instead of a method tailored to for one specific task, WikiRank provides a general

framework where additional knowledge can be added to improve performance.

CHAPTER 2

OVERALL RESEARCH AGENDA

The goal of my research was to develop a means to estimate the relevance of each entry in an encyclopedic knowledge source and be able to utilize the estimates for NLP tasks. On the one hand is the requirement to test the true usefulness of the relevancy values given to an individual entry. On the other is the requirement to test ways of aggregating the specific values assigned to specific entries into higher-order values useful for comparing two or more objects with each other based on the encyclopedic knowledge accessed by each individually. The usefulness of the second method depends on the quality of the first. The better the system can accurately estimate the relevancy of knowledge and provide plausible indexing terms, the better the system can also recognize similarity and difference between objects using those estimates of knowledge relevancy, and by extension the better its use as a feature source for supervised systems. The goal therefore is to in some way quantify the applicability in these areas. Since there exists a dependency in the system elements and required research, the initial focus developed and characterized the ability to rank the relevancy of encyclopedic knowledge to a particular text. Initial experiments quantified the ability of the system to return plausible (to humans) ranked lists of terms. Intermediate experiments measured the ability of various similarity metrics to use the information provided to correlate with human standards of textual similarity. Later experiments tested the ability of a supervised machine learning system to use the similarity metrics as features for classifying texts or identifying formal terms in ontologies linked to encyclopedic references as being highly similar or not.

2.1 Development of Biased Ranking for Topic Identification

The initial set of experiments explored and evaluated ways of interfacing text and knowledge sources with the bias function and additional graph structures. This included usage as an unsupervised method that delivered ranked lists of plausible terms and examined unsupervised methods that recognized similarity using such lists.

It will be shown that WikiRank produces useful results when compared with other methods, and compares favorably with human performance at identifying relevant topics. As such it should make a good feature source in a supervised system (possibly combined with other knowledge sources) and useful input to similarity evaluation functions.

2.2 Text-to-Text Comparisons Using Similarity of Identified Topics

Currently a number of low-level methods exist to identify when two pieces of text are similar on a lexical level. It would be especially useful to recognize when two pieces of text (or entities) are similar at a higher level. Given initial positive results of the dynamic ranking process applied to the task of topic identification, the focus for further research was on extending WikiRank in ways that could utilize the encyclopedic knowledge ranking for other language and knowledge processing tasks. The primary theme involved developing the capability to recognize semantic similarity through the biased ranking results.

The encyclopedic ranking system, as constructed, does not process detailed semantic level relationships. A link between articles in Wikipedia indicates roughly "The author of the article found the following target article likely relevant to understanding this source article." The biased ranking does, however, generate a ranking representing a "gist" over the universe of topics provided by the encyclopedia, based on the relative

visitation frequency that would be given to each article if the browsing process is run to infinity. I first examined ways the estimation of the relative distribution of knowledge access can be compared, along with text-based variants. Then I evaluated system performance by discriminating similar/dissimilar textual inputs using textual paraphrases and recognizing correct answers to short answer questions. In both cases the system performance is competitive with system developed specifically for these tasks.

2.3 Comparison of Ontology Terms Using Similarity Metrics

Given the results of applying background knowledge to text, the focus turned to checking the ability to recognize similar concepts in more formal setting, similar to that found in semantic web applications. In the first experiment, terms from different ontologies that have vetted links to Wikipedia (for ranking) and WordNet (for verification) were used. This evaluation posed the problem of performing the pairwise comparison of large subsets of two large ontologies. Processing the dataset required embedding and extending WikiRank and similarity computation process in a framework called MapReduce to meet the throughput requirements. The parallel process developed illustrates an example of applying encyclopedic knowledge on a large scale.

The final set of experiments tested the ability of the system to recognize matching concepts in two formal ontologies by reading textual descriptions of their elements and comparing the similarity of the informational access patterns produced by the biased ranking process. The system demonstrated the ability to recognize that concepts are the same by making reference to the background information contained in the encyclopedic knowledge source applied to textual input. This could be useful for matching ontologies provided a textual description can be generated for entries, or matching

individual textual elements into ontologies in a manner similar to the topic identification task.

CHAPTER 3

BACKGROUND MATERIALS AND RESOURCES

> Imagine a world in which every single person on the planet
> is given free access to the sum of all human knowledge.
> That's what we're doing.
>
> *– Jimmy Wales, founder, Wikimedia Foundation 2004*

Before delving into the body of the research, the next two chapters will provide an introduction to the larger context, the materials I used, and the method of WikiRank itself.

3.1 Reference Works and Knowledge Sources

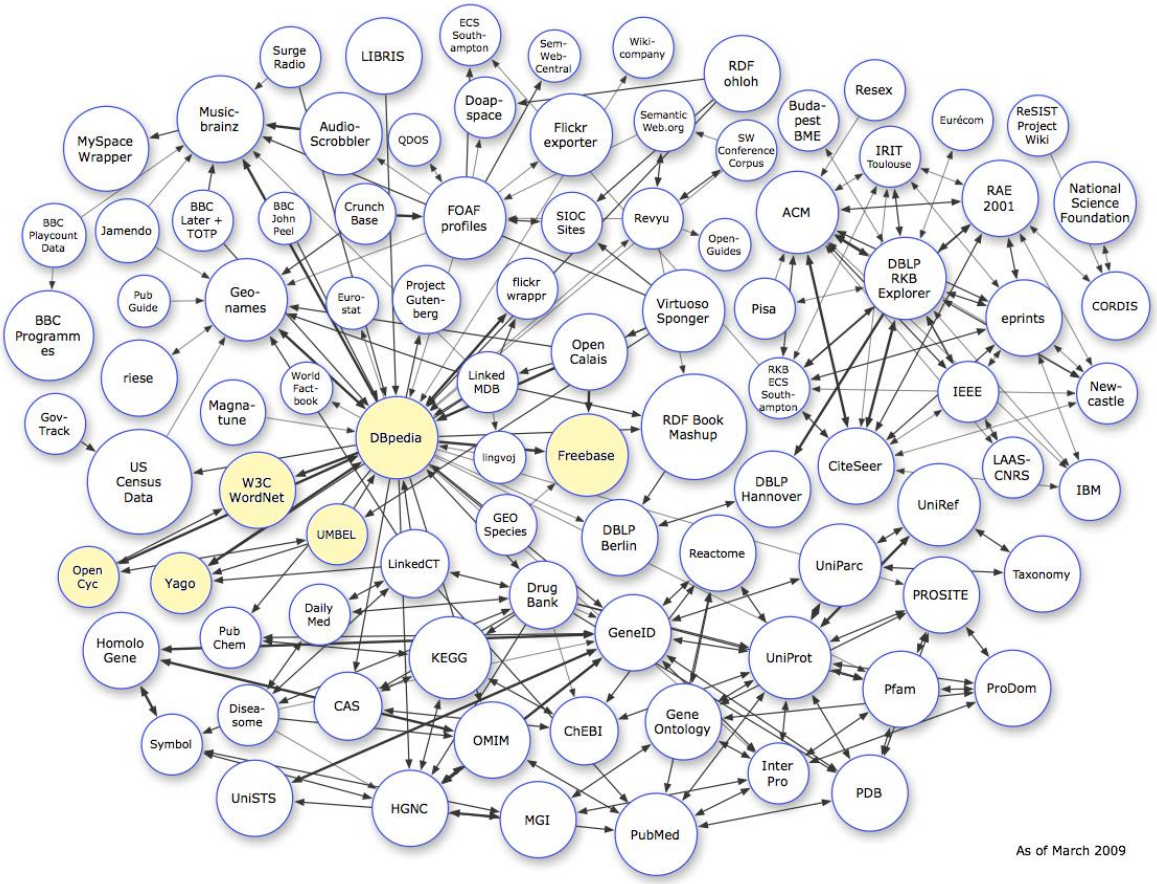

Figure 3.1 Encyclopedic Knowledge Sources in the Linking Open Data Project

Figure 3.1 illustrates the influence of several encyclopedic knowledge sources in linking various semantic web ontologies together[2,3]. A large number of the domain specific ontologies use them as a central connection point to connect with one another, as well as acting as a well connected entry point. In this section I describe how several of these resources are interrelated and how they connect to the datasets used in my research.

3.1.1 Wikipedia

Wikipedia[4] is a free online encyclopedia, representing the output of a continuous collaborative effort of a large number of volunteer contributors. Virtually any Internet user can create or edit a Wikipedia webpage, and this "freedom of contribution" has a positive impact on both the quantity (fast-growing number of articles) and the quality (potential mistakes are quickly corrected within the collaborative environment) of this online resource.

Wikipedia has grown to become one of the largest online repositories of encyclopedic knowledge, with millions of articles available for a large number of languages. Currently (as of spring 2009), Wikipedia editions are available for more than 200 languages, with a number of entries varying from a few pages to more than three million articles per language.

The basic entry in Wikipedia is an *article* (or *page*), which defines and describes an entity or an event, and consists of a hypertext document with hyperlinks to other pages within or outside Wikipedia. The role of the hyperlinks is to guide the reader to pages that provide additional information about the entities or events mentioned in an article.

---

[2] http://www.linkeddata.org/
[3] http://www4.wiwiss.fu-berlin.de/bizer/pub/lod-datasets_2009-03-27.html
[4] http://en.wikipedia.org

Each article in Wikipedia is uniquely referenced by an identifier, which consists of one or more words separated by spaces or underscores, and occasionally a parenthetical explanation. For example, the article for *bar* with the meaning of "*counter for drink*" has the unique identifier *bar_(counter).*

The July 2008 version of the English Wikipedia used for the experiments conducted so far consists of about 2.75 million articles. In addition to articles, Wikipedia also includes a large number of categories which represent topics that are relevant to a given article. This same Wikipedia version includes more than 385,000 such categories. The category links are organized hierarchically, and vary from broad topics such as "history" or "games" to highly focused topics such as "military history of South Africa during World War II" or "role-playing game publishing companies."

Other projects have sought to access the information in Wikipedia. One of the more notable ones is the Semantic Wikipedia project[5]. Their goal is to provide the ability to manually annotate Wikipedia articles with semantic information that could be automatically extracted. In this way, while editing the Wikipedia, information needed to transform it into an ontology is captured. However, this process is not automatic, depending on the human editing of every article.

DBpedia[6] (Auer et al., 2007) is a dataset of structured information extracted from Wikipedia represented in resource description framework (RDF) format and made available for semantic web applications. As of spring 2009, it consists of 274 million RDF triples on 2.6 million instances, with links to external web pages, other RDF datasets, Wikipedia categories and YAGO (yet another great ontology) categories. At

[5] http://www.semanticwiki.jp
[6] http://dbpedia.org

the time of writing, DBpedia plays an important role in the Linking Open Data project shown in Figure 3.1.

Freebase[7] (Bollacker et al., 2007) is a "wiki"-modeled collaborative knowledge base. It contains information harvested from a number of sources including Wikipedia. The information is provided under the Creative Commons license, and the data is available for use as a database dump, semantic web accessible RDF endpoint, or via an application program interface (API). Freebase provides an interface for non-programmers to add metadata to entities in the system.

3.1.2 WordNet

WordNet[8] (Miller et al., 1990) is a semantic lexicon intended to model the lexical knowledge of English of a typical English speaker, and was developed by the Cognitive Science Laboratory of Princeton University. It was designed to provide easy software access to the lexicon. Links to nouns, verbs, adjectives and adverbs are grouped into sets of synonyms that represent the same core concept called *synsets*. For example, the concept of "dog" may include "dog," "domestic dog," and "Canis familaris" in its shared synset. Each synset has associated with it a description called its *gloss*. As an example, the gloss for dog is "(a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) 'the dog barked all night.'" Also associated with each synset are semantic links between synsets. Thus WordNet provides a semantic network defined at the lexical level. Ambiguous words with multiple meanings are included in multiple synsets. As of Version 3.0, WordNet contains 11798 nouns in 82115 synsets, and has 206941 word-sense pairs defined. The hypernym/hyponym relation encodes the

[7] http://www.freebase.com
[8] http://wordnet.princeton.edu/

relation between sub- and super- concepts, and forms a directed acyclic graph rooted at the primary source node "Entity."

The relationships are defined by linguistic criteria. An example would be hyponym which is defined when "native speakers accept sentences constructed from such frames as 'An x is a (kind of) y.'" While accepted linguistically, some cases occur where strict subsumption does not hold, primarily due to overloading of the hypernym/hyponym/ISA relations. The WordNet sections of (de Melo et al., 2008b) and (Guarino, 1998) provide examples of inconsistencies caused by this overloading.

3.1.3 Cyc

The Cyc project (Lenat et al., 1983; Lenat, 1995) was initiated to overcome the brittleness issue of existing expert systems by providing computers with a store of formally represented general commonsense knowledge in which domain-specific expert knowledge could also be embedded and to which programs can draw on when faced with situations partially or wholly outside their original domain. Human "Cyclists" have been manually entering knowledge into the Cyc knowledge base in various ways since 1983. As of spring 2009, approximately 900 person-years of effort was used to represented over two million facts and rules about more than 300K entities and types of entities, 26K relationships, and close to three million assertions and rules.

The inference engine supports deductive, abductive and inductive inference over the knowledge base by integrating more than 700 specialized reasoners for commonly occurring classes of sub-problems. The knowledge base (KB) is intended to support unforeseen (and even unforeseeable) future knowledge representation and reasoning tasks by providing facilities to represent and reason over first and n^{th}-order predicate logic (Ramachandran, 2005). It also supports the ability to segment the knowledge into

local inheritable contexts (Guha, 1991) called *microtheories*. Microtheories can be mutually consistent with their parent contexts but can contradict their siblings. This allows multiple, possibly contradictory, viewpoints to be represented simultaneously.

In addition to the logical ontology/KB, Cyc contains Natural Language (NL) processing tools and information. A well-developed English lexicon, containing the knowledge about syntax and semantics, allows it to translate between its formal representations and English. Cyc can provide links to WordNet.

The ontology provides a wide range of categories in order to be relevant across many domains. A fundamental distinction is made between collections and individuals. This is relevant to because different domains have different units of focus. Specific individual people, place and events tend to be the focus of history, while science tends to express information as properties ascribed to entire classes or conditions. The logical predicates provided for a domain indicate which level knowledge is expressed at.

An example of the type of data included in the Cyc KB is shown below for the concept of "Lobster":

Figure 3.2 Cyc Lobster Example

The knowledge is represented in the language CycL, which is a higher-order predicate calculus based language. Each assertion in CycL is made in a context, or a microtheory. The system of microtheories allows Cyc to process competing hypothesis or theories that may contain contradictions or be completely fictional. Cyc also provides an overall

framework which can allow general theorem proving along with optimized reasoning modules. Using the optimized modules Cyc can reason about collections, relationships, and sentences in CycL. One sentence in CycL can be about another sentence. Each term in CycL is unambiguous, along with each CycL sentence.

Cyc also provides a set of tools to process natural language and map natural languages like English into CycL terms and sentences, along with information to represent the properties of words, like part of speech or multi-word phrases. Cyc's concepts are focused on commonsense reasoning and thus do not map precisely to normal word senses. There are many terms in Cyc necessary to perform commonsense reasoning yet have no singular corresponding word or wordsense. This is to be expected since Cyc seeks to capture the information not commonly transmitted between agents exhibiting commonsense. For instance, Cyc makes a distinction between hurricane as an object ("Hurricane Gilbert moved northeast…") and as an event ("During Hurricane Gilbert several alerts were issued."). This distinction is important for commonsense reasoning, yet does not have separate word senses in an ontology such as WordNet. This ties in with the original goal of Cyc providing the knowledge required to allow a computer to interpret an encyclopedia.

Related to Cyc are OpenCyc and UMBEL. OpenCyc[9] includes a version of the core definitions of Cyc along with an inference engine. UMBEL[10] (upper mapping and binding exchange layer) is an ontology defined as a reduced subset extracted from the concepts and relations in OpenCyc. UMBEL's primary purpose is to encourage adoption of a Cyc-compatible vocabulary terminology by semantic web developers.

[9] http://www.opencyc.org
[10] http://www.umbel.org/

3.1.4 YAGO-SUMO

YAGO-SUMO[11] (de Melo et al., 2008a) is the merger of two systems, YAGO (yet another great ontology) and SUMO (suggested upper model ontology). SUMO is a formal ontology that contains both general and domain-specific concepts, with the goal of providing useful axiomitization for automated inference. As of spring 2009, the core of SUMO contains roughly 1000 terms, 4000 axioms and 800 rules. Additions include the mid-level ontology (MILO) and a number of domain ontologies. The combined set is approximately 20000 terms, 70000 axioms, and 3000 rules. SUMO focuses on providing an upper ontology, and has limited instance knowledge focusing instead on defining relationships. SUMO has manually linked to all of WordNet using synonymy, subsumption, and instance relations. YAGO merges WordNet with Wikipedia, and provides a detailed mapping between WordNet synsets and Wikipedia. YAGO-SUMO uses both component ontologies mapping to WordNet as the primary interface mechanism. Wikipedia also forms a basis for the DBpedia and FreeBase project, and thus connections can be made from these ontologies into YAGO-SUMO at the instance level. The merged taxonomy is represented in Figure 3.3 below:

[11] http://www.mpi-inf.mpg.de/~gdemelo/yagosumo.html

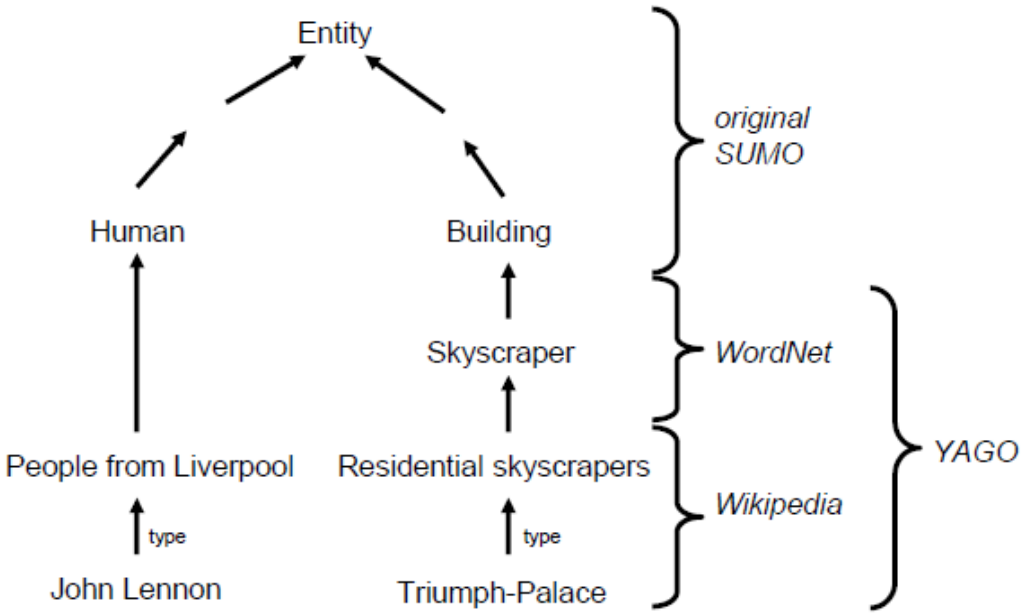

Figure 3.3 The Merged YAGO-SUMO Taxonomy

3.1.5 Microsoft Research Paraphrase Dataset

The Microsoft Paraphrase Corpus[12] (Quirk et al., 2004; Dolan et al., 2004) is a dataset of 5801 pairs of sentences collected over 18 months from multiple web news sources. Each pair has an associated annotation if multiple human annotators considered the pair to be close enough in meaning to be a paraphrase. Each sentence is annotated by two human judges, and disagreements are resolved by a third judge. A separate company carried out the annotation. After conflict resolution, 3900 of the pairs were marked as "semantically equivalent." However, "non-equivalent" is not "unrelated," as information may be added or deleted between the pair such that while related, they are not paraphrases of each other. Examples are provided in Figure 3.4.

[12] http://research.microsoft.com/en-us/downloads/607D14D9-20CD-47E3-85BC-A2F65CD28042/default.aspx

The genome of the fungal pathogen that causes Sudden Oak Death has been sequenced by US scientists.

Researchers announced Thursday they've completed the genetic blueprint of the blight-causing culprit responsible for sudden oak death.

Scientists have figured out the complete genetic code of a virulent pathogen that has killed tens of thousands of California native oaks.

The East Bay-based Joint Genome Institute said Thursday it has unraveled the genetic blueprint for the diseases that cause the sudden death of oak trees.

Figure 3.4 A Set of Example Paraphrases from the Microsoft Paraphrase Corpus

3.1.6 Student Answer Dataset

Mohler and Mihalcea (2009) produced a collection of short student answers and grades for an introductory undergraduate Computer Science course[13]. The data set consisted of 21 questions, each provided with the teacher's answer and a ranked set of 30 student answers. The data set was developed to test automatic short answer grading systems. The goal of short answer grading is to provide a grade given for a one to three sentence answer. Mohler and Mihalcea (2009) provides a comprehensive evaluation of a number of text similarity measures applied to short answer grading, and showed that the best knowledge graph based and corpus-based methods were comparable for the task. Significant improvement was found for latent semantic analysis (LSA) when a medium size domain-specific corpus was extracted from Wikipedia and used. This Student Answer dataset was used in the research. The results reported in (Mohler and Mihalcea, 2009) were used as a baseline for comparison.

[13] http://lit.csci.unt.edu/~rada/downloads/ShortAnswerGrading_v1.0.tar.gz

3.2 Code Bases

3.2.1 Wikify!

To automatically identify the important encyclopedic concepts in an input text, I used the unsupervised system Wikify![14] (Mihalcea and Csomai, 2007), which identifies the concepts in the text that are likely to be highly relevant (i.e., "keywords") for the input document, and links them to Wikipedia concepts. This process, known as *Wikification* (a term used by Wikipedia editors), is that of connecting raw text to relevant articles in Wikipedia.

Wikify! works in three steps, namely: (1) candidate extraction, (2) keyword ranking, and (3) word sense disambiguation., illustrated in Figure 3.5.

[14] http://lit.csci.unt.edu/~wikify/

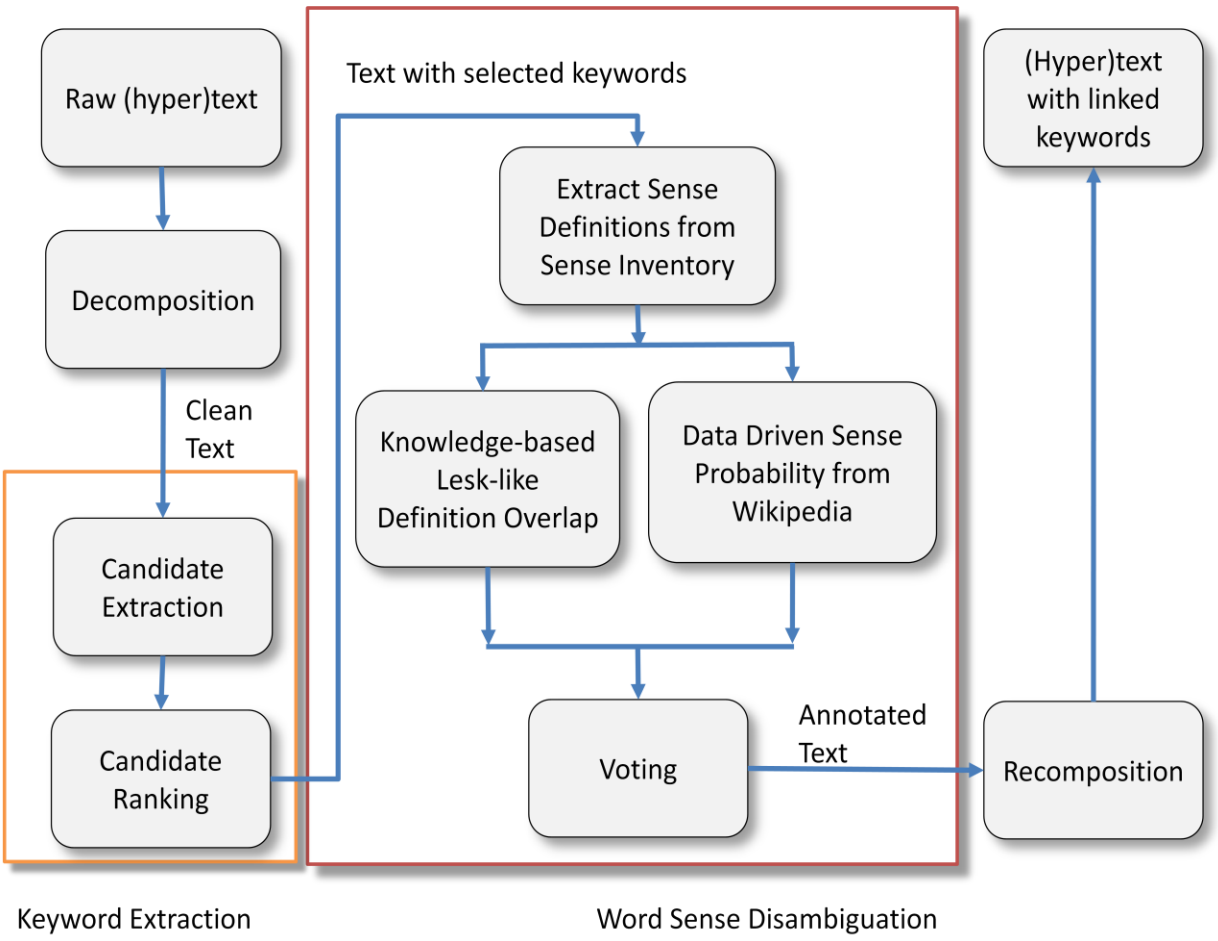

Figure 3.5 The Architecture of Wikify!

The candidate extraction step parses the input document and extracts all the possible n-grams that are also present in the vocabulary used in the encyclopedic graph (i.e., anchor texts for links inside Wikipedia, or article or category titles).

Next, the ranking step assigns a numeric value to each candidate, reflecting the likelihood that a given candidate is a valuable keyword. Wikify! uses a "keyphraseness" measure to estimate the probability of a term W to be selected as a keyword in a document by counting the number of documents where the term was already selected as a keyword. These counts are collected from all the Wikipedia articles.

$$P(keyword|W) \approx \frac{count(D_{key})}{count(D_W)}$$

This probability can be interpreted as "the more often a term was selected as a keyword among its total number of occurrences, the more likely it is that it will be selected again." Although this probability estimate could become unreliable for marginal cases where the counts are very low, only words that appeared at least five times in Wikipedia are considered, which addresses this problem.

Finally, a simple word sense disambiguation (WSD) method is applied, which identifies the most likely article in Wikipedia to which a concept should be linked. This step is trivial for words or phrases that have only one corresponding article in Wikipedia, but it requires an explicit disambiguation step for those words or phrases that have multiple meanings (e.g., "plant") and thus multiple candidate pages to link to. For the non-trivial cases the algorithm uses symbolic methods that attempt to maximize the overlap between the current document and the candidate Wikipedia articles combined with statistical methods that identify the frequency of meanings in text.

The symbolic, knowledge-based approach is inspired by the Lesk algorithm (Lesk, 1986), and uses contexts for WSD. Given an ambiguous term, the system finds its possible meanings (the articles and categories it could possibly link to), and the contexts in which they occur. It then determines the overlap between the current context of the term and the context of the term when having a particular meaning. Thus, given "…*it is danced in ¾ time, with the couple turning 180 degrees every bar*…" the system compares the context for "bar" with the contexts listed for "bar_(music)" and "bar_(counter)."

The statistical method is based on using Naïve Bayes applied to the sense probabilities derived from Wikipedia. For each possibly ambiguous word, a training feature vector is generated from the local and global context of the anchor word, with the classification

being the article pointed to by the link. The local context is taken to be the current word and its part of speech, and the three words to either side of the term and their parts of speech. The global context is specified as at most five sense-specific keywords that co-occur at least three times in the context of the target meaning.

A mapping between a term and an article is assigned if both methods agree, improving the precision of the overall system. During development, disagreement between the statistical and symbolic method occurred in 17% of the cases, indicating an area of uncertainty and possible error. For the application of Wikification, precision has a higher importance than recall.

3.2.2 WEKA

WEKA is a commonly used generic machine learning package developed at the University of Waikato (Witten and Frank, 2004). Written in Java, the system implements several machine learning methods and includes tools for preprocessing, classifier generation and evaluation, regression, clustering and visualization. The system accepts attribute-relation file format (ARFF) formatted files along with a more recently developed XML format for training and testing data input. WEKA provides an excellent cross-section of supervised machine learning methods.

CHAPTER 4

DYNAMIC RANKING OF ENCYCLOPEDIC KNOWLEDGE

> 'Knowledge,' in the sense of information, means the working capital, the indispensable resources, of further inquiry; of finding out, or leaning more things.
>
> *– John Dewey, Democracy and Education*

WikiRank is based on the premise that external encyclopedic knowledge can be used to identify relevant topics for a given document. WikiRank consists of two main steps. In the first step, a knowledge graph of encyclopedic concepts based on Wikipedia is constructed, where the nodes in the graph are represented by the entities and categories that are defined in this encyclopedia. The graph contains 5.8 million nodes, and 65.5 million edges. The edges between the nodes are represented by their relation of proximity inside the Wikipedia articles. The graph is built once and then it is stored offline, so that it can be efficiently used for the identification of topics in new documents.

Figure 4.1 below shows a small section of the knowledge graph as built starting with the article on "Corpus Linguistics."

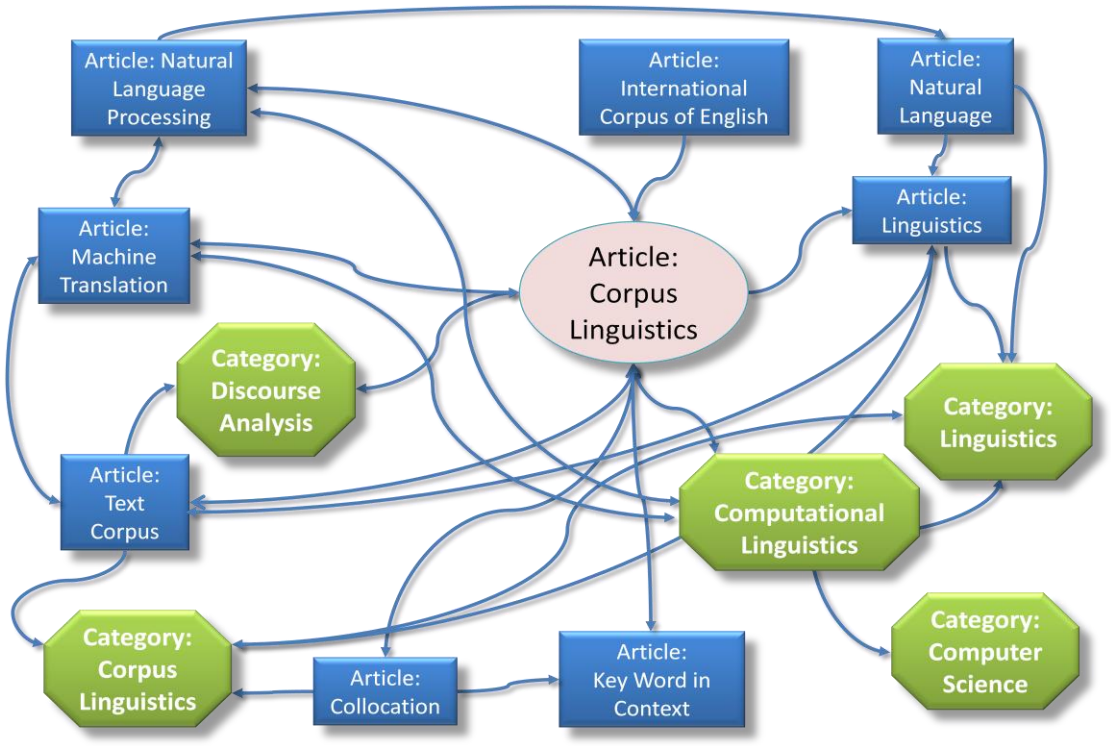

Figure 4.1 Graph centered on "Corpus Linguistics"

In the second step, for each input text, the important encyclopedic concepts in the text are identified, and thereby create links between the content of the text and the external encyclopedic graph.

Next, a biased graph centrality algorithm is run on the entire graph, so that all the nodes in the external knowledge repository are *ranked based on their relevance to the input text*. A variation of the PageRank algorithm is used, which accounts for both the relation between the nodes in the document and the encyclopedic graph, as well as the relation between the nodes in the encyclopedic graph itself.

The goal of the overall process is to find what concepts in the encyclopedia are important to understanding the text, *including those concepts not explicitly mentioned*. In this chapter I will describe the dynamic ranking process for an encyclopedic graph and its possible uses. I will begin by describing the unbiased Markovian random walks and

their relation to the traditional PageRank. Next, I will describe the dynamic biased version, and provide an illustration of the process. Finally, I will discuss the relationship between random walks, the information they can provide regarding knowledge access, and their use.

4.1 Unbiased Markovian Random Walk Simulations and PageRank

Graph-based ranking algorithms such as PageRank are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. One formulation is in terms of a random walk through a directed graph. A "random surfer" visits nodes of the graph, and has some probability of jumping to some other random node of the graph, and the remaining probability of continuing their walk from the current node to one in its list of outgoing connections (or successor list of that node). The rank of a node is an indication of the probability that one would find the surfer at that node at any given time.

Formally, let $G=(V,E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of $V * V$. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it (its predecessors), and let $Out(V_i)$ be the set of vertices that vertex V_i points to (its successors).

The score of a vertex V_i is defined as follows (Brin and Page, 1998):

$$S(V_j) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where d is a damping factor usually set to 0.85. This random jumping factor addresses the problem of cycles and unconnected nodes, since in principle every node has a non-zero probability of being reached.

Given the "random surfer" interpretation of the ranking process, the *(1-d)* portion represents the probability that a surfer will jump to a given node from any other node at random, and the summation portion indicates that the process will enter the node via edges directly connected to it.

Implicit in this description are the background assumptions about linking between web pages. A link from some page V_p to some other page V_q denotes an endorsement by the author of V_p of page V_q being an authority or relevant to some subject usually denoted by the anchor text (if any) provided in the link between the two. Based on this view, the random walk determines the relative authority of pages, independent of knowledge of the content, based on the "votes" of the author's that choose to link to a given page. The idea behind applying PageRank to the web graph is that it can return a query-independent authority value for each page indexed. In this way it is using the "common sense" or "common knowledge" of the web authors to indirectly infer those pages that they use as authorities. The stationary distribution the system finally converges to is a measure of where (after an infinite amount of time) a surfer will eventually visit, using the amount of visitation as a measure of authority. In the application of a search engine, given two pages of equal relevance, the system prefers to list them with the highest PageRank first, and thus hopefully one with higher considered authority.

Of course, when given no other information than just the graph this process is completely unbiased and is simply computing the *a priori* relative authority value for each node. Suppose some nodes in the graph have been identified as being of special interest. How can this information be utilized?

4.2 Biased Ranking of the Wikipedia Graph

Starting with the graph of encyclopedic knowledge, and knowing that some nodes in this graph can be mapped to terms in an input document, one goal is to rank all the nodes in the graph so that one obtains a score that indicates their importance or authority relative to the given input text. This is done by using the same graph-ranking algorithm biased toward the nodes belonging to the input text.

Using a method inspired by earlier work (Haveliwala, 2002), the PageRank formula is modified so that the *(1-d)* component also accounts for the importance of the concepts found in the input text, and it is suppressed for all the nodes that are not found in the input document:

$$S(V_j) = (1-d) * Bias(V_j) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

where $Bias(V_i)$ is only defined for those nodes initially identified in the input document:

$$Bias(V_i) = \frac{f(V_i)}{\sum_{j \in InitialNodeSet} f(V_j)}$$

and 0 for all other nodes in the graph. *InitalNodeSet* is the set of nodes belonging to the input text.

Note that $f(V_i)$ can vary in complexity from a default value of 1 to a complex knowledge-based estimation. In my implementation, I use a combination of the "keyphraseness" score assigned to the node V_i and its distance from the "Fundamental" category in Wikipedia. Other functions can be also used to represent the relative amount of "time or interest" a surfer would give to a particular node in the graph.

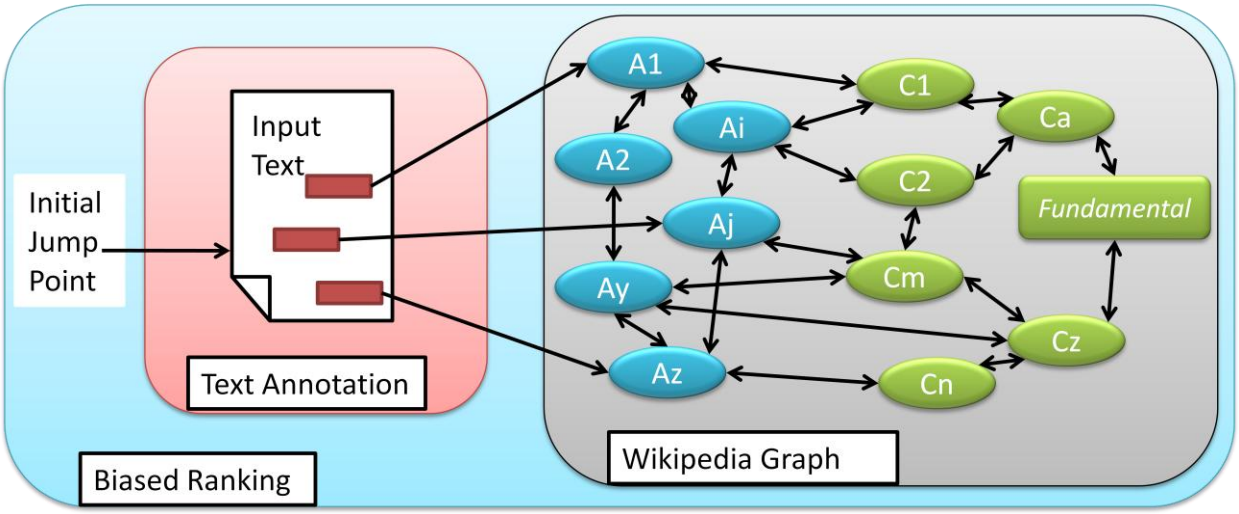

Figure 4.2 Biased Ranking of Wikipedia Graph Using Text

These reinforced nodes are the "implied related topics" of the text. Assume the input text is merged into the overall graph using an annotation process that created the required edges with the pre-existing graph as shown in Figure 4.2. Nodes in the pre-existing graph that receive an edge from the input text have a non-zero *Bias* assigned. The use of *Bias* assigned to each node means the surfer's random jumps will be limited to only those nodes connected to the original query or input. This simulates the effect of a surfer after exhausting a thread of enquiry, returning to their original input text and starting the process again from that point. The graph-ranking process thus becomes biased and focused on those topics directly related to the input. It also accumulates activation at those nodes not directly found in the input text, but linked through indirect means, thus reinforcing the nodes where patterns of activation intersect and creating a constructive interference pattern in the network. Those nodes receiving reinforcement are the "authoritative articles" the surfer would visit repeatedly due to graph topology created by the encyclopedia's authors interacting with the connections found to the input text.

4.3 Illustration of the Process

To illustrate the ranking process, consider as an example the following sentence: "The United States was involved in the Cold War."

First the text is passed through the Wikify! system with a key ratio of 0.5, which returns the articles "United States" and "Cold War." Taking into account their "keyphraseness" as calculated by Wikify!, the selections are given an initial bias of 0.5492 ("United States") and 0.4508 ("Cold War").

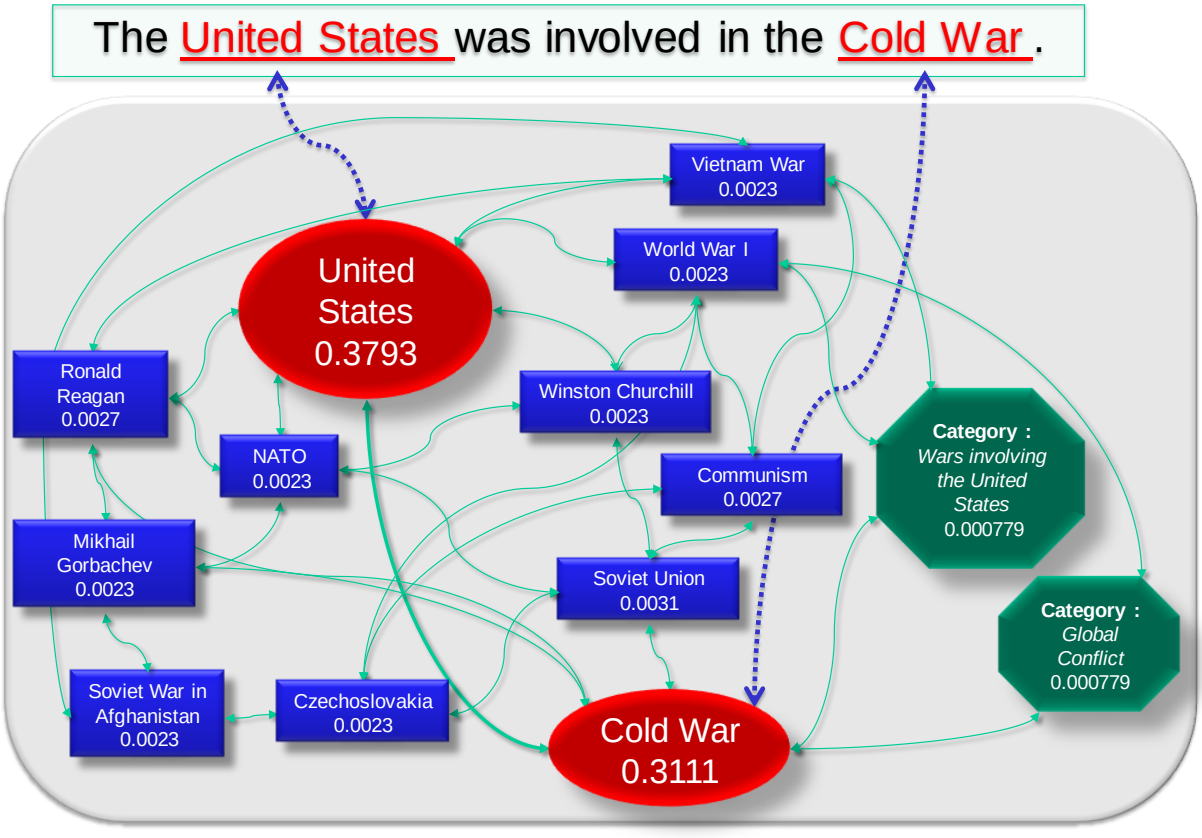

Figure 4.3 Ranking of the Subgraph between "United States" and "Cold War"

After the first iteration the initial activation spreads out into the encyclopedic graph, the nodes find a direct connection to one another, and correspondingly their scores are changed to 0.3786 ("United States") and 0.3107 ("Cold War"). After the second

iteration, new nodes are identified from the encyclopedic graph, a subset of which is shown in Figure 4.3. The process will eventually continue for several iterations until the scores of the nodes do not change. The nodes with the highest scores in the final graph are considered to be the most closely related to the input sentence, and thus selected as relevant additional knowledge.

In order to see the effect of the initial bias, consider as an example the ranking of the nodes in the encyclopedic graph when biased with the sentence "The United States was involved in the Cold War," versus the sentence "Microsoft applies Computer Science." A comparison between the scores of the nodes when activated by each of these sentences is shown in Table 4.1.

Table 4.1 Node Ranking Differences when Encyclopedic Graph is Biased with Different Inputs: (1) "United States" and "Cold War" (US/CW) vs. (2) "Microsoft" and "Computer Science" (MS/CS)

NODE TYPE	WIKIPEDIA ENTRY	US/CW	MS/CS	DIFF
article	United States	0.393636	0.006578	0.387058
category	Computer Science	0.000004	0.003576	-0.003571
article	World War II	0.007102	0.003674	0.003428
article	United Kingdom	0.005346	0.002670	0.002676
category	Microsoft	0.000001	0.001839	-0.001837
category	Cold War	0.001695	0.000006	0.001689
category	Living People	0.000835	0.002223	-0.001387
category	Mathematics	0.000029	0.001337	-0.001307
category	Computing	0.000008	0.001289	-0.001280
category	Computer Pioneers	0.000002	0.001238	-0.001235

4.4 What Can Random Walks for Graph Centrality Tell You?

The initial success of the search engine Google has been attributed in part to the relevancy of the search results it returned at the time, which was due to its use of the PageRank algorithm. The probability distribution over the web pages indicates in part the likelihood that a average surfer would *a priori* be aware of a given page, and when

given two possibly informative results, return the one most likely to be accessed first. By applying the random surfer model to the whole web PageRank was able to provide a soft proxy for "common sense" or "common knowledge." The number of links by other pages to a given page tended to indicate (before search engine optimization became popular) how well known in general a given page was. If many website owners expended the time to make a link to a page then that page first had to be known by those people, and thus the number of linking sites approximates how well known a particular page is. Also implicit is the assumption that if many people are linking to a page, then the source page's authors thought that the target page would contain something that a reader of their page would find "interesting" (for any number of reasons). In other words, the ranking given a page is also a measure of the page satisfying some information need of the visitors and thus worthy of their time or attention.

A text can be parsed and converted into a graph, with the words or phrases, or sentences represented by nodes and the linguistic connections detected by the parser forming the edges. This is the idea behind TextRank (Mihalcea et al., 2004) which has been used to perform keyword extraction and extractive summarization. Running the random walk over the linguistic graph returns a visitation or access frequency for each node, and by proxy the textual element it represents. At the word level this provides a ranking for potential keywords, while at the sentence level it identifies the central sentences of the text. In both cases it estimates the amount of time a reader that constantly follows the connections would encounter each element.

When an unbiased random walk is run over the Wikipedia graph, the final distribution is a measure how often a surfer would read a given article (and by assumption become familiar with its content). This static distribution would be an estimate of the surfers'

eventual familiarity with each article in the entire network, given an infinite amount of time.

When the random walk over the Wikipedia graph is applied in a *biased* manner, the values returned are skewed to represent the surfer's *awareness distribution* if they constantly restarted when bored at the nodes of the initial biased set. The biased set of nodes is derived from the textual input, and creates a virtual start point node linked into the graph. Using this virtual start point results in the distribution that represents the eventual "article awareness level" of a surfer focused (or obsessed) with the input document.

Just as the unbiased random walk provides background "article awareness" and uses it as a proxy for estimating commonness of knowledge (or familiarity), the biased random walk provides an estimate of "focused article awareness"—in a sense, estimating what articles and pages the user would *eventually* read and become aware of in the future given enough time.

The central question WikiRank answers when applied to an encyclopedic graph is: Given a certain input, which articles (and the knowledge they contain) will the user eventually come to access the most?

Answering this question is useful in a number of ways. First, it can provide a ranking determining which articles (and knowledge they contain) should be most relevant to a given input set. It can be used to provide articles in order so a user with limited time can spend it reading in a way to best approximate one with infinite time. The activation level provides information on how related a given encyclopedia entry is, and this in itself is information usable by other algorithms ("is it more about politics or sports?" or "which method is more relevant, neural networks or decision trees?"). When ranked, the articles and categories can provide a form of metadata potentially useful for

indexing the original object. One can also compare two input texts in terms of the differences in article awareness each prompts, which would be useful as a whole text similarity measure. Additionally, one can also compare the activation pattern caused by the current input relative to the *a priori* distribution generated by an unbiased ranking, with those articles with a higher deviation being viewed as contextually more informative. Given a text that is continuously updated (such as a news feed), one can notice changes over time, and provide an alert to significant novel changes. The values assigned to entries can be used as input to supervised machine learning algorithms as in (Coursey, 2007) or as a heuristic to problem solving algorithms. And finally, when external information and knowledge sources are associated with the article, the value provides an estimation of the relevancy of those additional materials.

This description of WikiRank as a way to estimate eventual article awareness (and by proxy eventual knowledge) also highlights areas for improvement in that estimation. The better the *Bias* function approximates the relative value of true interests, the better the estimation of eventual awareness. While this value is derived for each query, it could also include information based on each user. The Wikipedia style guidelines limit the number of links to a number that may be less than what a typical interested human reader would know and use. That is, instead of giving up and restarting, a human might issue a search query to jump to a new set of pages. However, this process can be emulated through adding those initial jump nodes to the bias set (making them part of the relevant jump to set), or one could add additional links to the graph. If the surfer is looking for something in particular, then different links should have different weights. Also, one could look outside of Wikipedia to the greater web to see which Wikipedia article's *external* sites point to in an effort to provide an estimate of *a priori* awareness. If a page's access frequency exceeds a threshold then additional knowledge from that

page could be used, possibly by Wikifying the text it contains to obtain more links or Wikifying text of external pages that it points to and adding them to the bias set as well.

Given both the article network and the text of the articles, one could combine both to create an integrated network, with nodes consisting of words, phrases, sentences and articles, and the ranking would return values on all levels. If each individual article is modeled as its own TextRank graph, then the links between articles can be modeled in context (i.e., between the anchor phrase in the source text graph connected to the article title node in the destination text graph). Implementing and testing an integrated TextRank/WikiRank system at this level, while interesting, is outside the immediate scope of the current system.

Finally, the transition probability used at each node is uniform. This follows the basic assumptions of the maximum entropy (MaxEnt) class of algorithms and models (Ratnaparkhi, 1996; Ratnaparkhi, 1998). The core idea used in MaxEnt is that the models that are most uniform while satisfying any known constraints are preferred. Thus, given a node with four possible outgoing links, each destination node would receive 25% of the value. However, if some condition is known that causes one link to be preferred 40% of the time, then (with no other information) the rest would each receive 20%. Currently PageRank weighting of the transitions is the same as MaxEnt with no constraints. MaxEnt could provide one way for weighting while meeting the constraints provided by any additional information.

The analysis of the visitation model and the modifications proposed offer ways to improve the estimation process. Broadly this can be done through including more task specific knowledge (primarily encoded in the bias function or in the future, link weights) or improving the accuracy of the elements being modeled. As will be shown in the remaining chapters, even without these additions the estimation of article

awareness via the soft proxy of estimated visitation frequency proves useful in a number of tasks.

CHAPTER 5

EXPERIMENTS ON TOPIC IDENTIFICATION

> There is no branch of detective science which is so important
> and so neglected as the art of tracing footsteps.
> *– Sir Arthur Conan Doyle, 1891*

Beyond the task of keyword extraction (Mihalcea and Csomai, 2007) lies the task of topic identification. The primary difference is relevant topics may not be directly mentioned in a document, but instead have to be obtained from background knowledge in some field or general global knowledge. Topic identification is different from text classification as described in (Gabrilovich and Markovitch, 2006) in that either the topics are unknown in advance or are provided in the form of a controlled vocabulary with thousands of entries. In topic identification the goal is to find topics or categories that are relevant to a given document, and thus potentially enrich the document by linking it to relevant external knowledge.

In order to measure the effectiveness of the dynamic ranking process, three sets of experiments[15] were performed for the task of topic identification, measuring the relevancy of automatically identified topics with respect to manually annotated gold standard data sets.

In the first experiment, the identification of the important concepts in the input text (used to bias the topic ranking process) was performed manually by the Wikipedia users. In the second and third experiment, the identification of these important concepts is done automatically by the Wikify! system. In all the experiments, the

[15] The work reported in (Coursey and Mihalcea, 2009a) and (Coursey et al, 2009) was reproduced here with permission from the Association for Computational Linguistics.

ranking of the concepts from the encyclopedic graph was done automatically by using the dynamic ranking process described in Biased Ranking of the Wikipedia Graph (in Chapter 4).

In the first two experiments, presented in (Coursey and Mihalcea, 2009a), a data set consisting of 150 articles from Wikipedia are randomly selected following the constraint that they each contain at least three article links and at least three category links; these articles were explicitly removed from the encyclopedic graph. All the articles in this data set include manual annotations of the relevant categories, as assigned by the Wikipedia users, against which the quality of the automatic topic assignments is measured.

The task was to rediscover the relevant categories for each page. Note that this was non-trivial, since there are more than 385,000 categories to choose from. The quality of the system was evaluated through the standard measures of precision and recall.

5.1 Effect of Ranking on Manual Annotation of the Input Text

In the first experiment, the articles in the gold standard data set also include manual annotations of the important concepts in the text, i.e., the links to other Wikipedia articles as created by the Wikipedia users. Thus, only the accuracy of the dynamic topic ranking process (without interference from the Wikify! system) was measured.

Two main parameters could be set during a system run. First, the set of initial nodes used as *Bias* in the ranking can include: (1) the initial set of articles linked to by the

original document (via the Wikipedia links); (2) the categories listed in the articles linked to by the original document[16] (each getting a weight of N^{-1}); and (3) both.

Second, the dynamic ranking process could be run through propagation on an encyclopedic graph that includes: (1) all the articles from Wikipedia; (2) all the categories from Wikipedia; or (3) both articles and categories from Wikipedia.

Figure 5.1 and Figure 5.2 show the recall and precision obtained for the various settings. In the plots, *Bias* and *Propagate* indicate the selections made for the two parameters, which can be each set to Articles, Categories, or Both.

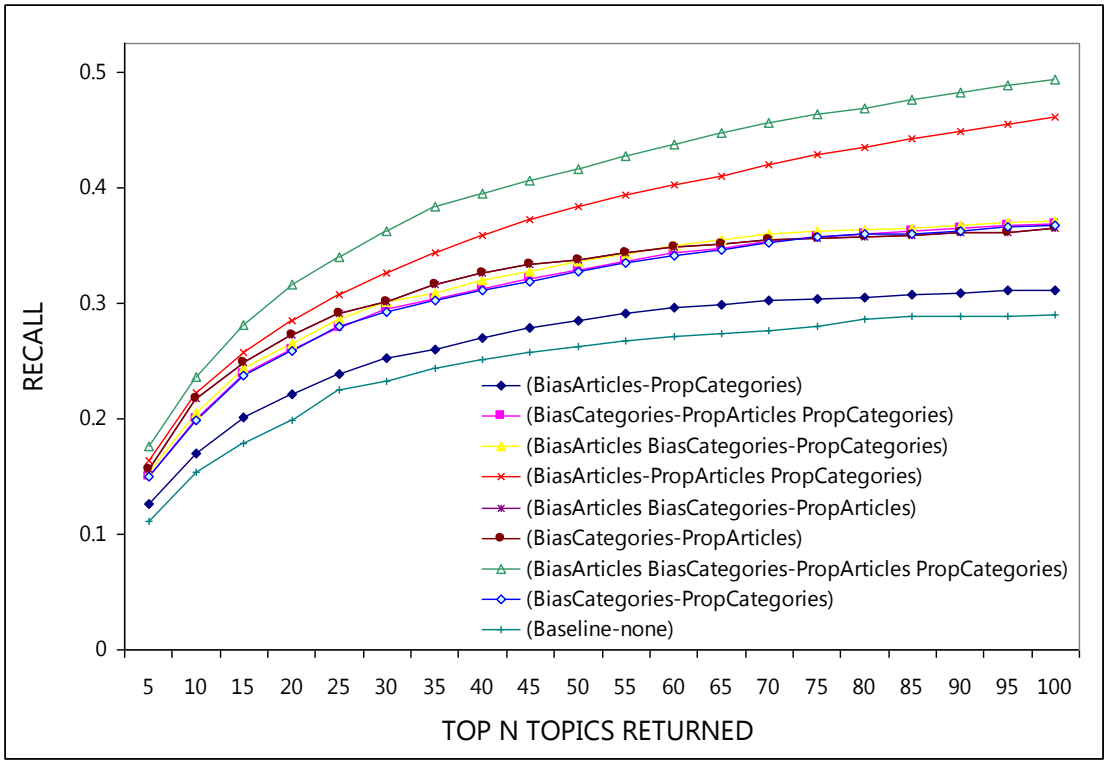

Figure 5.1 Recall Based on Ranking of Manual Annotations

[16] These should not be confused with the categories included in the document itself, which represent the gold standard annotations and are not used at any point.

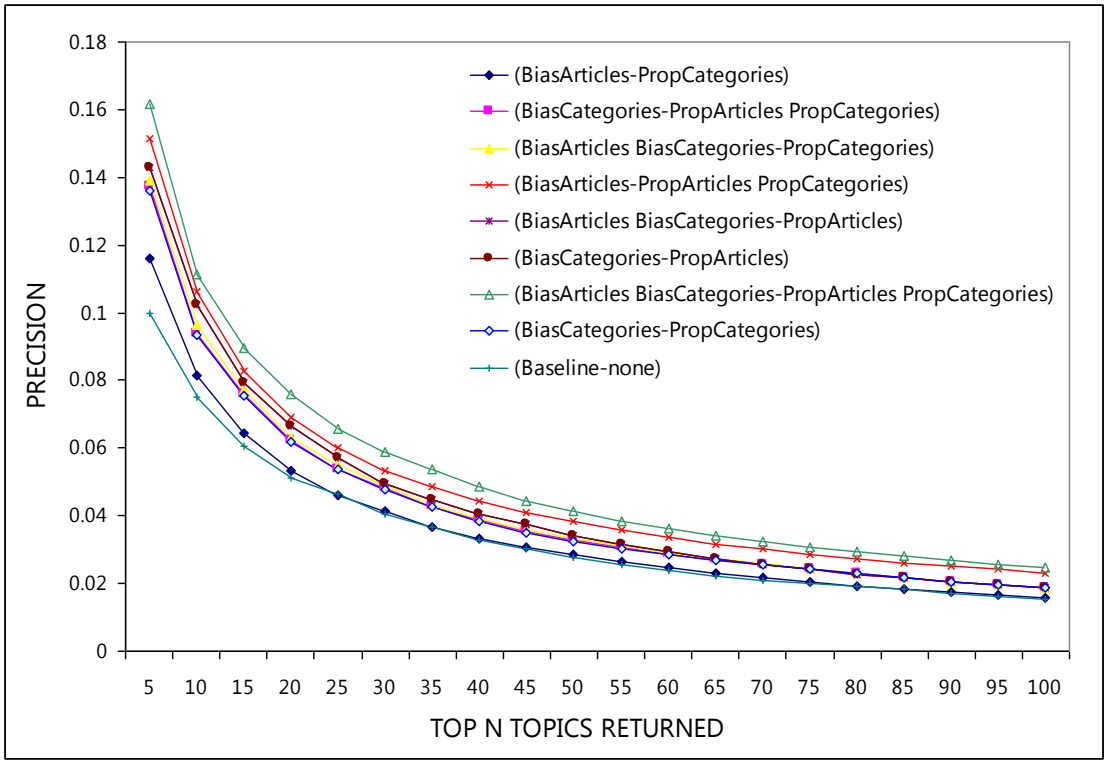

Figure 5.2 Precision Based on Ranking of Manual Annotations

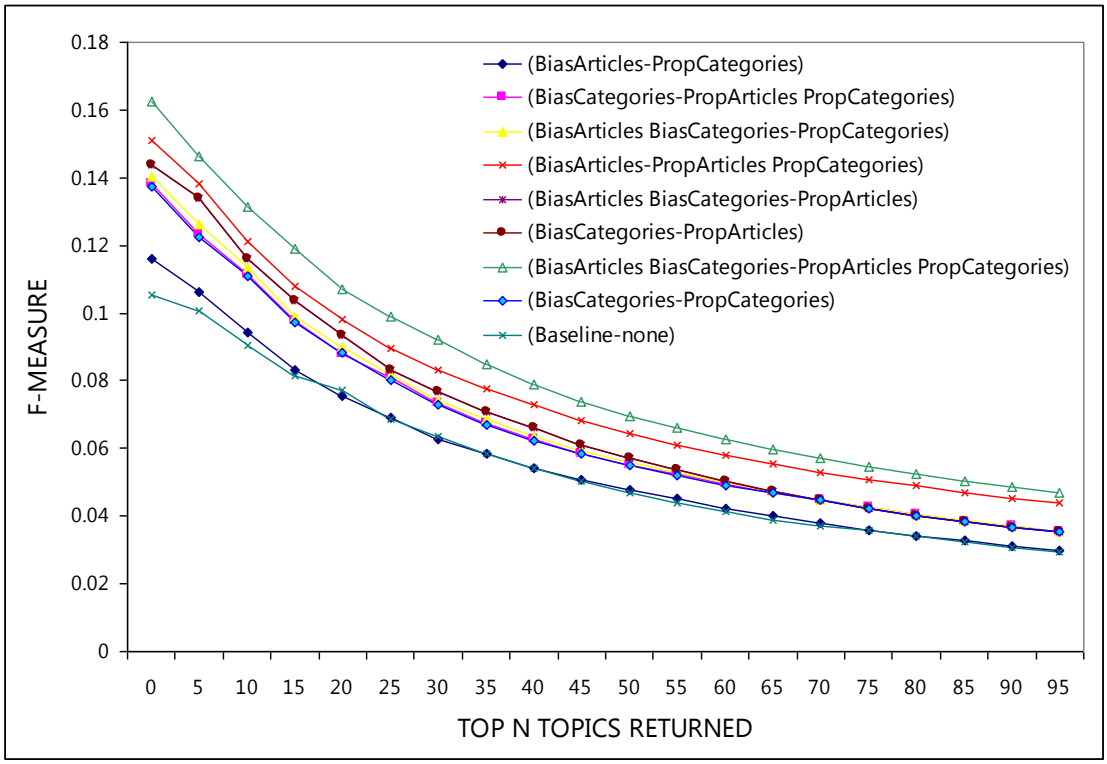

Figure 5.3 F-Measure Based on Ranking of Manual Annotations

As seen in the figures above, the best results were obtained for a setting where both the initial bias and the propagation include all the available nodes, i.e., both articles and categories. Although the primary task was the identification of the categories, the addition of the article links improves the system performance.

To place results in perspective, compare the precision, recall and F-measure of approximately 0.16 as obtained for the top five topics returned by the system, with the random baseline of 5 x 1/350,000 if the topics were randomly chosen from the entire set of more than 350,000 categories available in Wikipedia. Additionally, a baseline was calculated (labeled as "Baseline" in the plots), which selects by default all the categories listed in the articles linked to by the original document.

5.2 Automatic Annotation of the Input Text

The second experiment was similar to the first one, except that rather than using the manual annotations of the important concepts in the input document, the Wikify! system automatically identified these important concepts by using the method briefly described in Wikify! (in Chapter 3) and illustrated in Figure 5.4 below:

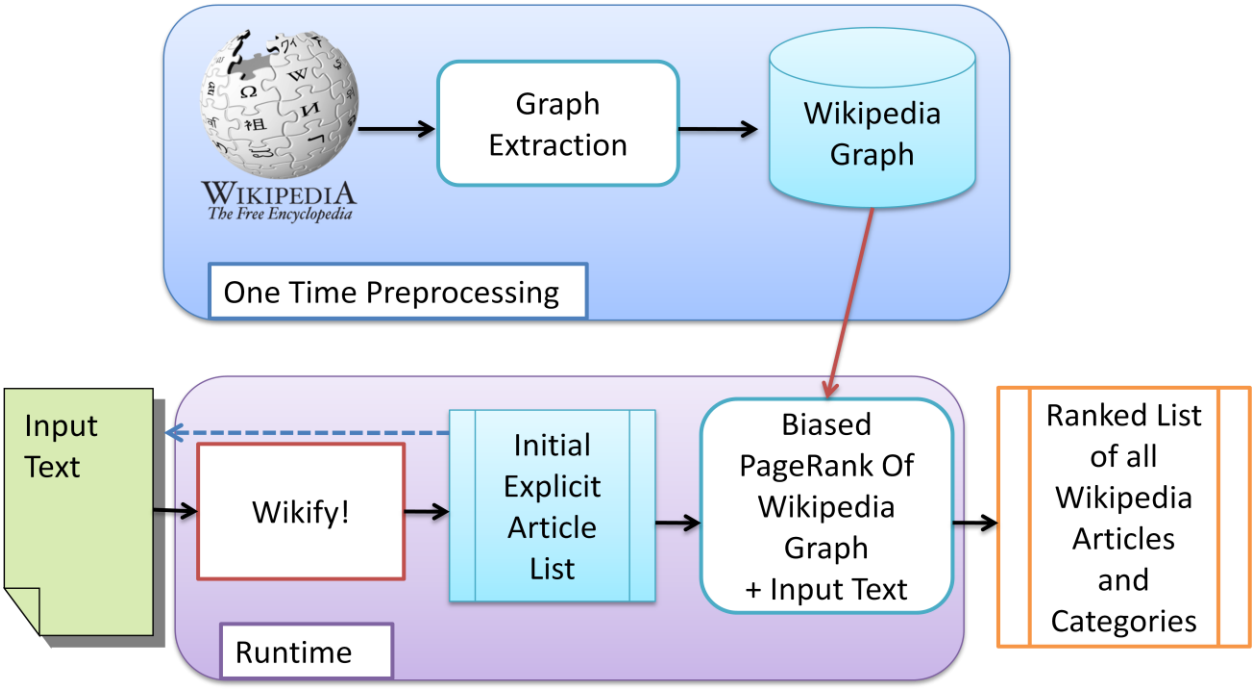

Figure 5.4 Process Flow of Biased Ranking Using Wikification

The article links identified by Wikify! were treated in the same way as the human anchor annotations from the previous experiment. In this experiment an additional parameter was adjusted, which consists of the percentage of links selected by Wikify! out of the total number of words in the document. This parameter is called the keyRatio. The higher the keyRatio, the more terms are added, but also the higher the potential of noise due to mis-disambiguation.

Figure 5.5, Figure 5.6 and Figure 5.7 show the effect of varying the value of the keyRatio parameter on the precision, recall and F-measure of the system. Note that in this experiment, only the best setting for the other two parameters as identified in the previous experiment is used, namely an initial bias and a propagation step that includes all available nodes, i.e., both articles and categories.

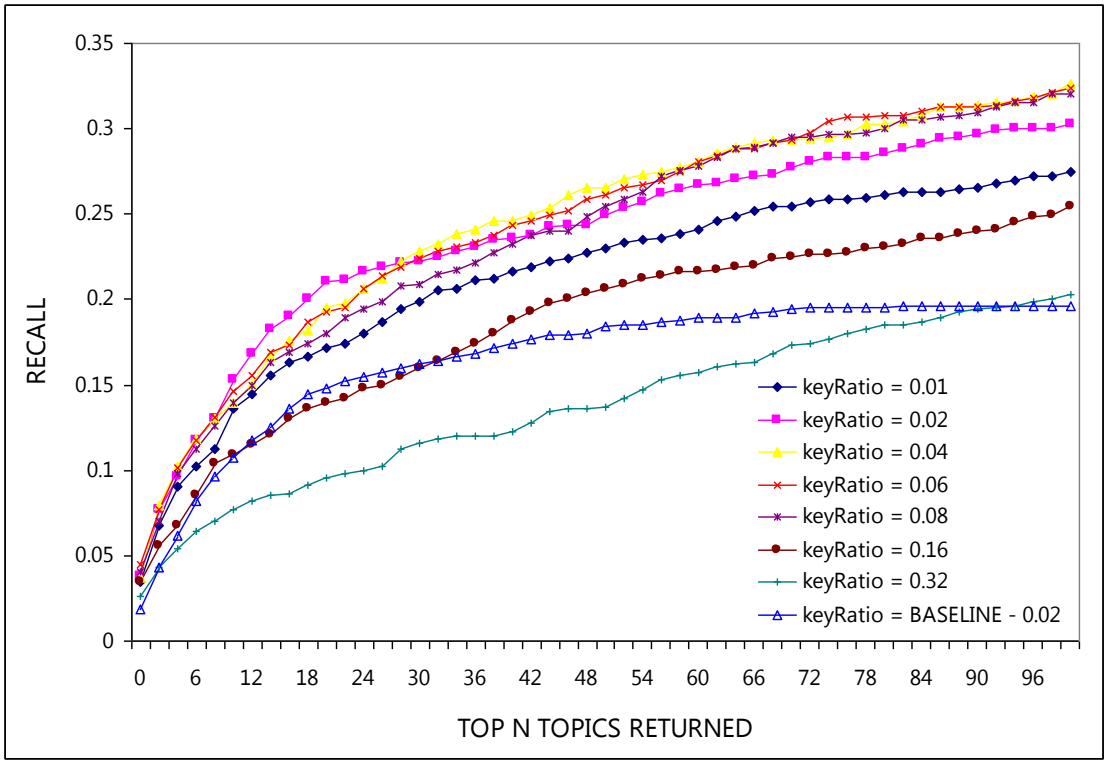

Figure 5.5 Recall Based on Ranking of Wikify! Annotations

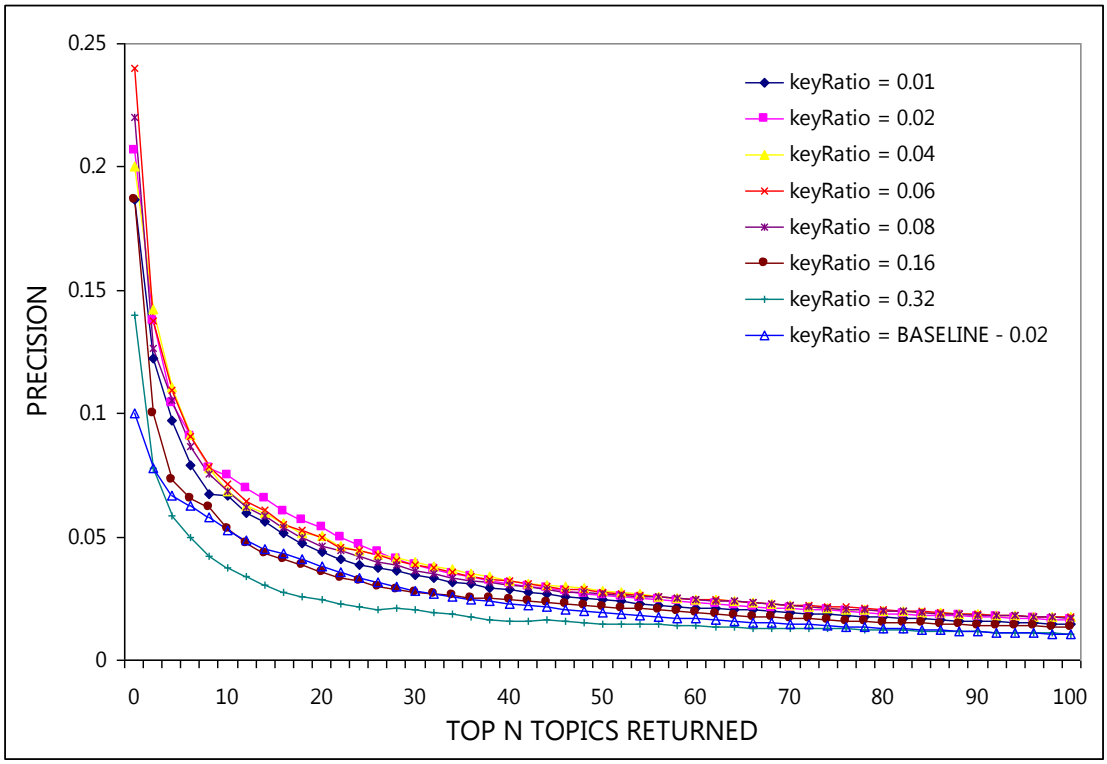

Figure 5.6 Precision Based on Ranking of Wikify! Annotations

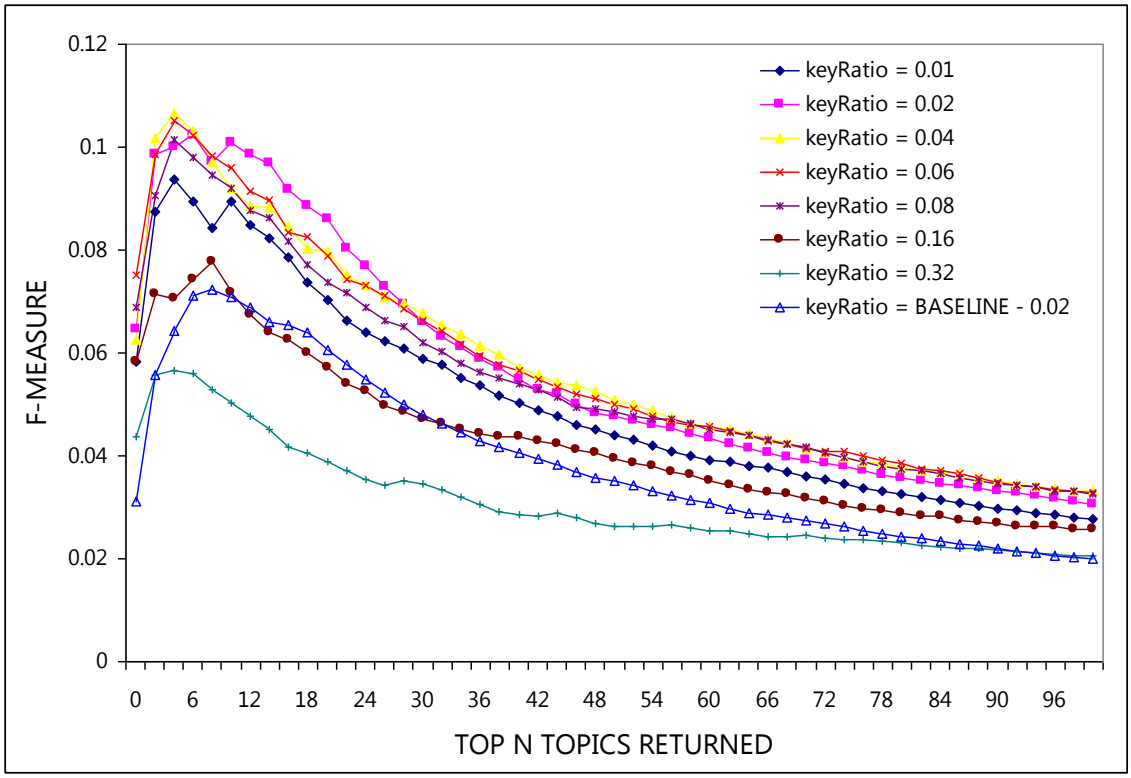

Figure 5.7 F-measure Based on Wikify! Annotations

The system's best performance occurs for a keyRatio of 0.04 to 0.06, which coincides with the ratio found to be optimal in previous experiments using the Wikify! system (Mihalcea and Csomai, 2007).

As before, a baseline was calculated, which selects by default all the categories listed in the articles linked to by the original document, with the links being automatically identified with the Wikify! system. The baseline is calculated for a keyRatio of 0.04.

Overall, the system manages to find many relevant topics for the documents in the evaluation data set, despite the large number of candidate topics (more than 385,000). The system exceeds the baseline by a significant amount, demonstrating the usefulness of using the biased ranking on the encyclopedic graph.

5.3 Article Selection for Computer Science Texts

In the third experiment, also presented in (Coursey et al., 2009), the Wikify! system was used again to annotate the input documents, but this time the evaluations were run on a data set consisting of computer science documents. The data set used was introduced in previous work on topic identification by (Medelyan and Witten, 2008a) (also see the Waikato Topic Indexing Experiments section in Chapter 8) where 20 documents in the field of computer science were independently annotated by 15 teams of two computer science undergraduates. The teams were asked to read the texts and assign to each of them the title of the five Wikipedia articles they thought were the most relevant and that they thought the other groups would also select. Thus, the consistency of the annotations is an important measure for this data set. Medelyan and Witten (2008a) define consistency as a measure of agreement:

$$Consistency = \frac{2C}{A + B}$$

where A and B are the number of terms assigned by two indexing teams, and C is the number of terms they have in common. In the annotations experiments reported in (Medelyan and Witten, 2008), the human teams consistency ranged from 21.4% to 37.1%, with 30.5% being the average[17].

[17] The consistency for one team is measured as the average of the consistencies with the remaining 14 teams.

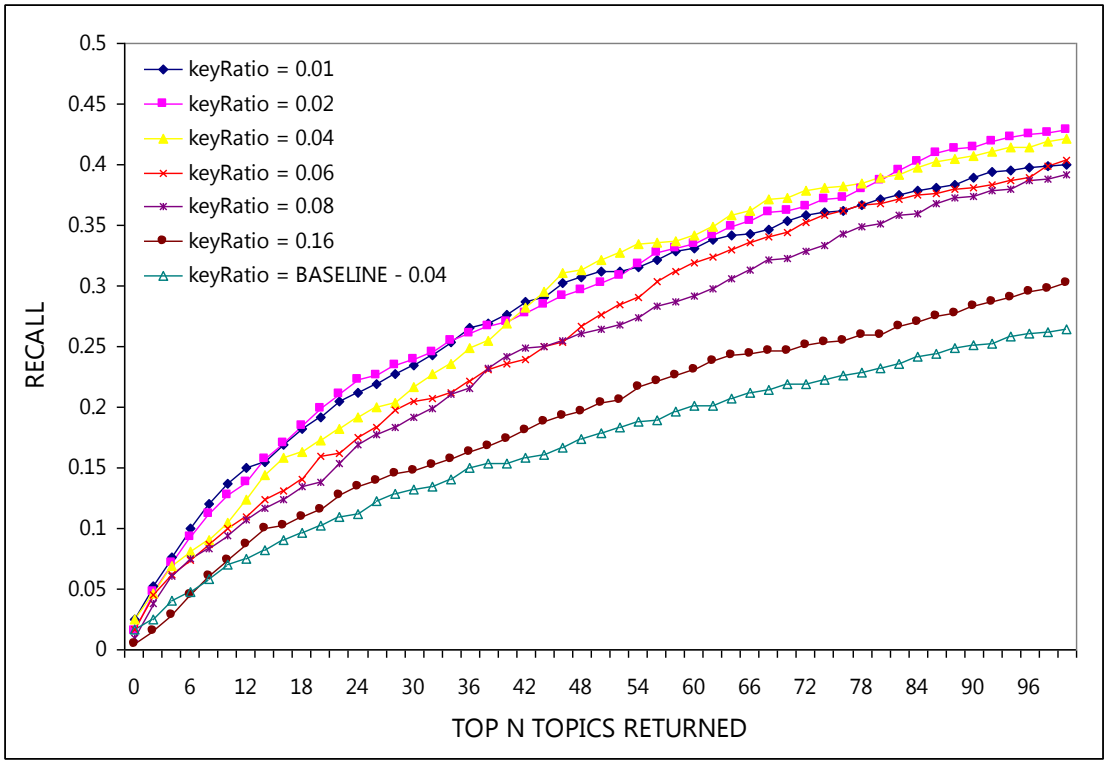

Figure 5.8 Recall for Automatic Annotation of Waikato Dataset

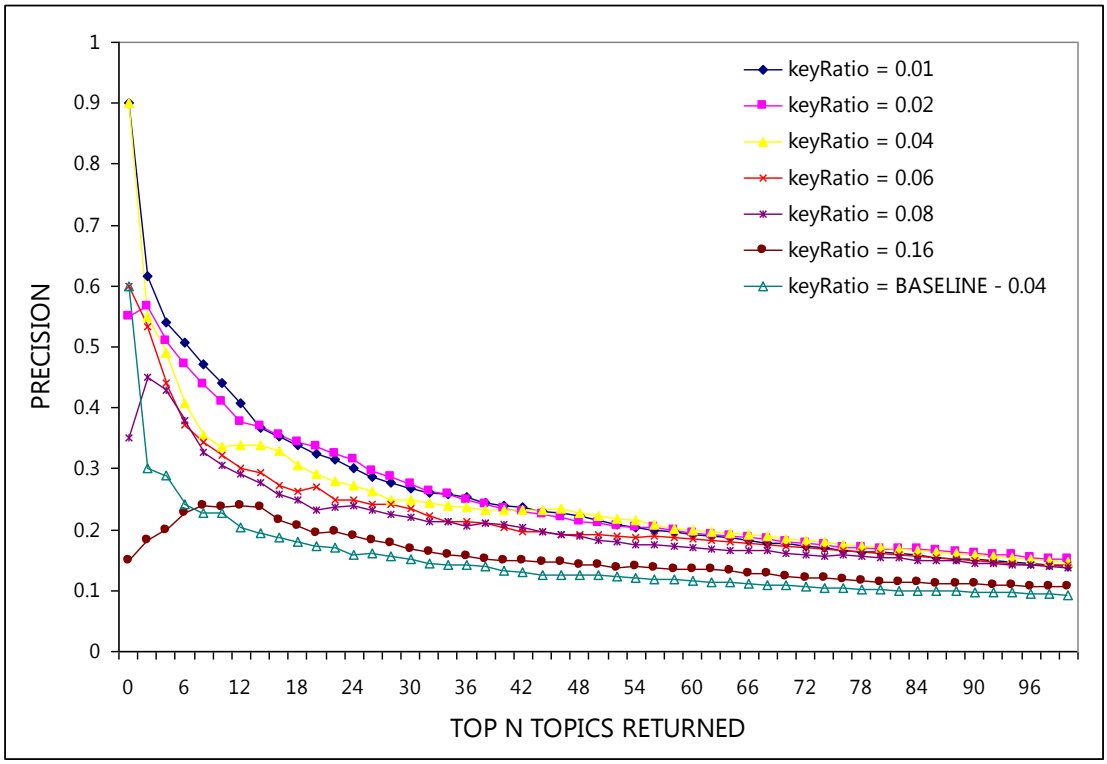

Figure 5.9 Precision for Automatic Annotation of Waikato Dataset

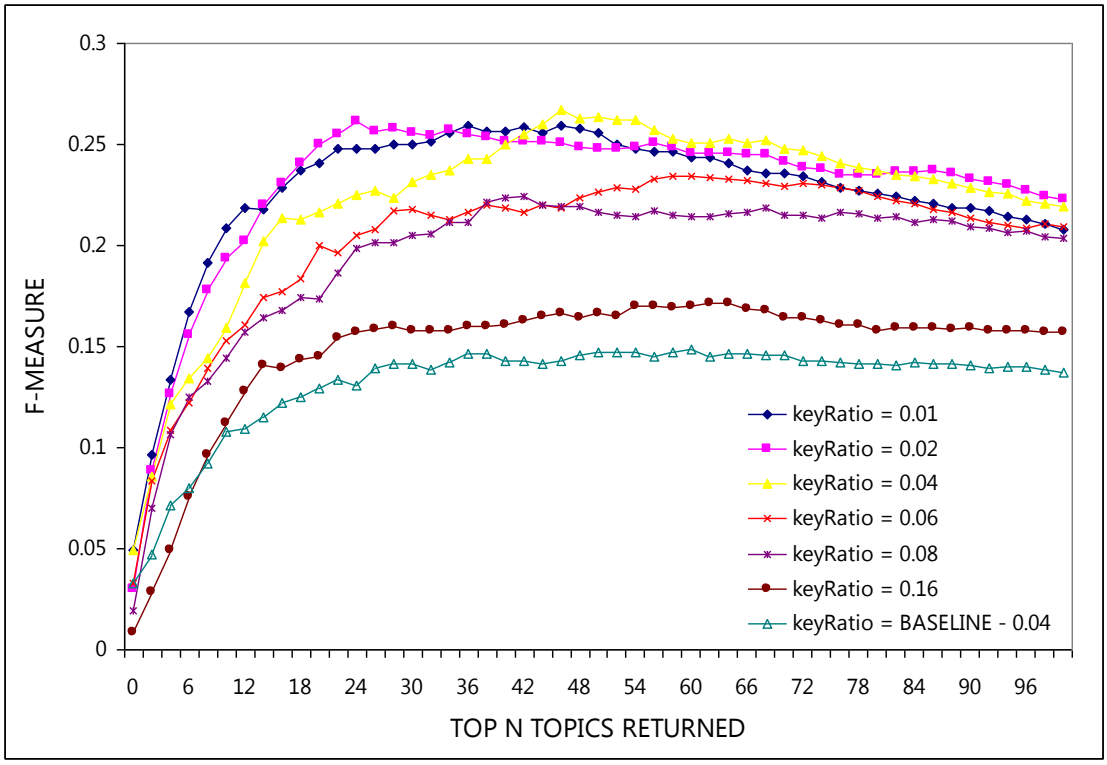

Figure 5.10 F-measure for Automatic Annotation of Waikato Dataset

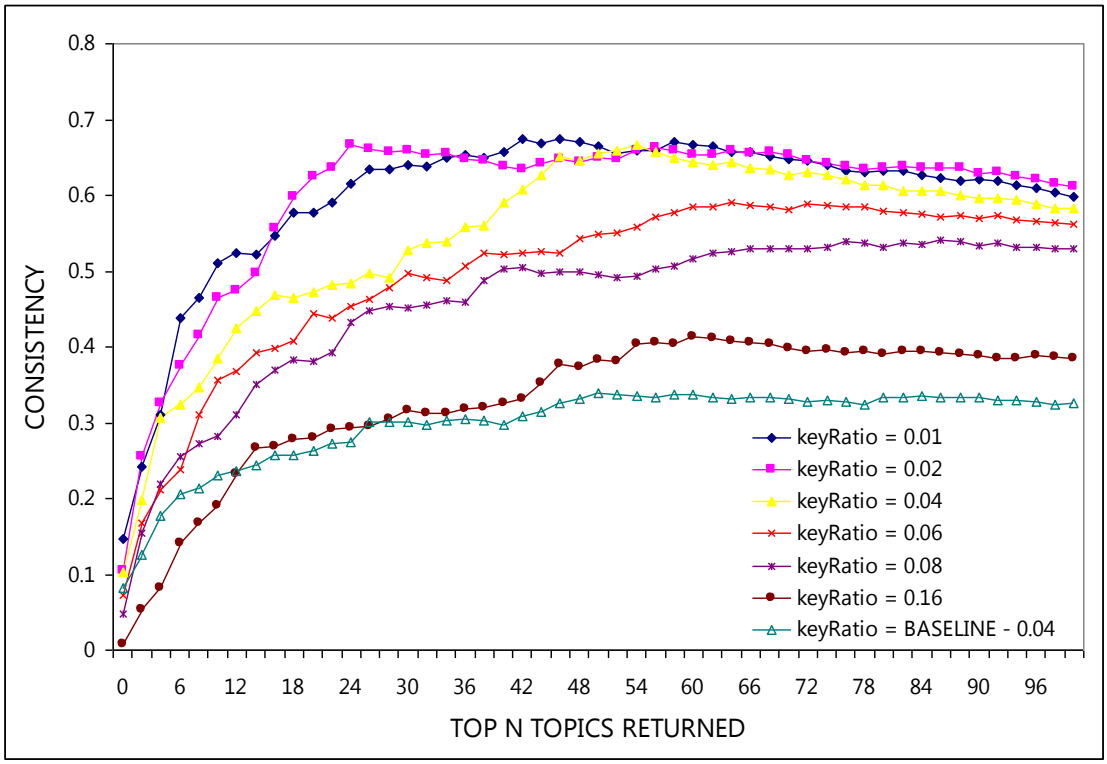

Figure 5.11 Consistency for Automatic Annotation of Waikato Dataset

Figure 5.8, Figure 5.9, Figure 5.10 and Figure 5.11 show the performance of the system on this data set, by using the Wikify! annotations for the initial bias, and then propagating to both articles and categories. The plots also show a baseline that selects all the articles automatically identified in the original document by using the Wikify! system with a key ratio set to 0.04.

When selecting the top five topics returned by the system (the same number of topics as provided by the human teams), the average consistency with respect to the 15 human teams was measured at 34.5%, placing it between the 86% and 93% percentile of the human participants, with only two human teams doing better.

Compare this result with the one reported in previous work for the same data set. Using a machine learning system, (Medelyan et al., 2008a) reported a consistency of 30.5%. The result of 34.5% is significantly better, despite the fact that this method is unsupervised.

In a second evaluation, the union of all the terms assigned by the 15 teams was also considered. On average, each document was assigned 35.5 different terms by the human teams. If allowed to provide more annotations, the system peaks with a consistency of 66.6% for the top 25 topics returned.

In general these results indicate that WikiRank can be used an unsupervised algorithm for topic identification when used with Wikipedia. Despite the large number of possibilities for the evaluation data set WikiRank is able to identify many relevant topics. This leads to the question of "how can this generally useful capability be extended to other areas?"

CHAPTER 6

ESTIMATING TEXTUAL SIMILARITY

> The question 'What makes things seem alike or seem different?' is one so fundamental to psychology that very few psychologists have been naïve enough to ask it.
>
> – *Fred Attneave, Dimensions of Similarity, American Journal of Psychology 63:516-556*

6.1 WikiRank and Text Similarity

A natural question to ask about two objects is if they are similar and if so, in what ways and by how much. When it comes to text, this can be done based on surface features like the co-occurrence of letters and words in text, or it can be based on more abstract features associated with the objects in question. The development of dependable semantic similarity metrics for passages of text would aid the relevancy evaluation function of many natural language processing (NLP) tasks such as summarization, entailment, and information retrieval.

Often such comparisons are made at the lexical level using lexical overlap and vector space methods. Instead of focusing on the space created by Bags-of-Words which can be ambiguous, I examine the effect of comparing Bags-of-Concepts as provided by WikiRank when analyzing a text. WikiRank provides a way to generate associational descriptions in terms of the relative frequency of entries accessed directly or indirectly and thus allows utilization of general knowledge not explicitly mentioned by the text nor in the words definitions.

How can this and other information be used to quantify similarity? In this section I will describe some metrics used to make comparisons on both the semantic and textual level.

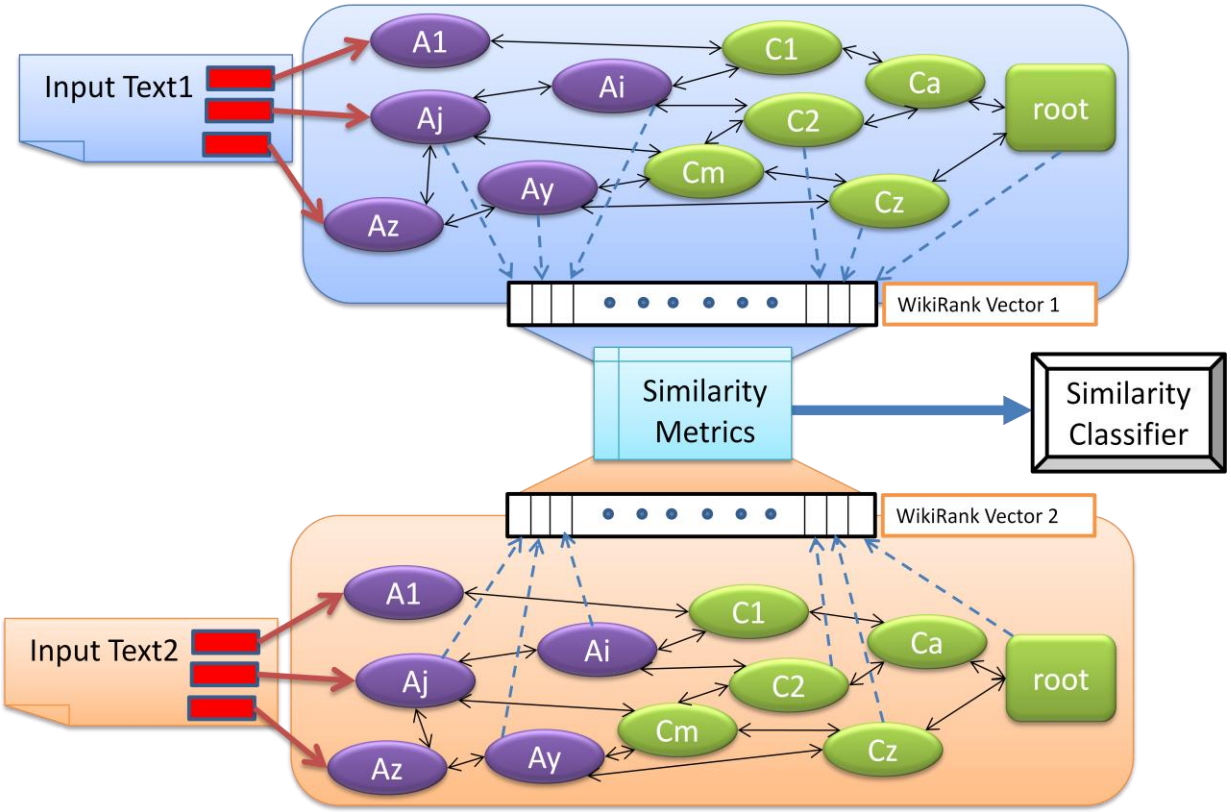

Figure 6.1 Basic Framework for using WikiRank for Textual Comparisons

Figure 6.1 illustrates how WikiRank can be adapted to compute the relatedness of two texts. First, the links between the input texts and the encyclopedic graph are found and the bias function values are defined for each node. In the current set of experiments this is done with Wikify! as described in the previous chapter. Biased PageRank is then run until convergence for each, and the values of each node in the graph are used to define a dimension in a vector representing the visitation frequency that would be generated by each text. The vectors representing each text are then compared using the vector and distribution similarity metrics described in the next section. These similarity metrics

can then be optionally fed as input to a classifier trained to recognize appropriate similarity for a given domain. The use of the similarity metrics serves two functions: first is dimensionality reduction, providing (hopefully) a more relevant feature set for machine learning, and second is to provide generality. While the whole WikiRank vector can be used, any classifier generated would probably be overly specific relative to the standard test sets. Using similarity metrics provide features geared toward whole document similarity versus specific concept comparisons.

For the Figures and Tables below, the following sentences were used:

S1: Heather, 35, who lost a leg in a road accident, is thought to have steel plates fitted in her hips, which would make natural childbirth impossible.

S2: Former model Lady McCartney lost a leg in a road accident in 1993 and is understood to have steel plates fitted in her hips which would make natural childbirth difficult.

S3: Model/activist Heather Mills braved her way through Dancing with a broken pelvis caused by a motorcycle accident in 1993 which resulted also in her leg amputation.

Figure 6.2 and Figure 6.3 below show the partial result of processing two sentences from a paraphrase corpus with WikiRank with Figure 6.4 for comparison. Wikify!, lacking additional context, associates "Heather" with the plant in S1. Both the S1 and S2 rankings give the bulk of the weight to the topic of childbirth, with some the interest in car safety and steel. In the case of S3, the system recognizes Heather Mills, and focuses on the amputation and pelvis and associated anatomy, along with her connection to activism, all links included in the article on her. One key thing to remember is that activation spreads out to hundreds of nodes not shown, and they make up part of any comparison. Table 6.4 shows the results of computing the similarity metrics for each pair of graphs. The mathematics of the metrics are covered later in this chapter. As can be seen, most metrics indicate that the pair S1 and S2 are the most similar. In the case of a supervised learning system, the values of these metrics

would form the features used to learn and perform classification. Each metric is derived from different assumptions of what constitutes similarity, and it is left to the machine learning system to determine which is most useful for the task given it.

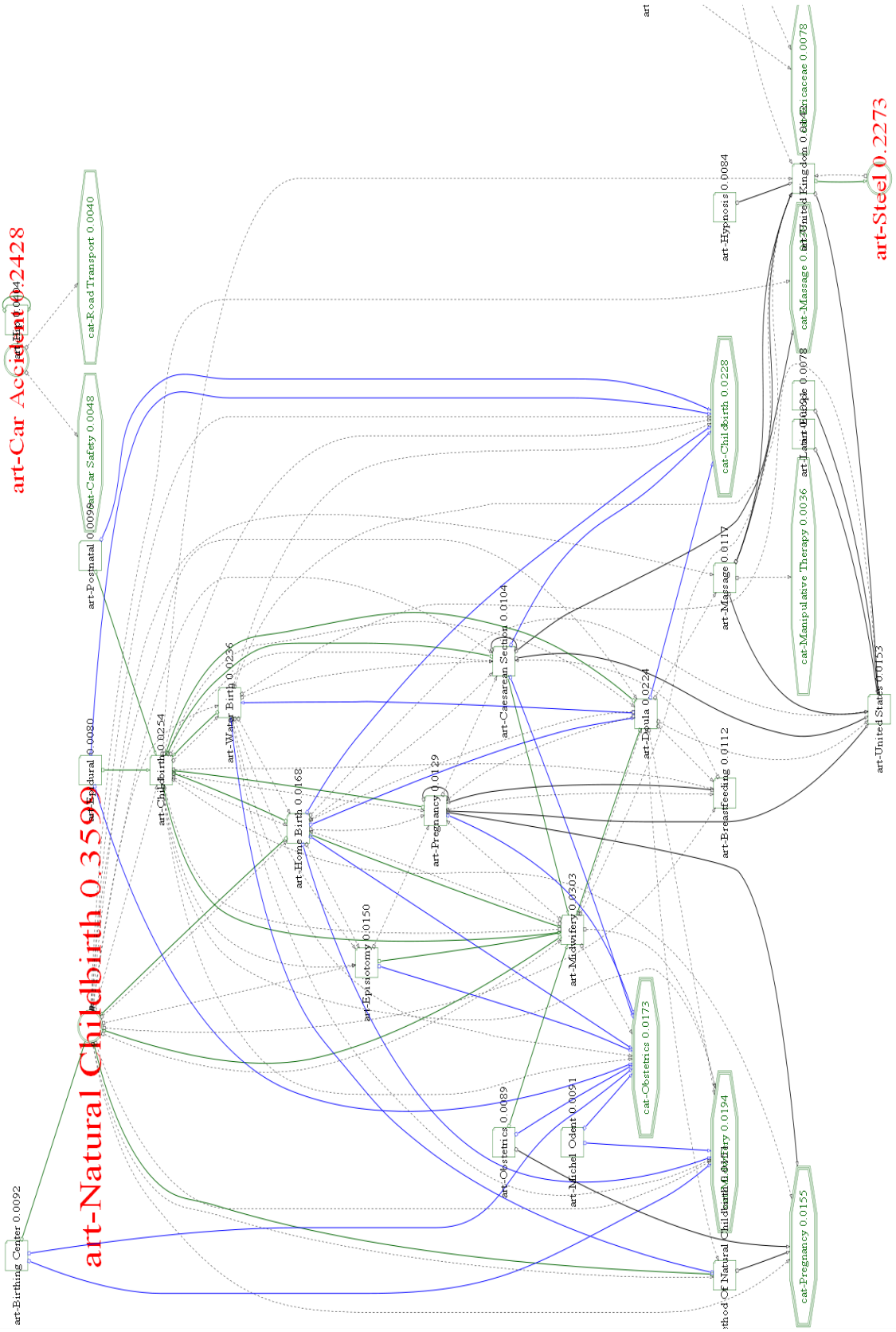

Figure 6.2 Ranking and Linking Caused by Processing S1 with keyRatio=0.2

Table 6.1 The Highest Ranked Elements for Sentence S1 in Figure 6.2

WIKIFY EXTRACTIONS	RANKED WIKIPEDIA ARTICLES	RANKED WIKIPEDIA CATEGORIES
Heather (1.8563065479343)	Natural Childbirth (0.359935)	Childbirth (0.022790)
Car_accident (3.0834056428651)	Car Accident (0.242788)	Midwifery (0.019388)
Steel (2.8442488071422)	Steel (0.227350)	Obstetrics (0.017278)
Hip_(anatomy) (0.49552550392879)	Calluna (0.147419)	Pregnancy (0.015524)
Natural_childbirth (2.2884339744585)	Hip (0.040379)	Massage (0.013779)
	Midwifery (0.030253)	Flora of Europe (0.009764)
	Childbirth (0.025395)	Ericaceae (0.007791)
	Water Birth (0.023574)	Flora of the united kingdom (0.007398)
	Doula (0.022400)	Flora of Estonia (0.007316)
	Home Birth (0.016767)	Human reproduction (0.005184)
	United States (0.015284)	Biota of Europe (0.004901)
	Bradley Method Of Natural Childbirth (0.015112)	Car safety (0.004807)

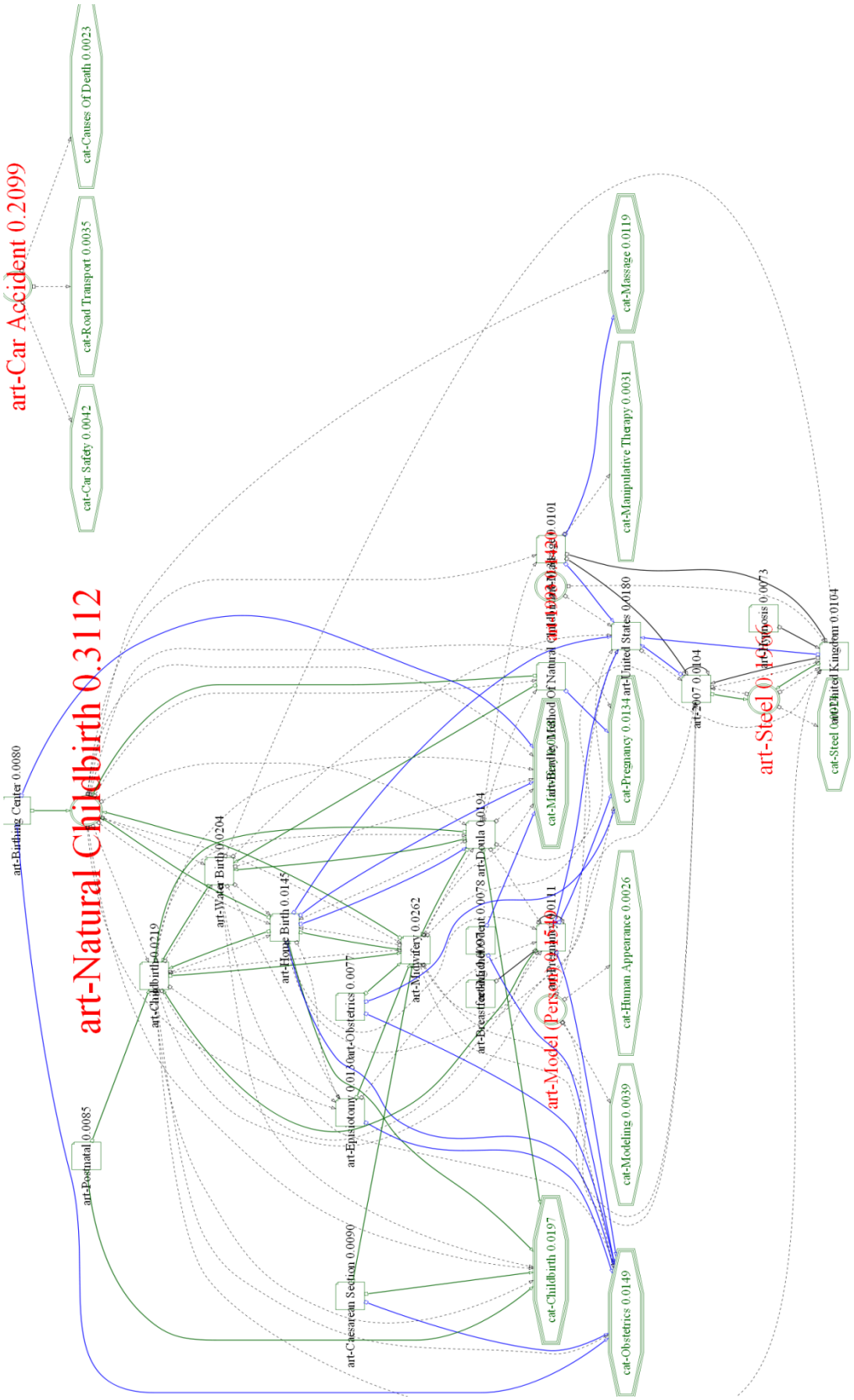

Figure 6.3 Ranking and Linking Caused by Processing S2 with keyRatio=0.2

Table 6.2 The Highest Ranked Elements for Sentence S2 in Figure 6.3

WIKIFY EXTRACTIONS	RANKED WIKIPEDIA ARTICLES	RANKED WIKIPEDIA CATEGORIES
Model_(person) (2.2839645868849)	Natural Childbirth (0.31120864)	Childbirth (0.01970198)
Car_accident (3.0834056428651)	Car Accident (0.20992723)	Midwifery (0.01676163)
1993 (2.0804726952873)	Steel (0.19658996)	Obstetrics (0.01493419)
Steel (2.8442488071422)	Model (Person) (0.15485232)	Pregnancy (0.01341762)
Natural_childbirth (2.2884339744585)	1993 (0.14200896)	Massage (0.01191193)
	Midwifery (0.02615248)	Human Reproduction (0.00447890)
	Childbirth (0.02192087)	Car Safety (0.00415642)
	Water Birth (0.02038179)	Healthcare Occupations (0.00363202)
	Doula (0.01936613)	Modeling (0.00392524)
	United States (0.01796719)	Manipulative Therapy (0.00314229)
	Home Birth (0.01448955)	Human Appearance (0.00264305)
	Bradley Method Of Natural Childbirth (0.01306585)	Steel (0.00244624)

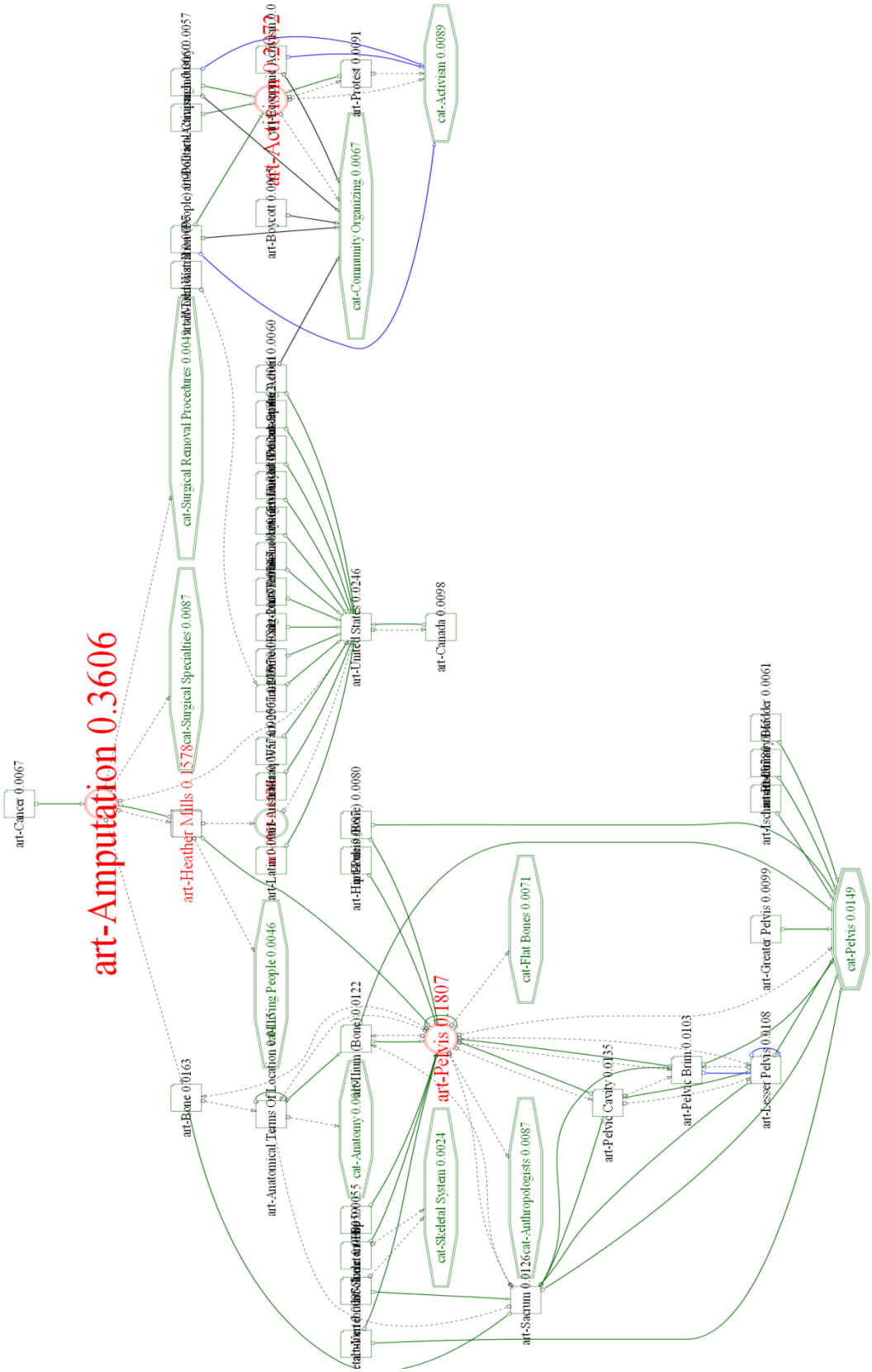

Figure 6.4 Ranking and Linking Caused by Processing S3 with keyRatio=0.2

Table 6.3 The Highest Ranked Elements for Sentence S3 in Figure 6.4

WIKIFY EXTRACTIONS	RANKED WIKIPEDIA ARTICLES	RANKED WIKIPEDIA CATEGORIES
Activism (3.4142802219599)	Amputation (0.36056986)	Pelvis (0.01489139)
Heather_Mills_McCartney (2.6020019872088)	Activism (0.20721374)	Activism (0.00889853)
Pelvis (2.6722097000295)	Pelvis (0.18073598)	Anthropologists (0.00871858)
1993 (2.0804726952873)	Heather Mills (0.15780620)	Surgical Specialties (0.00870814)
Amputation (3.0041831923338)	1993 (0.12695001)	Flat Bones (0.00710031)
	United States (0.02461082)	Community Organizing (0.00672267)
	Bone (0.01630794)	Human Anatomy (0.00544909)
	Pelvic Cavity (0.01351470)	Surgical Removal Procedures (0.00488748)
	Sacrum (0.01257444)	Society (0.00419822)
	Ilium (Bone) (0.01221707)	Medicine (0.00376150)
	Anatomical Terms Of Location (0.01152445)	Humans (0.00243825)
	Lesser Pelvis (0.01075205)	Skeletal System (0.00243360)

Table 6.4 Similarity Metric Values for the Graphs Representing S1, S2, and S3 of Figures 6.2, 6.3, and 6.4. The most similar pair based on a given metric is highlighted.

SIMILARITY METRIC	MORE SIMILAR VALUE WOULD BE	VALUE(S1,S2)	VALUE(S1,S3)	VALUE(S2,S3)
Distance L1	Smaller	2.509497	7.429348	6.299644
Distance L2	Smaller	0.270996	0.722552	0.665972
Cosine	Larger	0.858325	0.013107	0.087506
SkewD	Smaller	2.789524	11.799116	9.357191
ZKL	Smaller	2.277644	13.336911	11.198257
JSZKL	Smaller	0.531241	2.047981	1.754537
NGD	Smaller	0.966299	0.901352	0.916916
JAC	Larger	0.740812	0.455838	0.509417
DICE	Larger	0.999753	1.005713	1.005960
JSSKEW	Smaller	0.440702	1.873167	1.597399

6.2 Related Work

As expected, given the relative importance of so basic a function as detecting textual similarity, a great deal of work has been done in the area. Some methods are detailed in Related Work (Chapter 8). Here I will highlight some of the work directly related to my approach.

As a freely available lexical resource, WordNet has been a natural target for research on comparing the similarity of text. Many of these are based on the taxiomatic and graphical nature of WordNet and use this to compare individual words or concepts listed by synsets. The simplest utilize the number of links required to go from one node in the graph to another, or path distance. This assumes that items with shorter paths are more similar. However, due to the way ontologies and taxonomies are constructed, siblings deep in the graph are often more similar than those found higher, or having to find a connection through a highly placed node, and several methods adjust for this property. Budanitsky and Hirst (2006) gives a survey of several of these methods, and several are described in the Non-Distributional Similarity Measures section in Chapter 8. Others have utilized the gloss that appears in the definition of each synset. These include the Extended Lesk of (Banerjee and Pedersen, 2003) and the Gloss Vector of (Patwardhan and Pedersen, 2006). Pedersen et al. (2004) also provides a general implementation of several common WordNet-based methods in the WordNet::Similarity[18] package, which I used for reference purposes.

Wikipedia has also been utilized by others as a resource for textual semantic comparisons. Gabrilovich and Markovitch (2006) develop Explicit Semantic Analysis, or ESA, which compares input text with each Wikipedia article using TF-IDF and Cosine Similarity. The result of this process is each article can be treated as its own

[18] http://search.cpan.org/~tpederse/WordNet-Similarity/

dimension in a vector space, and vector space comparisons can be applied. (Milne and Witten, 2008b) describe a method modeled after the Normalized Google Distance of (Cilibrasi and Vitany, 2007) where the links shared between words to common articles used to compute a similarity metric. Strube and Ponzetto (2006) adapt WordNet similarity metrics to Wikipedia by utilizing the category network structure.

Many of the methods developed identify the similarity between two individual terms, and not between whole texts. Mihalcea et al. (2006) describes a method for computing the similarity between two texts by combining the individual pairwise word-similarity values between their elements.

Others have also considered the use of PageRank for similarity detection. Hughes and Ramage (2007) provides a model that applies a form of generalized PageRank to compute a stationary distribution across the WordNet graph given an input of individual words of interest, and use it for estimating word similarity. The form of PageRank propagation used corresponds roughly to defining WikiRank's bias as equal to 1 for a given word of interest. The resulting distribution is converted into vector format and compared. One of the primary goals of their work was to incorporate the different types of links between nodes that natively exist or could be defined (such as gloss overlap). They examined the performance of forming links based of different definitions and propose the Zero-KL divergence (described in detail later) to measure the similarity between distributions. In their evaluation they were able to reach the limit of human inter-annotator agreement on the similarity data sets they examined. Ramage et al. (2009) reports on an extension of applying PageRank to WordNet using Cosine, DICE and the Jensen-Shannon metrics.

Agirre et al. (2009) and (Agirre and Sora, 2009) also examine usage of PageRank with WordNet for similarity detection and word sense disambiguation. They reported better

results than (Hughes and Ramage, 2007) using Cosine similarity and disambiguated WordNet glosses to provide an additional set of connections between synsets. This use of disambiguated glosses mirrors the use of links between Wikipedia articles. Their best method for WSD involved linking all words except the target word being disambiguated to the WordNet graph and performing personalized PageRank it, then assigning the target word to the highest valued possibility. This procedure selects the sense that the set of context words cause to be accessed most frequently.

The most interesting point of comparison with WikiRank is called WikiWalk (Yeh et al., 2009), that also extends (Hughes and Ramage, 2007). WikiWalk applies personalized PageRank to Wikipedia, combining random walks initialized with ESA (Gabrilovich and Markovitch, 2006). In this work they evaluated the performance on word semantic relatedness datasets of (Miller and Charles, 1991) and (Finkelstein et al., 2002) and on semantic document similarity using (Lee et al., 2005).

There are points of similarity and difference between WikiWalk and WikiRank. The teleport vector of WikiWalk corresponds to WikiRank's Bias function. While trying to reduce the overall graph size to factor out noise, they inadvertently created partitioned graphs with disconnected islands. To remedy this they proposed initialing the WikiWalk teleport vector with ESA which performs a TF-IDF Cosine Similarity comparison between an input text and all Wikipedia articles. Using the top 625 matches allowed an initial value to be present in each potentially relevant yet disconnected subgraphs. They were able to gain small improvements over the state-of-the-art results reported in (Lee et al., 2005) using ESA as an initializer. However, in contrast to WikiRank, they found additional pruning of their dictionary and Wikipedia link structure gave them better results, and that using all available links decreased their performance. WikiRank uses Wikify! with an adjustable keyRatio instead of the top-n ESA results as its primary text-to-article linking process. Wikify! thus for each word

linked selects just one article (instead of n), performs word sense disambiguation, and provides a parameter to adjust the acceptable level of noise. Given this initialization condition, WikiRank is able to function with the original complete graph structure.

6.3 Methods for the Estimation of Similarity

As shown, using measures of similarity has been a central concept used in many areas that process natural language. In the following section I examine some of the metrics that are computed by the system. This includes those that focus on the distribution, and those that focus on simple textual properties. Additional metrics not used by the system are provided in the section Non Distributional Similarity Metrics in Chapter 8.

6.3.1 Estimating Distributional Similarity

Given that the random walk over a graph produces a probability distribution over all the nodes for a given input, a natural operation is to estimate the similarity or difference between two inputs based on the differences in resulting distribution. If two random walks from two different inputs results in visitation of similar nodes at similar rates, then a natural assumption is that the two inputs are semantically related with respect to the graph. A number of options exist for measuring the similarity between probability distributions. A survey of various methods can be found in (Lee, 2001) and (Mohammad and Hirst, 2005). Here I examine those based on comparisons of vectors and those based on differences in information content and probability distributions.

6.3.1.1 Vector Similarity

One option is to view the value of each node in the graph as a value in the dimension of a vector in a suitable space. In the following, n is the dimensionality of the generated

vector or the number of nodes, while P and Q are the two distributions and P_i is the probability value associated with node i in distribution P.

The *Cosine Similarity* is the measure of the cosine of the angle between the two distributions:

Cosine Similarity Formula

$$\cos(P, Q) = \frac{\sum_{i=0}^{n} P_i Q_i}{\|P\| * \|Q\|}$$

Also, L_1 (or Manhattan distance) and L_2 (or Euclidean distance) are defined as:

L1 Formula

$$L_1(P, Q) = \sum_{i=0}^{n} |P_i - Q_i|$$

L2 Formula

$$L_2 = \sqrt{\sum_{i=0}^{n} (P_i - Q_i)^2}$$

6.3.1.2 Information and Probability Divergences

Another option is to utilize an information theory based Kullback-Leibler divergence. KL divergence measures the extra *number of bits of information required* to code samples from distribution P using a code based on Q instead of using one based on P.

DKL Formula

$$D_{KL}(P||Q) = \sum_{i=0}^{n} P_i \log \frac{P_i}{Q_i}$$

or

$$D_{KL}(P||Q) = \sum_{i=0}^{n} P_i (\log P_i - \log Q_i)$$

One problem with using KL divergence is when the model Q has a zero, in which case the weight of that instance becomes infinite. Modifications to KL have been proposed to handle this condition.

One modification, skew divergence, posited in (Lee, 1999) modifies the formula to mix in the value of *P*, controlled by parameter α. α is set very close to 1 (as in 0.99), where its performance is close to that of KL, and provides both smoothing and protection against undefined values.

Skew Divergence Formula

$$Skew_{\alpha}(P||Q) = \sum_{i=0}^{n} P_i \log \frac{P_i}{\alpha Q_i + (1-\alpha)P_i}$$

The other modification, called Zero-KL divergence (Hughes and Ramage, 2007), utilizes the normal KL formula when it is defined, but uses a fixed smoothing parameter when the Q_i term is zero

Zero KL Formula

$$ZKL_Y(P,Q) = \sum_{i=0}^{n} P_i \begin{cases} \log \frac{P_i}{Q_i}, & Q_i \neq 0 \\ Y, & Q_i = 0 \end{cases}$$

The divergences vary from 0 for maximum similarity to infinity for maximally dissimilar.

NGD is modeled after the Normalized Google Distance of (Cilibrasi and Vitanyi, 2007) and is defined between each possible candidate and all the unambiguous context articles.

Normalized Google Distance

$$NGD(x,y) = 1 - \frac{\max(\log(|X|), \log(|Y|)) - \log(|X \cap Y|)}{\log(N) - \min(\log|X|, \log|Y|)}$$

where X is the set of articles x links to (or in our case is above a threshold), Y is the set y links to, and N is the total number of Wikipedia articles. NGD is a distance metric, and thus 0 is equality and 1 is no similarity. However, in some cases when used on the web NGD will report values greater than 1.

JAC represents the Jaccards Similarity Coefficient which measures the ratio of the size of the intersection divided by the size of the union of the two sample sets.

Jaccards Similarity Coefficient

$$JAC(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

where X and Y are defined as the set of elements in the two distributions that exceed some threshold (in my case 0.0001).

DICE Coefficient

$$DICE(X,Y) = \frac{2 * |X \cap Y|}{|X| + |Y|}$$

DICE also is known as the Sorensen similarity index, and the *Consistency* measure.

Jensen-Shannon Divergence Using ZKL

JSZKL is the Jensen-Shannon divergence using ZKL. Jensen-Shannon (JS) divergence measures the information that would be lost if one used only the average distribution. I defined JSZKL as just substituting ZKL for KL in the equation:

$$JSZKL(P,Q) = \frac{1}{2}\left[ZKL\left(P, avg_{p,q}\right) + ZKL\left(Q, avg_{p,q}\right)\right]$$

$avg_{p,q}$ is the average distribution between p and q.

JS is also known as information radius (Schütze and Manning, 1999) or total divergence to the average, and is symmetric. In this application it would measure how unique each distribution is.

6.3.2 Similarity Tradeoffs

The different similarity metrics being derived from different assumptions have naturally different properties. The L1, L2 and Cosine Similarity metrics are symmetric while the information theoretic ones are not. The intuition behind the information theoretic asymmetry is that the distribution model P may be implied by the distribution model Q, but not the converse, and additional information may be needed to derive Q from P. This leads to a concept similar to entailment, where the distribution generated by one document is implied by the one generated from another. Associated with this would be the divergence of the distribution of a document under analysis from either a

reference document of a domain or other background reference distribution. Areas of divergence would show the areas nodes with higher information content relative to the background.

The Cosine Similarity measure also ignores the relative magnitude of each vector, and responds more to the "shape" of the distribution since when two distributions seen as vectors are pointing in the same direction. The Cosine measure thus shows a form of scale invariance. This property may be useful for detecting the relative importance of nodes. On the other hand, the distance and divergence measures are sensitive to finer-grained differences. Table 6.4 provides an example of these sensitivities. NGD, Jaccards and DICE metrics are set based and the nodes of the graph require conversion into elements of the set based on their values. Automatically adjusting the threshold of set inclusion for optimal performance remains an area for future research.

6.3.3 Estimating Textual Similarity

In contrast to semantic level similarity comparisons are surface level or textual comparisons. These metrics have a natural appeal for the reason that often words or phrases that overlap at the surface have similar meaning either due to inflection or composition. In natural language processing the first is the reason for stemming of words to a common root lemma form. Thus "connect," "connected," "connecting," "connection," and "connections" are all related, and a system like the Porter Stemmer (Porter, 1980) will reduce them to a common form. The compositional form is seen in noun phrases like "medieval religion," "computer technology," or "post-modern architecture" and other noun phrases that share text elements are likely to be in some way related. Cohen (2003a) and (Cohen, 2003b) offers an overview of various text

based methods applied to string similarity estimation while the SimMetrics[19] library offers an open source library of many metrics: textual, vectoral and informational.

In addition to the metrics listed in the previous section on distributional metrics, four additional text based metrics are also computed for comparison and use:

TRI: Letter trigrams similarity
LEV: Normalized Levensthein edit distance
TRIS: trigram applied to the serialized anchor text selected by Wikification
LEVS: Edit distance applied to the serialization of anchor text selected by Wikification

Each of these are computed using the following definitions:

$$Wikification\_of(X) \quad = application\ of\ Wikify!\ to\ string\ X$$
$$WikiAnchors(X) = \ concatenation\ of\ anchors\ in\ Wikification\_of(X)$$
$$ngram(X, N) = \ set\ of\ substrings\ of\ length\ N$$

$$ngram\_similarity(X, Y, N) = \frac{|ngram(X, N) \cap ngram(Y, N)|}{|ngram(X, N) \cup ngram(Y, N)|}$$
$$TRI(X, Y) = \ ngram\_similarity(X, Y, 3)$$
$$TRIS(X, Y) = TRI(WikiAnchors(X), WikiAnchors(Y))$$

LEV and LEVS are based on the common Levenshtein Edit Distance defined in (Levenshtein, 1965), and is the minimum number of insertions, deletions and substitutions operations required to transform one string into another. Each of these operations is given a cost of 1, and is normalized by the maximum string length being examined.

[19] http://simmetrics.sourceforge.net/

$$levenshtein\_edit\_distance(X,Y) = \min_{op_n(\ldots op_1(X))=Y} \left(\sum_{i \in I} Cost(Op_i) \right)$$

$$Op_i \in \{insert(c,i), delete(c,i), subsitute(c,c',i)\}$$

Such that the cost for using each operation is defined by:

$$Cost(Op_i) = \begin{cases} 1, & Op_i = insert(c,i) \\ 1, & Op_i = delete(c,i) \\ 1, & Op_i = subsitute(c,i) \end{cases}$$

$$LEV(X,Y) = \frac{levenshtein\_edit\_distance(X,Y)}{\max(|X|,|Y|)}$$

$$LEVS(X,Y) = Lev(WikiAnchors(X), WikiAnchors(Y))$$

The primary intuition behind these metrics is many paraphrase detection methods operate on the lexical level. The first two (TRI and LEV) function purely at the text level (after normalizing case and removing stopwords). The second two (TRIS and LEVS) perform the same analysis but only on the words selected by the Wikification process at a certain keyRatio level. This depends on the Wikifier for selecting the important fragments in the two texts.

S1: Heather, 35, who lost a leg in a road accident, is thought to have steel plates fitted in her hips, which would make natural childbirth impossible.

W1(kr=0.2): heather road accident steel hips natural childbirth

S2: Former model Lady McCartney lost a leg in a road accident in 1993 and is understood to have steel plates fitted in her hips which would make natural childbirth difficult.

W2(kr=0.2): model road accident 1993 steel natural childbirth

LEV(S1,S2): 0.365

TRI(S1,S2): 0.450

LEVS(W1,W2): 0.314

TRIS(W1,W2): 0.552

Figure 6.5 Example of Metrics LEV, TRI, LEVS, TRIS Applied to a Sentence Classified as a Paraphrase

In the next two sections I will illustrate the application of WikiRank and the similarity metrics using two common tasks that utilize textual similarity detection: recognizing correct answers given by students to questions, and detecting paraphrases from news.

6.4 Short Answer Grading Using Text-to-Text Similarity Comparison

A common task in education is to evaluate what the student has learned. To aid in this task multiple types of examination methods are employed. A common one is to allow the student to provide a short (typically 1-3 sentences) answer to a question. Recognizing an acceptable answer given to a question is non-trivial even when given a reference answer to compare against. The answer given may imply the information sought without using the same words, and may refer to more general or more specific concepts. In order to verify that knowledge has been absorbed and integrated by students the ability to flexibly (and sometimes imaginatively) recognize the acceptability of free form short answers to exam questions is a skill often used.

Here I examine the use of WikiRank applied to recognizing the similarity of student answers against a reference answer. The goal of this series of experiments was to determine if contextually biased ranking could be used as a method for grading by recognizing the quality of a student's answer relative to a given question and reference answer. The primary work is based on comparing the system described here and in (Coursey et al., 2009b) and extending it to handle the UNT Short Answer Dataset and compare the results with the performance of the methods described in (Mohler and Mihalcea, 2009).

WikiRank ranked the encyclopedic nodes using both the reference text and the student answer and measured the difference in activation of the nodes using the various similarity metrics. The performance of the metrics are then compared against the

human provided scores. The research supports the hypothesis that WikiRank could possibly provide a "gist"-based similarity metric which is correlated with acceptable answers, and can do so at a fine grain (within a domain). The performance of the various similarity metrics is compared against the performance of other reported methods using a standard test corpus. Individually the metrics correlate well with human judgments, and when are combined via machine learning were found to be competitive with other existing methods at this task.

6.4.1 Background

Automatic grading of short answers has attracted prior research. Pulmand and Sukkarieth (2005) explored the use of both manually and statistically derived patterns and word matching methods. When matched the derived patterns indicate that a satisfactory answer was given to a question. The manual method of pattern generation required extensive knowledge of the domain and the ability to make accurate answer set predictions. The statistical methods examined required a pre-existing annotated corpus and implied a reuse of the answer sets. In addition, additional machine learning methods including Bayesian, decision tree, and inductive logic programming were explored. Leacock and Chodorow (2003) detailed the C-Rater system, which takes a more syntactic driven approach to extract relational triples using predicate argument templates. This relational approach makes the system sensitive to both word and concept order, something that Bag-of-Word systems find difficult. Mohler and Mihalcea (2009) provides both the data set for this task and the primary point for comparison. In that work, a comprehensive evaluation of knowledge-based and corpus-based methods is given for the short answer grading task. They also examined the effects of corpus size, corpus domain and a form of relevancy feedback. Their best

results were with a domain-specific corpus using Wikipedia coupled with feedback from student answers.

In addition, automatic short answer grading is related to the areas of text similarity detection, and includes related areas of paraphrase detection and information retrieval, including work done with LSA and ESA discussed in the overall Related Work (Chapter 8). The AutoTutor system (Wiemer-Hastings et al., 1999; Wiemer-Hastings et al., 2005; Malatesta et al., 2002) utilizes a Bag-of-Words LSA approach embedded in an interactive "talking head" framework to evaluate and respond to students' answers.

Question: What is the role of a prototype program in problem solving?

Correct answer: To simulate the behavior of portions of the desired software product.

Student answer 1: A prototype program is used in problem solving to collect data for the problem. 1, 2

Student answer 2: It simulates the behavior of portions of the desired software product. 5, 5

Student answer 3: To find problem and errors in a program before it is finalized. 2, 2

Question: What are the main advantages associated with object-oriented programming?

Correct answer: Abstraction and reusability.

Student answer 1: They make it easier to reuse and adapt previously written code and they separate complex programs into smaller, easier to understand classes. 5, 4

Student answer 2: Object oriented programming allows programmers to use an object with classes that can be changed and manipulated while not affecting the entire object at once. 1, 1

Student answer 3: Reusable components, Extensibility, Maintainability, it reduces large problems into smaller more manageable problems. 4, 4

Figure 6.6 Short Answer Corpus Sample Question and Answers, with Grades Provided by Two Human Judges

6.4.2 Experimental Setup

The existing text based biased ranking system was extended in several ways to perform the short answer grading evaluation. Figure 6.7 outlines the process. An overall control program split the dataset into individual files consisting of the reference answer and student answers. Wikify! was used to produce the Wikification (the linking of text to

relevant articles as described in the Wikify! section of Chapter 3) and select the initial articles sets used to represent each text. WikiRank was run for each to produce a visitation distribution. The WikiRank engine used for the experiments in Chapter 5 was modified to be controlled by a dynamic job control language that included both the ability to store and manipulate multiple Wikipedia distributions, and the ability to utilize the various distributional similarity measures as primitives. For each reference-answer pair submitted the control program collects the similarity metric values produced between the two distributions. These similarity values along with the student id, question id, and assigned grade are placed in a summary file indexed by the keyRatio that was used by Wikify! to produce the Wikification. The files are then analyzed using Pearson correlation. keyRatios of 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, and 0.99 were tested.

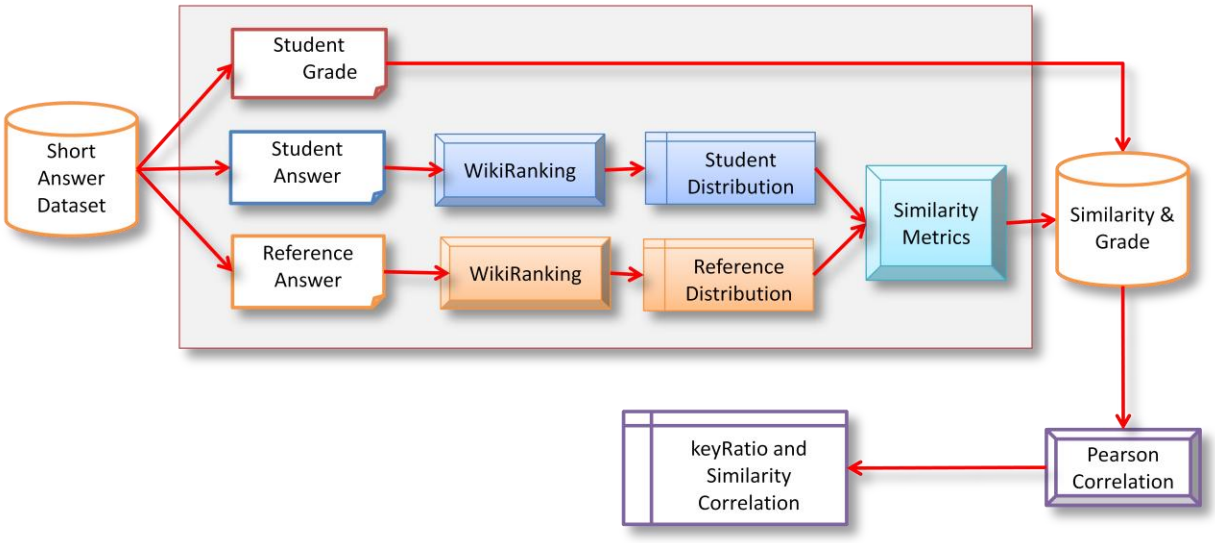

Figure 6.7 Similarity Comparison Process

For this task the reference answer is taken as the *derived* distribution *P* for the informational similarity metrics (outlined earlier in this chapter) that are asymmetrical. This is because while the student answer may be implied by the reference answer, it may be only a part, while the ideal case is the student provides all the information to

derive the reference answer. The reference distribution should therefore be implied by the students' distribution.

6.4.3 Results

As shown in Table 6.6 and Table 6.7, several of the similarity measures tested exceed the baselines of Table 6.5, with L1, Skewd, and JSZKL being competitive with both the best knowledge based and corpus based approaches. Indeed, only full ESA Wikipedia beats *JSZKL by 0.0005*. Interestingly enough, *L1* performs better than *L2* or Cosine similarity. The dimensions of the vector space created by the probability distributions are not uniform, and thus *L1* shows better performance. Also, the set intersection based methods of *NGD*, *JAC,* and *DICE* do not perform as well as the information theory-based measures. This could be due to defining an appropriate threshold for declaring an article as being accessed by both input texts.

Table 6.5 Baseline Per-Question Correlations (Mohler and Mihalcea, 2009)

	METHOD	CORRELATION
BASELINE	TF-IDF	0.3647
	Annotator agreement	06443
KNOWLEDGE-BASED	Shortest Path	0.4413
	Leacock & Chodorow	0.2231
	Lesk	0.3630
	Wu & Palmer	0.3366
	Resnik	0.2520
	Lin	0.3916
	Jiang & Conrath	0.4499
	Hirst & St. Onge	0.1961
CORPUS-BASED (GENERIC)	LSA BNC	0.4071
	LSA Wikipedia	0.4286
	LSA Wikipedia (small)	0.3518
	ESA Wikipedia	**0.4681**
CORPUS-BASED (DOMAIN-SPECIFIC)	LSA Wikipedia CS	**0.4628**
	LSA Slides	0.4146
	ESA Wikipedia CS	0.4385
RELEVANCY FEEDBACK APPLIED	WN Shortest Path	**0.4887**
	LSA Wikipedia CS	**0.5099**
	ESA Wikipedia Full	**0.4893**

Table 6.6 WikiRank Similarity Metric Correlations

METHOD	CORRELATION
L1	0.4557
L2	0.3011
Cosine	0.4138
Skewd	0.4629
ZKL	0.3753
JSZKL	0.4676
NGD	0.2784
JAC	0.3741
DICE	0.2070

6.4.4 Discussion of Individual Similarity Metrics

It was encouraging to find that applying similarity measures to the WikiRanked results can provide competitive results. The fact that all unaided (i.e., no answer feedback) methods seem to reach a maximum near 0.47 may indicate a maximum for methods that utilize either a Bag-of-Words or Bag-of-Concept methods. This may indicate that further performance gains may be had by including syntax and relation sensitive features in the comparison.

Another avenue of research includes exploring ways of combining multiple similarity measures. The best performing measure was *JSZKL*, which used *ZKL* as a component. Yet *ZKL* raw performance was close to baseline. Since *SKEWD* scored highly, a special run with JSSKEWD was made with 0.00001 defined as the threshold for set intersection measures and the keyRatio=0.99. This resulted in DICE=0.2004 while JSSKEW = 0.4692, slightly better than ESA Wikipedia by 0.0011.

Several modifications immediately present themselves. The similarity metrics have not yet been normalized, which would facilitate linear combinations. The application of student-answer-relevancy feedback resulted in a significant increase in the performance of (Mohler and Mihalcea, 2009) and may offer a similar performance increase for the distribution based methods. Another parameter that would bear examination is the threshold used by the set intersection based methods *NGD*, *JAC,* and *DICE*. The fact that *DICE* peaks for keyRatio 0.5 may indicate an optimal point between the threshold used for set construction and the WSD noise induced by Wikification.

Unlike the other tasks, the best performance tends to occur at the high end of available keyRatios (0.9 and 0.99) instead of the low end (0.02 to 0.06) for topic identification tasks. The primary reason is the limited text (1-4 sentences with possibly fewer than 15 words) to link to articles. Thus one needs to link as much as possible, and accept the

WSD error noise. This is coupled with the fact that Wikification tends to focus on noun phrases, since most Wikipedia anchors and articles are about the terms noun phrases refer to. Adding relational analysis should help address this problem. While increasing the keyRatio increases the percentage of raw keywords accounted for, WikiRank is able to utilize the additional knowledge based on its performance above the word-based baseline of TF-IDF.

Table 6.7 keyRatio and Similarity Correlation

KEY-RATIO	L1	L2	COS	SKEWD	ZKL	JSZKL	NGD	JAC	DICE
0.05	0.2283	0.2749	0.2852	0.1929	0.137	0.2232	0.1437	0.2713	0.0438
0.1	0.3134	0.2993	0.3241	0.3116	0.3048	0.283	0.1981	0.2299	0.1982
0.3	0.2844	0.171	0.3538	0.2908	0.2632	0.3006	0.2467	0.2544	0.1081
0.5	0.3721	0.2271	0.3854	0.4336	0.3713	0.3909	0.2356	0.2796	0.3124
0.7	0.4289	0.2706	0.3864	0.4378	0.3701	0.4349	0.2605	0.3233	0.2264
0.9	0.4509	0.2953	0.4138	0.4629	0.3732	0.464	0.2784	0.3741	0.207
0.99	0.4557	0.3011	0.4073	0.4612	0.3753	0.4676	0.2762	0.3664	0.1866

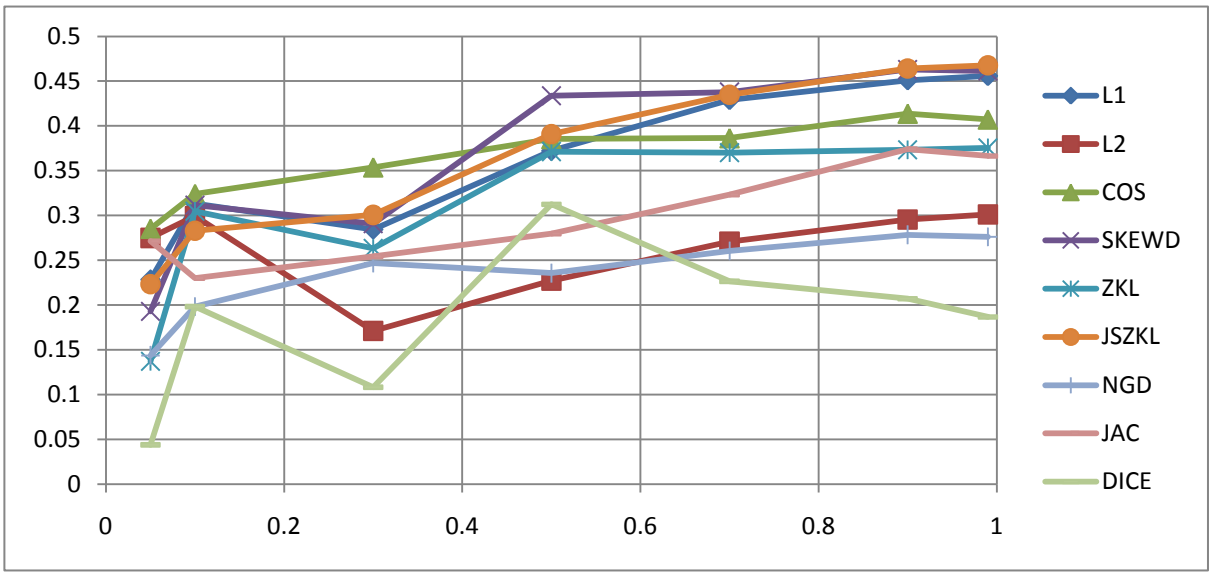

Figure 6.8 Correlation of Similarity Metrics at a given keyRatio for Short Answer Grading

6.4.5 Machine Learning Applied to Short Answer Learning Similarity Metrics

Given the good correlation results of the individual metrics, it is natural to seek ways in which they can be combined to enhance performance. The similarity metrics between the vectors generated by a reference input and a test can be used to generate a *summary metric vector* with each dimension being the value of each metric and the summary metric vector can be used by machine learning algorithms to test if additional information can be extracted. This should be possible since each metric makes different assumptions about what should make two distributions similar.

To test this hypothesis, the similarity metric vectors for the Short Answer Grading corpus files were processed using the machine learning package WEKA (Witten and Frank, 2004). A 10-fold Cross-validation test was performed on each machine learning algorithm available on the whole SAG set with keyRatio set at 0.99. An examination of the possible learning algorithms show several are able combine the information from the different similarity metrics to improve on the correlation of any single one alone, doing better than other methods.

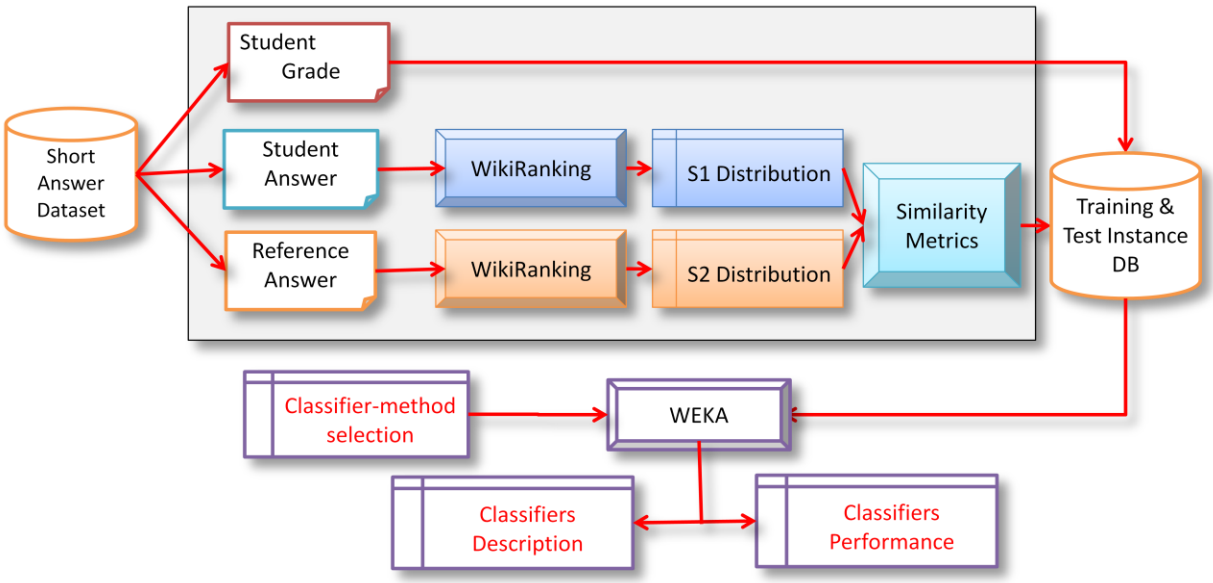

Figure 6.9 Supervised Machine Learning Evaluation

Table 6.8 Correlation Performance of Various WEKA Classifiers Using Similarity Metric Vectors for Short Answer Grading

METHOD	CORRELATION COEFFICIENT
Linear Regression	0.3005
M5P	0.4895

Two of the classifiers generated were able to outperform the LSA full Wikipedia score of 0.5099. A full listing of classifiers is in Table B.1 of Appendix B.

6.4.6 Discussion

The short answer grading evaluation shows that the metrics when applied to the distributions generated by the biased PageRanking operation do correlate with the human grading judgment on par with ESA and LSA based methods. In addition, when the features are used as inputs to a supervised learning system, additional improvements can be gained. Each metric utilizes a different set of assumptions, and provides a condensed higher order description of similarity and difference. In applications where a finer grained evaluation is needed, the relative scores of specific article activations could be included and would be an area of future research. The supervised learning case shows that the similarity metrics do provide enough information to evaluate the quality of answer better than the baselines, and that existing supervised learning systems can combine the information in the different metrics to improve performance.

6.5 General Paraphrase Recognition

Detecting paraphrases of texts is an active area of research in natural language processing and text analysis. Developing effective and robust means of detecting paraphrases would positively impact the area of both information retrieval and

plagiarism detection along with aiding answer extraction and machine translation technologies. In this section I examine the performance of using biased ranking relative to other methods for detecting paraphrases of text. This work supports the hypothesis that WikiRank can provide a gist-based semantic similarity metric useful for a wider domain (while the Short Answer Grading explored a narrow domain). Note also that the short answer grading data set has a scaled grading, whereas paraphrase data set examined has a binary yes/no classification. The system is shown to perform comparably to other methods when evaluated against a standard paraphrase corpus.

6.5.1 Background

The primary method explored to date involves the use of lexical comparison methods to provide a similarity value based on textual overlap. Methods used to improve the performance of algorithms have involved normalization through POS-tagging, stemming, stop-word lists, subsequence matching, and in general various weighting schemes (Salton and Buckly, 1997). While improving performance relative to simple lexical matching, they lack the ability to take into account the similarity on a semantic level.

Various methods of similarity comparisons based on the word-to-word level exist, with some outlined in the chapter on Related Work (Chapter 8). Mihalcea et al. (2006) examines ways in which various word-to-word comparison methods can be combined to provide whole text-to-text evaluation.

6.5.2 Primary Resource

The Microsoft Research Paraphrase Corpus (MSRPC) (Quirk et al., 2004; Dolan et al., 2004) was chosen for evaluation. The MSRPC contains 4076 training and 1725 testing pair sentences. This corpus of paraphrase pairs was collected over an 18-month period

from various Web news sources, and manually labeled by two human annotators as to semantic equivalence. The intra-annotator agreement was approximately 83%, defining the upper bound for recognition. Several evaluations have been done by others using this resource.

P1: The jawbone is similar to those of other early modern humans found in Africa, the Middle East and later in Europe.

P2: Most of their features were similar to those of early humans whose fossils have been found at sites in Africa, the Middle East, and later in Europe.

Figure 6.10 Positive Paraphrase Example

N1: Russian stocks fell after the arrest last Saturday of Mikhail Khodorkovsky, chief executive of Yukos Oil, on charges of fraud and tax evasion.

N2: The weekend arrest of Russia's richest man, Mikhail Khodorkovsky, chief executive of oil major YUKOS, on charges of fraud and tax evasion unnerved financial markets.

Figure 6.11 Negative Paraphrase Example

Two natural baselines were reported in (Mihalcea et al., 2006). A random baseline returned true/false randomly for each pair. The vector-similarity baseline used Cosine Similarity with TF-IDF weighting over lexical terms, in the way commonly used in information retrieval. In addition the performance of Pointwise Mutual Information using Information Retrieval (PMI-IR) and Latent Semantic Analysis were examined. Accuracy is computed relative to the correctly classified instances in the test data set. Precision, Recall and F-measure are given for the positive paraphrase instances. Concurrent with my development, (Ramage et al., 2009) also applied PageRank over WordNet to the dataset, and used the Cosine, DICE, and Jensen-Shannon metrics for comparisons.

Table 6.9 MSRPC Paraphrase Recognition Baselines

METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
Random	51.3	68.3	50	57.8
Vector-based	65.4	71.6	79.5	75.3
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
Mihalcea, et al., 2006 (combined)	70.3	69.6	97.7	81.3
Ramage, et al., 2009 (RW Cosine)	68.7	n/a	n/a	78.7
Ramage, et al., 2009 (RW DICE)	70.8	n/a	n/a	80.1
Ramage, et al., 2009 (RW JS)	68.8	n/a	n/a	80.5

6.5.3 Experimental Setup

Performance was tested in a manner similar to the short answer grading system outlined in the previous section. Each record in the corpus consists of a pair of sentences and the human evaluation of relevance. Wikify! is used to extract initial candidate articles for each sentence to provide the initial biasing points and biased PageRank over Wikipedia is performed. For each pair of sentences the distributional similarity metrics are collected and a training record is produced. In this system, in addition to distributional similarity metrics, the four textual metrics (TRI, LEV, TRIS, LEVS) are also included in the feature set. The system was tested on a nearly complete set of the MSRPC (4075/1722 versus 4076/1725 due to 4 returning null processing errors). The machine learning system WEKA was then used to generate and evaluate the performance of the various classifiers. A performance scan over a subset of records determined that a keyRatio of 0.2 produced the best results for this task.

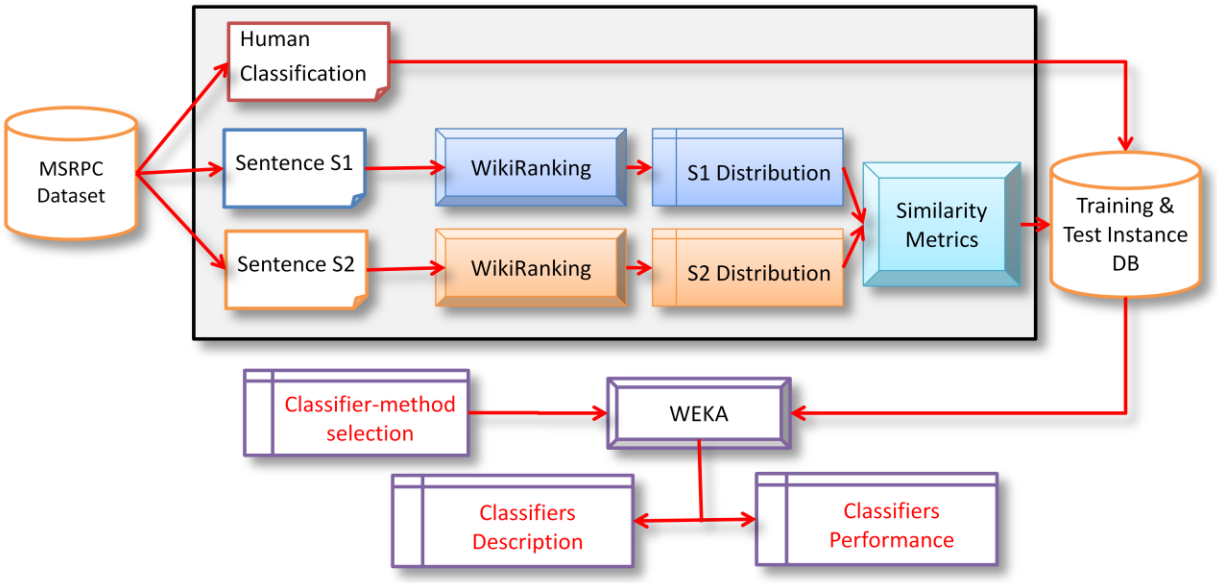

Figure 6.12 Paraphrase Evaluation

Table 6.10 Correlation of Functional Methods with Paraphrase Classification

ANALOG METHOD	CORRELATION
Linear Regression	0.4282
MP5	0.4282

Table 6.11 Performance of Classifiers for Paraphrase Classification

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BFTree	72.1835	0.745	0.884	0.809
J48	72.1835	0.745	0.884	0.809

6.5.4 Results and Discussion

The correlation in Table 6.10 was found to be similar to that of the Short Answer Grading task, and the final performance of supervised classifiers in

Table 6.11 was in the range set between LSA, PMI-IR and the metric of (Mihalcea et al., 2006) which combined several similarity measures into a composite value. The best classifiers performance occurred when the text level features are included, and all utilize TRI as the highest weighted or initial component. Initial testing showed a

preference for TRI (simple trigrams) and LEVS (Levenshtein edit distance applied to Wikification anchors), but the full test preferred simple TRI. Why would the best supervised results occur with relatively shallow processing?

An examination of the dataset and the criteria used by the evaluators is useful. First, strict logical entailment is not enforced in the human grading process. Instead pairs marked equivalent contain "mostly bidirectional entailments." Each sentence of a pair marked as a positive example can contain additional information not contained in the matching sentence. Dolan et al. (2004) reports that "experiments aimed at making the judging task more concrete resulted in uniformly degraded interrater agreement". They also felt that by enforcing strict bidirectional entailment only trivial changes similar to one or two word changes would be considered as positive examples. Thus to make the dataset interesting, they required a minimum word-based Levensthtein distance of at least eight (8). Also, due to the method of collection, the negative instances can vary in their semantic relatedness. They caution that "not equivalent" should not be taken as negative training data.

A system should thus focus on identifying the information the annotators used in their decision process. A simple two part strategy becomes apparent. The first step uses the trigram similarity metric and is able to identify cases where there is substantial low level textual overlap. If however the raw textual similarity is not sufficient, one can examine the edit distance applied to the serialized Wikify! anchor text. While this process seems textually based and not very knowledge-intensive, it actually does apply a great deal of knowledge. The Wikify! process utilized several million anchor annotations by humans identifying that information which they found relevant to aid others' understanding. When seen in this light, a great deal of lexical knowledge or *compiled interest* is in fact embedded in the Wikification process. In addition, while the Wikification selection of terms may be at times be noisy, the selection of which words to

focus on is less so. Currently the system has a higher probability of selecting the correct word in the Wikified text to annotate than selecting the correct final article to link that annotation to, which would affect the vector based similarity metrics. Increasing the Wikification to the point where it finds relevant anchors may start to introduce additional noise in the ranking process. In these cases identifying what is most important in a sentence may be a better strategy than identifying what it entails.

WEKA linear regression function derived the following formula with a correlation of 0.42:

```
grade = (1.0743 * TRI) + (-0.3854 * JSZKL) + (0.3482 * JSSKEW) +
        (-0.2136 * TRIS) + (-0.182 * LEV) + 0.368
```

Also classifiers with similar performance to each other were generated and their strategy examined. The first was produced by WEKA's J48 (an implementation of the C4.5 (Quinlan, 1993) decision tree classifier) performs a split using the TRI metric, with those passing a threshold being taken as a paraphrase. Those that do not pass the TRI metric test are tested using LEVS. A similar strategy was found by the WEKA's Naïve Bayes Tree (NBTree) classifier (Kohavi, 1996) which combines decision trees with Naïve Bayes classifiers in the leaves. NBTree selected the first branch to split the data using the TRI metric, creating two subsets with different *a priori* classification distributions. The first branch had a roughly 3:1 ratio for paraphrase-vs.-non-paraphrase while for the other branch the ratio was 1:3. Each sub-classifier could then operate with this pre-filtering.

Another factor to consider is the length of text passages being analyzed. With longer text segments a system can lower the keyRatio and thus utilize higher probability annotations, and thus provide higher accuracy rankings. Such a system would find

usefulness in analysis of summaries of longer texts in a manner similar to the ROUGE (Lin, 2004) summarization evaluation system.

One natural question is how does system performance degrade if TRI and LEV are removed from the set of available features?

Table 6.12 Correlation without TRI or LEV in Feature Set for Paraphrase Classifiers

ANALOG METHOD	CORRELATION
Linear Regression	0.2171
MP5	0.2171

Table 6.13 Binary Classifier Performance without TRI or LEV in Feature Set

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BFTree	67.420	0.678	0.974	0.799
J48	67.015	0.680	0.951	0.793

The correlation drops significantly comparing Table 6.12 to Table 6.10 indicating that there is no simple linear combination of features that correlates as well without TRI or LEV. The performance of the binary classifiers based on F-measure drops as well when comparing Table 6.11 to Table 6.13 . For the binary classifiers the recall increases while the accuracy and precision falls. However, the binary classifiers are able to combine the available features into an overall paraphrase recognition strategy. Table B.2 and Table B.3 provide the full classifier performance list.

Figure 6.13 and Figure 6.14 below show two binary classifiers generated without TRI or LEV as features. In both cases they utilize the trigram similarity of the selected Wiki anchors. DISTL2 is the Euclidian vector distance, while JSZKL measures the information that would be lost if the average of the two were used (the shared information), and DICE measures the consistency or overlap of sets of articles selected.

Note that a TRIS value of 0 does not mean that Wikify! could find nothing, just that there was no overlap between what it found most interesting in each sentence given the keyRatio constraint.

```
TRIS <= 0.039216
|   JSZKL <= 2.583237:  class = 0 (160.0/80.0)
|   JSZKL > 2.583237
|   |   DICE <= 0.854881: class = 1 (4.0)
|   |   DICE > 0.854881: class = 0 (64.0/9.0)
TRIS > 0.039216: class = 1 (3847.0/1188.0)
```

Figure 6.13 J48 Classifier without TRI or LEV

The following is a translation of the classifier in Figure 6.13:

RULE 1	
IF TRIS > 0.039216 THEN Class = 1: paraphrase (accuracy 76.4%)	
	Rationale: High enough overlap of the anchors.
	Example with (TRIS = 0.602941, JSZKL = 0.614078, DICE = 1.007773):
	LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion - Australian football - as the world champion relaxed before his Wimbledon title defence.
	LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion "Australian rules football" as the world champion relaxed ahead of his Wimbledon defence.

RULE 2	
If TRIS <= 0.039216 AND JSZKL <= 2.583237 THEN Class =0: no-paraphrase (accuracy 66.7%)	
	Rationale: Not enough textual support to justify the high shared information content. Different terms were marked as important but lead JSZKL indicated both visitations are close from an information viewpoint.
	Example with (TRIS = 0.0, JSZKL = 1.857789 , DICE = 0.715169):
	Kroger Co., which owns Ralphs, and Albertsons Inc. bargain jointly with Safeway and locked out their union workers the next day.
	In a show of corporate solidarity, Kroger Co., which owns Ralphs, and Albertson Inc. locked out their workers the next morning.

RULE 3	
IF TRIS <0.039216 AND JSZKL>2.583237 AND DICE > 0.85 THEN Class=0: no-paraphrase (accuracy 87.7%)	
	Rationale: A broad topic area with broad graph overlap based on Dice value but no textual support and not enough shared information content.
	Example with (TRIS = 0, JSZKL = 2.672534, DICE = 0.984055):
	The tech-heavy Nasdaq Composite Index .IXIC was off 0.11 percent, or 1.78 points, at 1,594.13.
	The broader Standard & Poor's 500 Index. SPX was down 0.04 points, or 0 percent, at 971.52.

RULE 4	
IF TRIS <0.039216 AND JSZKL>2.583237 AND DICE <=0.85 THEN Class=1: paraphrase (accuracy 100.0 %)	
	Rationale: Lacking hard info simply use the default class.
	Example with (TRIS = 0.0, JSZKL = 3.299039, DICE = 0.739823):
	That investigation closed without any charges being laid.
	The investigation was closed without charges in 2001.

In the tree of Figure 6.14, generated by another decision tree generator called CART (Breiman et al., 1984), the Euclidian distance (L2) of the two visitation distributions is the first attribute tested and if close enough, the pair is classified as a paraphrase. If L2 is above a threshold then TRIS is compared to examine the amount of overlap in extractions. In this case CART develops a simple strategy that first examines the visitation distribution, and if the two distributions are not close enough, examines the overlap of interesting details.

```
DISTL2 < 0.46273: class = 1(1279.0/362.0)
DISTL2 >= 0.46273
|  TRIS < 0.03935: class = 0(135.0/91.0)
|  TRIS >= 0.03935: class = 1(1382.0/826.0)
```

Figure 6.14 CART Classifier without TRI or LEV

The following is a translation of the classifier in Figure 6.14:

RULE 1	
IF DistanceL2 < 0.46273 THEN Class=1: paraphrase (accuracy 77%)	
	Rationale: The encyclopedic networks values are close enough semantically.
	Example with (DISTL2=0.275064, TRIS=0.56):
	Because of the accounting charge, the company now says it lost $1.04 billion, or 32 cents a share, in the quarter ended June 30.
	Including the charge, the Santa Clara, Calif.-based company said Monday it lost $1.04 billion, or 32 cents per share, in the period ending June 30.

RULE 2	
IF DistanceL2 >=0.46273 AND TRIS < 0.03935 THEN Class=0: no-paraphrase (accuracy 62.5%)	
	Rationale: Far enough semantically AND not enough anchor support
	Example with (DISTL2=0.684854, TRIS=0.034483):
	A European Union spokesman said the Commission was consulting EU member states "with a view to taking appropriate action if necessary" on the matter.
	Laos's second most important export destination - said it was consulting EU member states "with a view to taking appropriate action if necessary" on the matter.

RULE 3	
IF DistanceL2 >=0.46273 AND TRIS >= 0.03935 THEN Class=1: paraphrase (accuracy 59.7%)	
	Rationale: Far enough semantically, however, the anchors overlap. Semantic mismatch may be due to a well connected noisy anchor.
	Example with (DISTL2=0.519943,TRIS=0.633333):
	The S&P 500 and the Nasdaq indexes recorded their third straight week of gains.
	Both the S&P 500 and the Nasdaq indexes have scored three straight weeks of gains.

While a great deal can be done with the semantic similarities, detailed commonality information appears necessary for this dataset. For the MSRPC corpus, simple letter trigrams appear to make a good baseline feature to include, facilitating error reduction by being able to note the 'obvious cases' where there are either high overlap or no-overlap. It also suggests that inclusion of first order semantic features, the actual scores given specific articles, in addition to the second order summary features derived from

those scores may be useful for other applications. Coursey (2007) showed an example of this using inferred connections to terms in the Cyc ontology as features for text classification. In addition, the overall similarity comparison using biased ranking of encyclopedic knowledge allows multiple similarity metrics and knowledge sources to be integrated in a way similar to (Mihalcea et al., 2006).

6.6 Conclusion

When augmented with an appropriate biasing function and similarity metrics, WikiRank can indeed be used to provide features to recognize paraphrases both in specific and broad domains. The short answer grading task illustrated the correlation between the individual metrics and human judgments, placing them in the range of performance as ESA and LSA, which would make a competitive unsupervised recognition process. When combined in a framework that utilizes the similarity metrics as features for machine learning, the metrics can be fused together to create task-specific processing strategies. I also introduced the metrics LEVS and TRIS which perform overlap comparisons of the anchor text selected by the Wikification process. LEVS and TRIS utilize the compiled interest choices embodied in the millions of human annotation choices that Wikify! uses when it performs Wikification. It is by using everything available that the method outlined in this chapter is able to develop a strategy that can recognize a correct answer when it sees it.

CHAPTER 7

ONTOLOGICAL TERM SIMILARITY

> This sense of Sameness is the very keel and backbone of our thinking.
>
> — *William James, The Principles of Psychology, 1890*

One important feature of the biased ranking process is the ability to map various items into a common space defined by encyclopedic knowledge relevancy. This becomes particularly important if more items on the semantic web are finding mappings to Wikipedia and other encyclopedic based or scaled systems (DBpedia, Freebase, Linking Open Data Initiative, etc.). One of the primary hypotheses of this work is that inputs denoting similar concepts from different sources will generate similar patterns of visitation in the encyclopedic knowledge network. Conversely, similar patterns in the network will denote that the same or similar underlying thing or phenomena is causing stimulation to the network, which may have different names in different ontologies. The ability of the system to notice similarities relative to human judgments has been outlined in previous chapters.

In this chapter I will examine two approaches to using biased ranking to compare terms from two ontologies. The first assumes that links already exist from the ontologies being compared into a common encyclopedic reference, and the rankings generated (similar to topic identification) can be used to derive the similarity. The second method merely assumes that terms in each ontology have a textual description, and utilize the paraphrase recognition method developed in the previous chapter. This "recognition through reading" method showed encouraging results.

7.1 Pairwise Comparison of Ontology Term Similarity

In this section the ranking process is examined for the ability to use a common encyclopedic knowledge source (Wikipedia) to return plausible ranked lists of terms from an external ontology. For terms in each ontology I use existing links to Wikipedia to perform WikiRank to produce a list of relevant Wikipedia entries. This allows the similarity metrics of the previous chapter to be used for comparisons. Such a comparison can provide a starting point to find relational links between different ontologies. The focus of this section is direct term-to-term comparison between ontologies.

7.1.1 Test Domain

To test the mapping approach I utilized two of the broadest coverage ontologies currently available: Cyc and YAGO-SUMO. Each is detailed in Background Materials and Resources (Chapter 3). Both are broad coverage formal ontologies with desirable characteristics (as described below), and with some degree of manual verification. The YAGO-SUMO and Cyc ontologies provide mappings between each of their internal terms, and to the WordNet ontology and to Wikipedia. However, YAGO-SUMO and Cyc do not currently have direct links between each other. Each WordNet synset is a *set* of associated words and a gloss along with associated relational links to other WordNet synsets. Given that each ontology provides official links to Wikipedia, one can use the WikiRank process starting with these "official" links, and use the WordNet links to provide a common basis for comparison.

To compute the similarity between elements of the two ontologies, the algorithm is placed in a framework to compute the Cartesian product of the similarity between

elements in both ontologies. An approximation is used to improve the efficiency of the overall process to make the computation tractable.

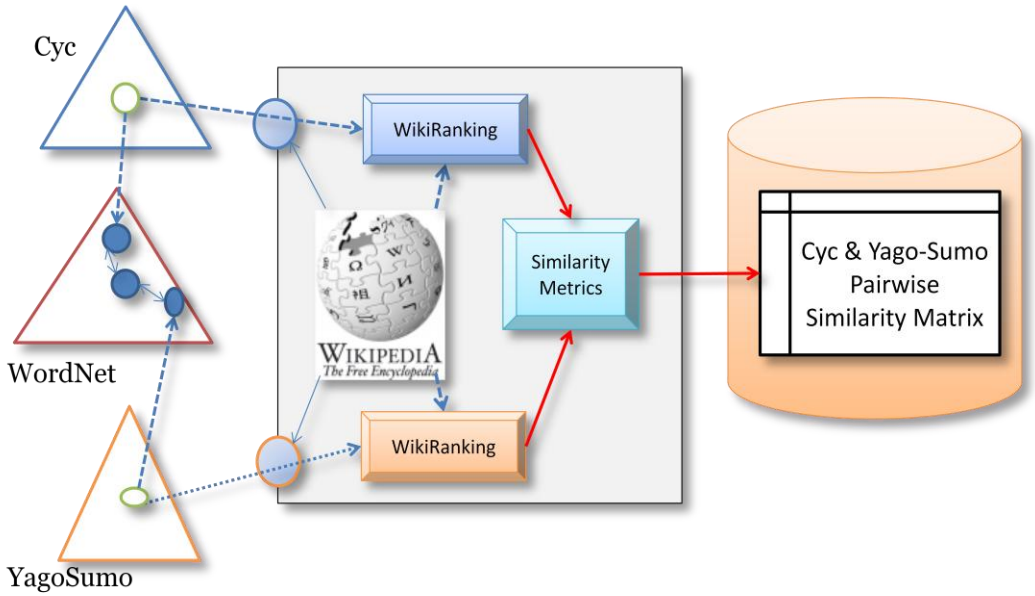

Figure 7.1 Relationship for Testing WikiRank Coverage and Mapping

7.1.2 Experimental Setup

Examining the similarity between terms in Cyc and YAGO using WikiRank is currently a computationally intensive process. Each term must be mapped into initial Wikipedia terms and then a biased PageRank produced. Then for each pair of terms in the two ontologies the value of the similarity metrics must be produced using each Wikipedia rank vector. This matching task given the size of the ontologies requires hundreds of billions of similarity computations. Because of this, the computation was reformulated to utilize a pre-existing framework designed for high-throughput computations: the MapReduce Framework.

7.1.2.1 Parallel Processing with MapReduce and Hadoop

Fortunately most of the pairwise similarity computation can be formulated in a highly parallel way. The creation of the distribution vectors for each ontological term is independent of each other. Other operations involve aggregation of information with respect to either one or two ontology terms. These are attributes that fit well inside of a MapReduce (Dean and Ghemawat, 2004) framework distributed over multiple machines.

MapReduce formalizes the basic processing pattern of performing a transformative computation over a set of independent records (the *map* phase), and then performing an operation over aggregated results (the *reduce* phase). Each record consists of a (key, value) pair each of which can be either simple or complex objects. Each phase can be null, and normally the output of the map phase is sorted by the output keys into bins that are then sorted and presented to the reducers. An important perquisite of the reduce phase is that all records with the same key will go to a unique reducer and in one contiguous set. This means that a reducer can perform min, max, union, intersection, summation and other aggregation based operations associated with a given key.

Hadoop[20] (White, 2009) is an open source Java-based implementation of MapReduce designed for clusters of computers. Hadoop also implements HDFS which is a fault tolerant distributed file system. While a native Java system, Hadoop provides a streaming interface which allows any program written in any language that supports the Unix Standard Input-Standard Output (STDIO) stream model. This includes both C/C++ (which the existing ranking engine is written in) and Perl (which is used for the similarity and other computations). Hadoop provides scheduling, monitoring, fault

[20] http://hadoop.apache.org

detection and recovery, and transparently performing the sort phase required between the map and reduce operations.

The primary process was run on the Amazon EC2[21] cluster using their Elastic MapReduce service. Twenty (20) machines were used (1 master, 19 workers). To maximize the resources, the largest instance available was used with each machine having 4 virtualized cores, 15 GB of RAM, and 1690GB of disk. The initial ranking mapper process is fairly large and computationally intensive. Selecting the EC2 large machine allows the ranking task to run internally as a mapper.

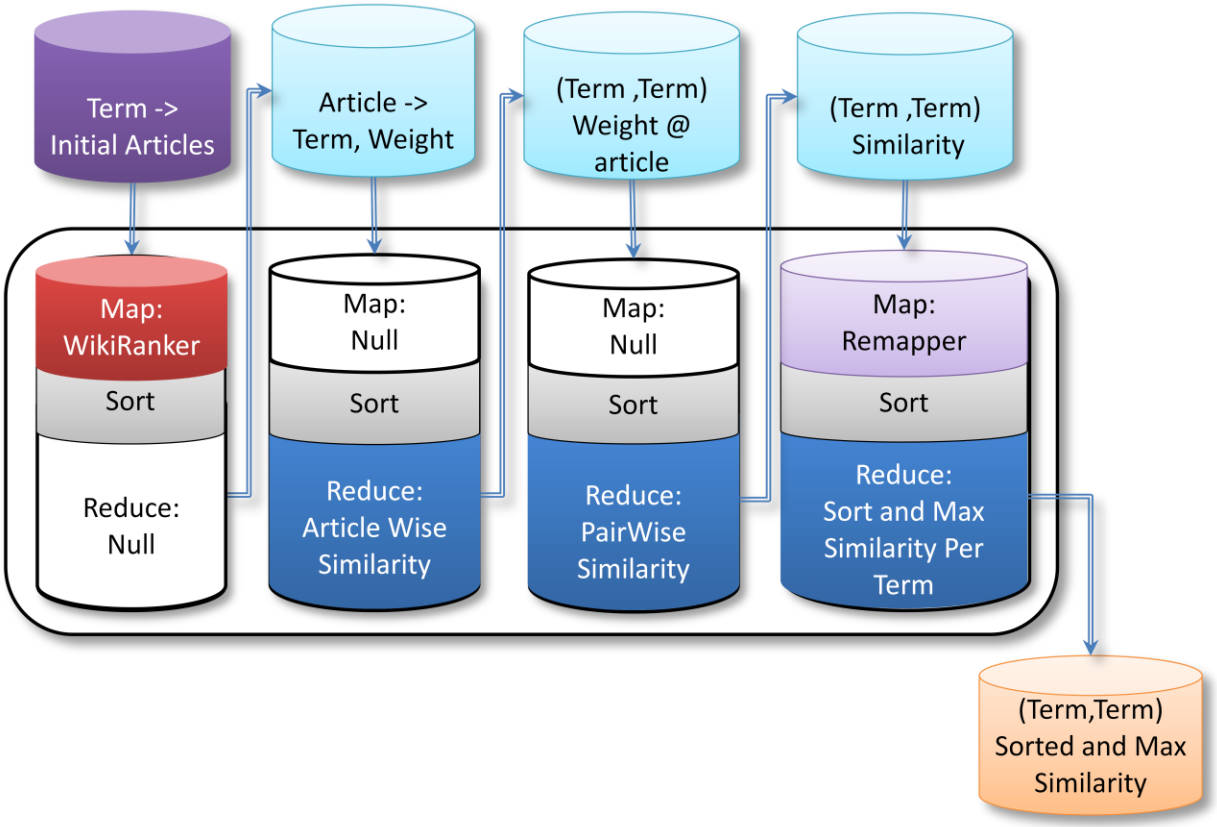

Figure 7.2 MapReduce Steps to Compute Term-Term Similarity

[21] http://aws.amazon.com/ec2/

Table 7.1 Data Volume at Each Stage

STAGE	INPUT RECORDS	INPUT SIZE (BYTES)	OUTPUT RECORDS	OUTPUT SIZE (BYTES)
Ranking	192 K	81,575 K	40,613 K	3,382,058 K
Article Co-Similarity	40,613 K	3,382,058 K	572,758 K	64,969,514 K
PairWise Similarity	572,758 K	64,969,514 K	2,889,089 K	294,575,862 K
Max Similarity	2,889,089 K	294,575,862 K	3,410 K	319,462 K
Sorted Similarity	2,889,089 K	294,575,862 K	19,572 K	1,851,621 K

7.1.2.2 Ranking and Similarity Detection in a MapReduce Framework

In order to estimate the similarity between terms in the two ontologies a batch ranking and co-similarity computation process was generated. This process is outlined in Figure 7.2. The existing ranking engine was modified to work with the Hadoop framework as a streaming application. In this format the ranking engine accepts records describing the ontological term and the initial Article ID's as specified by either Cyc or YAGO-SUMO:

(key = <ontologyTerm>,value = [<article-ID1>…<article-IDN>])

The ranking engine produced a listing of the top 200 ranked articles and their values. The Ranking mapper is fairly intensive for a current generation mapping process. Implementing unbiased PageRank in MapReduce for Wikipedia is a canonical exercise, and PageRanking the web was one of the initial uses of MapReduce by Google that sparked current widespread interest. While unbiased PageRank over Wikipedia is a typical example run on a small cluster of computers, each mapper is in fact that process, each mapper requiring a gigabyte to store its reference material. Here WikiRank is encapsulated as the mapper processes of the first stage in Figure 7.2 and is run for each entry from each ontology. The output records for the ranking stage are:

(key= <articleID>,value=<ontologyTermA,value>)

The next stage accepted the individual article-based scores and produced records where the keys were two ontology terms and the value contains their two values and the article they have those values at. This stage implements the filter process proposed by (Lin, 2009), where articles that have more entries than a threshold are not passed. The reason to implement a filter is that this is a $O(N^2)$ operation, and such articles are similar to stopwords in information retrieval, generating many additional records with low discriminatory value. By using a filter in this way only the most discriminative values are emitted to the next stage, and those that are eliminated are universally eliminated from consideration. The threshold was set as 500 for categories and 150 for all other entries. Table 7.1 illustrates that even using this filtering process results in large data volumes can still be seen. The output of this stage is of the form:

(key=<ontologyTermA , ontologyTermB>, value=<weightA, WeightB, articleID>)

The pairwise similarity process accepts all weights for a given pair of ontological terms and produces the composite similarity value for each metric. The output of this stage is:

(key=<ontologyTermA , ontologyTermB > , value=< metric , value >)

Two processes accept the output of the similarity processor, one to find the maximum for each metric, and one to find the top N (where N=6 for testing). In this step a simple mapper transforms each input record so the sorter or maximum finder can operate over all values of the first term. This remapper produces:

(key=<ontologyTermA>,value=<ontologyTermB, metric,value>)

The filtered output of the sorter and maximum finder is:

(key=<ontologyTermA, ontologyTermB>,value=<metric,value>)

7.1.3 Evaluating the Quality of Similarity Metrics for Ontology Matching

One of the first tasks is to provide a method to evaluate the quality of a proposed solution. The elements of the two ontologies examined, Cyc and YAGO-SUMO, provide links to Wikipedia and the ranking process can be applied, resulting in a visitation distribution of the nodes of the Wikipedia graph. This distribution is taken as a vector and the previous similarity metrics are applied, and correlated with a set of base associations.

7.1.3.1 Cyc to Wikipedia Linkage

We extracted an initial mapping of terms in the Cyc ontology to Wikipedia articles. A semantic web compatible version of OpenCyc[22] was used, and the process used allowed more general terms inherit the links of more specific terms. Thus "Dog" would inherit the links made between specific collections of dogs and Wikipedia entries. This file contains 11234 Cyc to WordNet 3.0 mappings and 19104 Wikipedia articles, with 4800 mappings to both resources.

7.1.3.2 YAGO-SUMO to Wikipedia Linkage

We extracted the initial mapping of Wikipedia entries from the N3 version of YAGO-SUMO. YAGO-SUMO provides the relation "y:describes" that links ontology terms to Wikipedia entries. As in the Cyc case, links to a specific term are propagated to the more general class for that term. Also, YAGO-SUMO has classes like "wikicategory_Forts_in_Maine". These are converted into the equivalent of "Category:Forts in Maine" and a link is created if a corresponding entry can be found in the Wikipedia graph. YAGO-SUMO has 3663 entries that directly connect both WordNet 3.0 and Wikipedia.

---

[22] http://sw.opencyc.org/downloads/opencyc_owl_downloads_v2/opencyc-2009-04-07-readable.owl.gz

7.1.3.3 Common Linkages

The Cyc OWL file contains 11234 Cyc-to-WordNet 3.0 mappings and 19104 Wikipedia URLs with 4800 Cyc terms having both Wikipedia and WordNet links. Both YAGO-SUMO and Cyc share 4723 WordNet 3.0 entries. However, the two ontologies do not share any common Wikipedia URL's, and none of the entries that share WordNet links have Wikipedia links.

YAGO has 3663 entries that directly connect both WordNet 3.0 and Wikipedia. It is harder to quantify the YAGO-to-WordNet mapping since most of the terms in YAGO are direct WordNet instances, but also have leaf node connections from elements representing Wikipedia entries.

So, simply put, there are no direct matches using Wikipedia. Zero simple cycles of length 4 linking Cyc, YAGO-SUMO, WordNet, and Wikipedia exist. This means that while Cyc and YAGO-SUMO may both have links to a common WordNet term, either one will lack a direct link to Wikipedia.

7.1.3.4 WordNet Path Distance as a Gold Standard

One question is how to objectively evaluate the mappings proposed by the system between ontologies. Both Cyc and YAGO-SUMO have links to WordNet, and those that they both share imply that the two terms are related to the concept denoted by the synset. This common set is taken as a "gold standard" since they were manually verified by the ontology creators. Using this gold standard one can verify how well any mapping function performs at recognizing similarities that correspond to the standards-defined linkages.

First, the subset of Cyc and YAGO-SUMO where a WordNet linkage could be identified was examined. For these, Wikipedia linkages were generated for the terms from each

ontology and a ranking vector was produced. Since each term maps to WordNet, the path distance between the corresponding WordNet entries could be measured using the WordNet::Similarity package (Pedersen et al., 2004) and associated with each pair, providing a graded gold standard. This then allows both correlation and evaluation of binary classifiers for determining if two terms should be considered within a certain distance with respect to WordNet.

Table 7.2 Example of ranking YAGO-SUMO entries given Cyc Term Polygon using string matching and Ranking Methods. String methods find exact match while ranking offers range of topical matches. Gold Standard is "Poloygon" in both YAGO-SUMO and WordNet

RANKING	1	2	3	4	5	6
Trigrams	polygon 113866144	poison 115032376	polymer 114994328	position 106208751	police station 103977678	policeman 110448983
Leven-shtein	polygon 113866144	poison 115032376	soliton 107346344	pylon 104028764	polymer 114994328	baryon 109215023
WR COS	Quadri-lateral 113879126	triangle 113879320	polygon 113866144	Johnson solids	Pyramids and bipyramids	Catalan solids
WR L1	Quadri-lateral 113879126	Archimedean solids	Platonic solids	polygon 113866144	Catalan solids	figurate numbers
WR L2	Arithmetic problems of plane geometry	Helicopter manufacturers of Japan	Sicilian mathe-maticians	elliptic functions	figurate numbers	rice dishes
WR ZKL	figurate numbers	quadrilateral 113879126	rice dishes	Archimedean solids	Platonic solids	Polyforms
WR SKED	Platonic solids	Archimedean solids	Pyramids and bipyramids	Obstacle billiards	mathematical theorems	Catalan solids
WR JSZKL	figurate numbers	quadrilateral 113879126	rice dishes	Archimedean solids	Platonic solids	Polyforms

Table 7.3 Selection of each method for YAGO-SUMO terms matching Cyc concept "ReligiousBuilding." Gold standard would be "place of worship" for both YAGO and WordNet

RANKING	1	2	3	4	5	6
Trigrams	religious leader	religious school	religious person	religious holiday	religious festival	office building
Levenshtein	religious leader	religious person	religious holiday	religious school	religious festival	office building
WR COS	Temples in the Campus Martius	Chapels in France	Roman Catholic churches in Ive arrondissement	Ancient Roman buildings and structures in Rome	Mosques in Jerusalem	Ancient Roman architects
WR L1	villages in Devon	Italian sculptors	Visitor attractions in France	Mosques in Jerusalem	Chapels in France	Roman Catholic churches in Ive arrondissement
WR L2	villages in Devon	Italian sculptors	Visitor attractions in France	Churches in Italy	Mosques in Jerusalem	Italian Baroque painters
WR ZKL	Hindu temples by deity	Chapels in France	Roman Catholic churches in Ive arrondissement	Churches in France	Churches in Siena	Mountains of Jerusalem
WR SKED	Arches and vaults	Taoist temples in the United States	Roman Catholic churches in Paris	Basilica churches	1st century Roman sculptures	churches in Montreal
WR JSZKL	Hindu temples by deity	Chapels in France	Roman Catholic churches in Ive arrondissement	Churches in France	Churches in Siena	Mountains of Jerusalem

7.1.4 Results and Discussion

Table 7.2 and Table 7.3 provide examples of the ranked results for the various similarity functions. While the textual metrics provide a strong first choice for "Polygon," the remainder are less topical. Also, the semantic similarity metric choices tend to be

related to the topic of "Polygons," even if in more imaginative ways. As for "ReligiousBuilding," the text-based offerings are less topical than the semantic similarity offerings, which are more directly related to religious structures, their locations, or things directly associated with them.

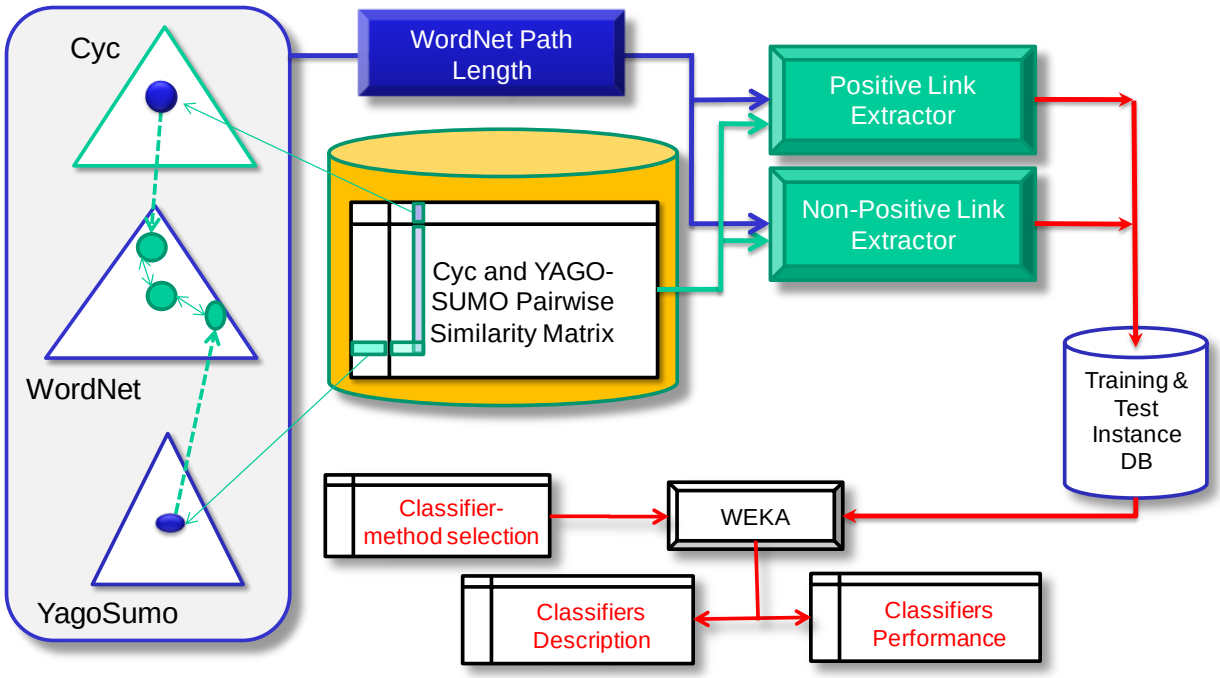

Figure 7.3 Generation of Pairwise Matching Evaluation Dataset

An evaluation and training set was generated using the process outlined in Figure 7.3. This consisted of a merger of those pairs identified previously as gold standards based on common WordNet linkages, those with non-zero WordNet::Similarity::Path metric, those with Cosine Similarity above a threshold and a random selection of those for which no WordNet connection could be found. Each training instance contains all the similarity metrics including TRI and LEV, and an evaluation equal to the WordNet::Similarity::Path value or zero if it is undefined. Two sub training sets were extracted from the pairwise similarity matrix and merged. The first subset was filtered by the list of term pairs for which known links exist. The second subset was unfiltered

but did include those pairs with non-zero cosine values (which the first subset might ignore). The similarity vectors and WordNet distances are merged into a final record so their usefulness can be evaluated. The WEKA machine-learning package was used to test performance of various classifiers. The data set constructed consisted of 2702 training and 1314 test instances, with the similarity vector as the feature vector and the WordNet::Similarity::Path value as the predicted variable. Table 7.4 shows the strong correlation between the regression classifiers and WordNet distance.[23]

Table 7.4 Correlation between Classifier for Cyc/YAGO-SUMO and WordNet Distance

ANALOG METHOD	CORRELATION
LinearRegression	0.7685
MP5	0.7813

Table 7.5 shows the performance of binary classifiers generated to recognize a path similarity measure above a threshold of 0.2. The majority of the classifiers constructed show both good accuracy and precision, with moderate recall and F-measure.

Table 7.5 Classifier Performance for Cyc/YAGO-SUMO Relevancy Detection

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BFTree	92.24	0.883	0.503	0.641
J48	92.16	0.815	0.558	0.662

One key goal of exploring the use of WikiRank and the various similarity metrics was to develop a useful feature set for evaluating the similarity and relevance of objects mapped into the common knowledge space. The evaluation again tested the ability of existing machine learning methods to utilize the information encapsulated in the similarity metrics in a setting different than the previous chapter.

[23] Additional details in Subproject: Ontology Vector Classification in Appendix A

There are many ways to utilize abstracted knowledge. Coursey (2007) showed that machine learning methods could utilize the formal ontology Cyc as a feature space allowing commonsense human interpretable descriptions of the developed classifiers. In this work I show that other ways exist to utilize association knowledge in the Wikipedia network through the use of similarity metrics. Having each metric derived from different assumptions provides a different viewpoint on what it means for two objects in a common semantic vector space to be similar or different. In this way they create a second order vector space, which is synthesized from the first order space created by the initial biased rankings. The base or surface level would be the actual words which makeup and map to the initial article set corresponding to the set of concepts. Indeed, multiple objects of different types can be mapped into either the surface or conceptual levels, as long as a mapping can be found between the input source and the initial vocabulary, such as the Picture-to-WordNet mapping found in the "80 million Tiny Images" project[24,25].

7.2 Ontological Mapping using Textual Similarity

In this section I examine the performance of the WikiRank method of recognizing that two terms from two different ontologies are indeed denoting the same thing by simply reading their textual descriptions and utilizing encyclopedic knowledge to recognize similarity.

Given two descriptions of a concept from two different sources, would they not be paraphrases of each other? Often the ontologies of interest may not have a direct mapping into a common point of reference. Lacking a common map, the initial method that humans would use to recognize that two terms were in fact describing the same

[24] http://people.csail.mit.edu/torralba/tinyimages/
[25] http://labelme.csail.mit.edu/tool.html

thing would be to read the descriptions. For most ontologies a textual description of their terms either exists or is derivable, thus the ability to utilize such a readily available source of information is desirable. Would it be possible to recognize potential matches between ontologies at a textual level, by simply reading descriptions of the two terms using WikiRank coupled with a suitable biasing function such as Wikification?

7.2.1 Constructing a Test Corpus

The mapping task described in the previous section outlined a major problem: that terms in Cyc and YAGO-SUMO, while having individual maps to both WordNet and Wikipedia, currently no cycles can be made directly linking elements from all four. For the task of this section I utilized a higher frequency mapping—that between Cyc and WordNet. Both Cyc and WordNet provide text describing the terms in each ontology. In Cyc this is the "comment," while in WordNet it is the "gloss." By reading each we can test the ability to recognize known equivalent concepts described by different sets of editors. Effectively, these should be paraphrases describing the same core concept.

In this experiment, potential Cyc and WordNet mappings were converted into a format similar to the paraphrase task. A corpus was created by pairing text from each ontology and generating a gold standard evaluation for each pairing. For each Cyc term examined, the corresponding WordNet term is known. From this WordNet term several related non-matching terms such as hypernyms and hyponyms can be located and their glosses used. In addition, terms that should be mapped to other Cyc terms can also be located. Since each Cyc term is paired with several WordNet entries, the evaluation can be based on the distance between each WordNet term and the "ideal" one. This is done by using the WordNet:Similarity:Path function. This is the inverse of the number of links between two WordNet terms, with terms in the identical synsets returning a value of 1, and those with no links having a value of 0. Thus the corpus

contains known 'golden' Cyc-to-WordNet mappings with a value of 1, a number of related mappings with fractional relatedness measure, and several unrelated mappings.

Cyc: SweatGland: SweatGland The animal body part which excretes sweat

WordNet: sweat_gland_n_1:sweat gland any of the glands in the skin that secrete perspiration

Grade: 1.0

Cyc: Monster: Monster The collection of all fictional mythical animals of strange or terrifying shape typically but not always imagined also to be very large in comparison with human beings

WordNet: mythical_monster_n_1: mythical monster a monster renowned in folklore and myth

Grade: 0.5

Cyc: SeaBattle: SeaBattle The collection of events that are battles at sea or on some other body of water

WordNet: iced_tea_n_1: iced tea strong tea served over ice

Grade: 0.06666

Figure 7.4 Example of Test Corpus Entries from Cyc and WordNet with Grades Assigned by WordNet::Similarity::Path Function.

7.2.2 Experimental Setup

Figure 7.4 illustrates the experimental setup. The corpus created was subsampled and randomly split into a training and test set. There were 375 training examples and 176 test examples. The corpus was generated in the same format as the Microsoft Research Paraphrase Corpus (MSRPC) used in Chapter 6 so the existing framework created for paraphrase processing was reused. Members of each set were processed by Wikification with a keyRatio of 0.2 and WikiRank over Wikipedia was performed. The similarity metrics measured between the two resulting distributions were then measured. These training and test sets were then processed by WEKA to evaluate both the correlation and the ability to develop classifiers for each machine learning method. For the correlation case the raw WordNet path similarity score was used, while for the binary classification case only those with a path similarity greater than 0.2 (5 links) were

considered as positive examples, and the rest were labeled as non-examples. This effectively separated the exact/hypernym/hyponym pairs with values greater or equal to 0.5 from the mass of irrelevant terms at 0.125 and below as no pairs in either set had grades between these two values.

As a baseline the performance of the Most Frequent Class classifier can be used, which select the most frequent class in the training set. Statistics are reported in the same way as was done for the MSRPC evaluation in Chapter 6. Accuracy is computed relative to the correctly classified instances in the test data set. Precision, Recall and F-measure are given relative to the instances matches that are considered matching (i.e. grade >= 0.5).

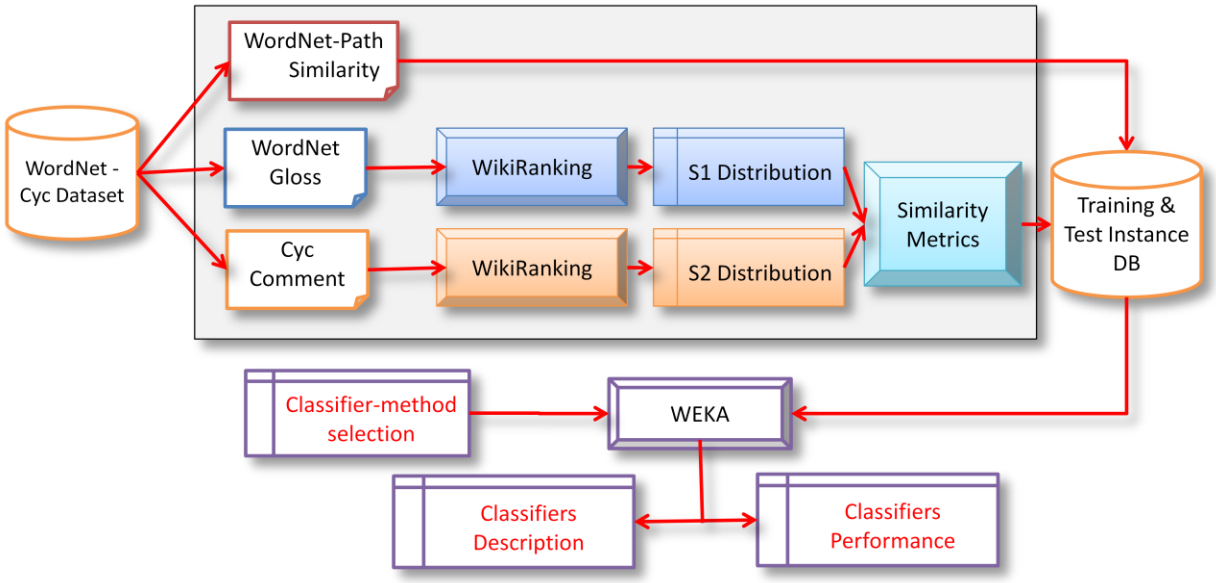

Figure 7.5 Basic Flow of Cyc–WordNet Textual Evaluation

Table 7.6 Members of Each Class in Training and Test Sets

CLASS	TRAINING	TEST
Non-matching	140	61
Matching	235	115

Table 7.7 Correlation of Functional Classifiers with WordNet Path Similarity Metric

ANALOG METHOD	CORRELATION
LinearRegression	0.6854
MP5	0.6854
Most Frequent Class	0.2506

Table 7.8 Binary Classifier Performance at Recognizing Near Matching Cyc–WordNet Terms

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BFTree	86.3636	0.942	0.843	0.89
J48	78.4091	0.953	0.704	0.81
Most Frequent Class	65.34	0.653	1.00	0.79

7.2.3 Results and Discussion

The results of processing the corpus are encouraging. The machine learning methods were able to detect a good correlation between the similarity metrics provided and the measured path similarity as seen in Table 7.7. WEKA was also able to develop classifiers with high accuracy, precision and overall high F-measure as seen in Table 7.8[26]. Interestingly enough, the F-measure for reading (albeit on a smaller sample) was significantly higher than that for direct links used in the previous section that used the ontology provided linkages. One reason for this could be that full Wikification of the descriptive text provides better links than the inheritance mechanism employed in the direct linkage version. This is a positive result since every entry in the ontologies examined posses a textual description, but not all terms have an inheritable direct link into Wikipedia. The results also indicated that the transformation steps of Wikification plus WikiRank required to map ontology text into the common encyclopedic knowledge space preserves enough information content that the similarity metrics

[26] The complete set of classifiers can be found in Subproject: Cyc-WordNet Paraphrases in Appendix A

using these estimated visitation distributions as input can indeed discriminate relevant from irrelevant pairings.

Previous work has also considered language-based ontology matching. Lin and Sandkuhl (2008) provides an overview of combining WordNet and linguistic methods to perform ontology matching, while (Mascardi et al., 2009a) examined the use of a WordNet-based version of the Lesk algorithm (Lesk, 1986) combined with a ontological constraint reasoner written in Prolog to find and repair matches using disjoint and contradiction detection between ontologies that could be mapped into WordNet. Lesk-based methods would view ontology matching as similar to WSD tasks, selecting the closest match based on textual overlap in the definitions. Sarjant et al. (2009) illustrates using logical constraints to improve the precision of ontological matching and merging Wikipedia entries into Cyc. Both (Mascardi et al., 2009a) and (Sarjant et al., 2009) demonstrate reasoning over the constraints placed on the underlying ontologies definition to recognize potential definitional conflicts that would occur if potentially incorrect links were made.

The positive performance of the paraphrase system suggests a simple linking mechanism where a language modeling technique recognizes that a definition is being given in text, then a method similar to the one explored here finds the closest matching definition in various ontologies. A sanity check similar to (Mascardi et al., 2009a) and (Sarjant et al., 2009) could be utilized to ensure consistency. Additionally, the paraphrase similarity classifier could be embedded as an evaluation oracle component inside of a larger ontology mapping search process. This external mapping process could then utilize the nature of the ontologies being explored to intelligently propose matches to examine and utilize both connectedness and disjointedness constraints to explore the space of matching possibilities.

The operation of recognizing ontological similarity through paraphrase similarity can be reformulated in the MapReduce framework of the previous chapter. All of the steps would be the same as detailed in the section on Pairwise Comparison of Ontology Term Similarity except the initial Wikification would be based on the textual description instead of the inherited links. This Wikification step is linear to the total number of terms to be examined in both ontologies, and in fact the bulk Wikification task can become its own MapReduce stage. Indeed, by using MapReduce, concepts in all ontologies of interest could be cross compared, provided the vectors were sparse enough.

It is the existence of broad coverage encyclopedic references that allows WikiRank to find both relevant association and entry points. With this coverage it appears that the simple expedient of recognizing definitions from different sources as paraphrases could become a valid approach.

7.3 Conclusion

In this chapter I examined two approaches for using WikiRank to identify possible matching terms used in differing broad coverage ontologies. The first method examined a more exhaustive cross comparison based initializing WikiRank with direct links into Wikipedia defined in each ontology, and managed by the MapReduce framework. This method was able to produce classifiers with high-accuracy and precision. The sorted results indicate that indeed semantically-related terms are more closely associated by the similarity metrics output than those offered by the textual methods examined. The other method of reusing the work done on paraphrase detection shows that the simple expedient of recognition through reading of the natural language description and definition could provide an interesting means of matching terms. The classifiers produced offered high-accuracy, precision, recall and some

offered relatively high F-measure. It does appear that biased ranking, given a suitably broad encyclopedic graph providing background knowledge, can indeed be one method for recognizing potential matches across ontologies.

CHAPTER 8

RELATED WORK

> A record, if it is to be useful to science, must be continuously extended, it must be stored, and above all it must be consulted.
>
> – *Vannevar Bush, As We May Think, 1945*

When speaking of encyclopedic knowledge there are a number of different types. Lexicons contain information for a specific language. Dictionaries contain knowledge idiosyncratic to a particular word, such as its part of speech, senses, origin, relation to other languages and usage. Encyclopedic knowledge includes everything about a concept. The underlying software of Wikipedia is MediaWiki[27], which allows references to pictures, sound, video and indeed any information representable as a file or accessible by URL. Dictionaries often contain encyclopedia knowledge to aid interpretation, while encyclopedias make reference to the various surface forms used in dictionaries. All of these references works are indexed by some vocabulary since they have been traditionally indexed linguistically.

Since the focus of an encyclopedia is to include information and knowledge on a wide range of topics, a variety of them exist and different systems have been created to utilize their contents or organization. I will describe some encyclopedic knowledge sources, some systems that use them (and are relevant to this research), and finally some general methods for estimating the similarity of concepts that can be related by the use of encyclopedic knowledge.

---

[27] http://www.mediawiki.org

8.1 Common Sense Knowledge Bases

One relevant area is the collection of common sense knowledge for use by computers. These knowledge bases collect information that is not typically explicitly mentioned in human readable encyclopedias but encode the knowledge typically assumed to be known and used by the typical reader of an encyclopedia. These include "relationships implied to be possible, normal, or commonplace in the world." Common knowledge can be viewed using an information channel model, and is the information the transmitter considers the receiver to already know. If the transmitter were to send this information then either:

- The receiver would consider the transmitter to "lack enough intelligence or experience" to know to filter out non-informative content.
- The receiver would believe that the transmitter thinks that the receiver "lacks intelligence or experience."
- Possibly the transmitter is clarifying among many possible common options they mean in a particular case (sense disambiguation).

Since both parties possess the common knowledge, to send it would result in negative information content. This partially explains why it is difficult to find explicit common sense knowledge, even on the Internet.

8.1.1 Open Mind Common Sense and ConceptNet

Open Mind Common Sense (OMCS) is a project initiated by (Singh et al., 2002). The goal of the project is to collect a large commons sense KB from web participants. Information collected is freely available and is used by other projects. As of spring 2009, OMCS has collected over a million statements in various languages, with over

830,000 in English. After parsing and analysis the ConceptNet (Havasi et al., 2007; Liu and Singh, 2004) and LifeNet (Singh and Williams 2003) semantic networks were derived. ConceptNet contains approximately 250,000 assertions. An example use of ConceptNet involved using it to improve speech recognition accuracy by using context and commonsense to perform disambiguation (Lieberman et al., 2005).

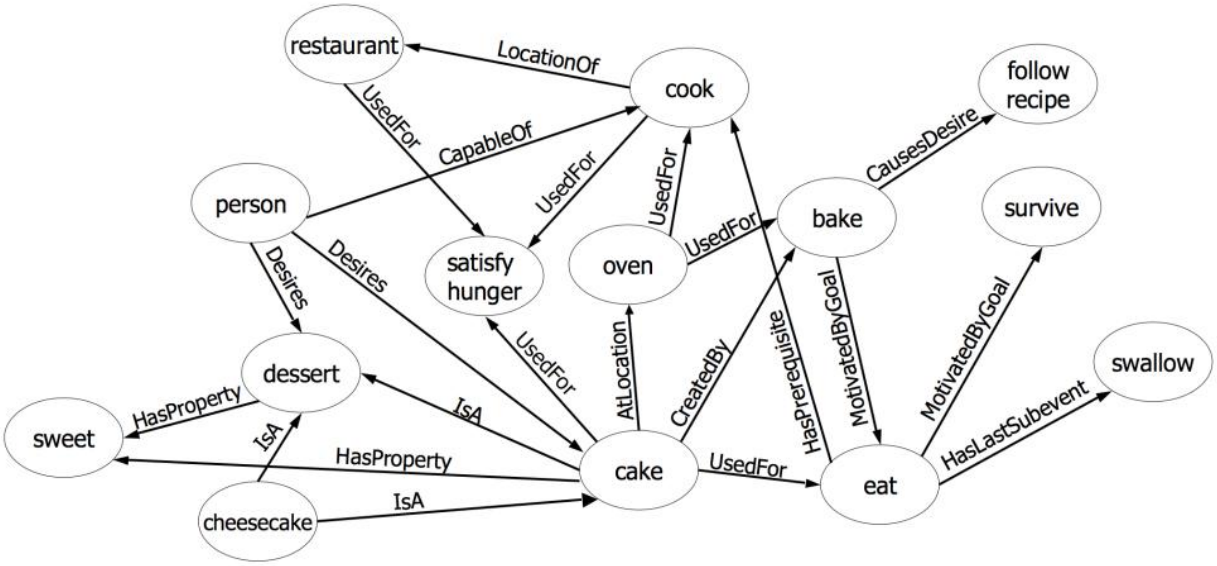

Figure 8.1 Small Section of ConceptNet

Some of the latest work (Speer et al., 2008; Speer et al., 2007) involves the use of Principle Component Analysis to create vector space representation called AnalogySpace[28], to show larger scale patterns, smooth over noise, and predict which data should be in the KB. This is related to the LSA methods described later in this chapter. Since relationships are described as triples (a common semantic web format), each triple can be seen as a pair of "concept/property" relations:

originalTriple = (conceptL, relation, conceptR)

leftPropertyPair = ((conceptL, relation), conceptR)

[28] http://analogyspace.media.mit.edu/

rightPropertyPair = (conceptL, (relation, conceptR))

One can then form an occurrence matrix using the property pairs, and perform similarity computations.

8.1.2 Mindpixel

Mindpixel[29,30] collected a KB of natural language true/false statements (called a 'mindpixel') from human participants on the web and validated by 20 other human participants. A grading mechanism was set up to filter results, and to give greater weight to participants that generated statements that were consistently agreed upon by others. By January 2004 the project had collected 1.4 million mindpixels. Loss of the server and the death of the author brought further development to a halt. McKinstry (2008) reports on research using the information.

8.1.3 ThoughtTreasure

Mueller (2003) describes the Cyc-inspired ThoughtTreasure project. ThougthTreasure includes approximately 100,000 elements of both declarative and procedural knowledge, with 25,000 concepts organized in a hierarchy. It also provides an architecture for NLP, with lexical entries for both English and French. A distinguishing feature of ThoughtTreasure is the use of multiple representations (logic, finite automata, grids and scripts) and the use of procedural processing mechanisms. For instance, questions requiring spatial reasoning can make use of a 2D representation and direct comparisons. The ThoughtTreasure KB was exported in multiple formats including CycL, and API's exist for Python, Java, or a generic socket-based interface.

---

[29] http://en.wikipedia.org/wiki/Mindpixel
[30] http://web.archive.org/web/*/http://mindpixel.com

8.1.4 Cyc and Related Systems

In addition to providing background knowledge to expert systems, part of Cyc's original goal was to provide computers with the implicit knowledge required to understand an encyclopedia. A more detailed description is provided in the Cyc section in Chapter 3.

8.2 Broad Coverage Knowledge Bases (Other than Common Sense)

8.2.1 Prior Referenced Systems and Wikipedia-Related

Several systems have already been mentioned as resources in Chapter 3 that were used during the research. Additionally, an excellent overview of various projects that utilize Wikipedia can be found in (Medelyan et al., 2008b). The *KBs/Ontology Projects Worldwide List*[31] also provides links to many of the projects described in Chapter 3.

8.2.2 MindNet

MindNet (Vanderwende et al., 2005) is a Microsoft knowledge representation project that uses its internal broad-coverage parser to build semantic networks from dictionaries, encyclopedias and text. It builds a semantic dependency graph of the logic form of each sentence it reads and combines the individual subgraphs into a total graph. Weights are associated with the subgraphs using the corpus frequency derived probabilities.

8.2.3 The Stanford WordNet Project

Snow et al. (2006) describe a method for extending WordNet by adding to the semantic network based on relations extracted from parsed text. The algorithm is a best-first

[31] http://www.cs.utexas.edu/users/mfkb/related.html

search over the space of possible hyponym additions to an existing taxonomy while seeking to maximize the conditional probability for the taxonomy being correct given the available evidence. Thus for "microsoft" to be added under "company" there must be evidence for it being compatible also with the hypernyms of company like "institutions" or possible siblings like "dotcom." By starting with 1000 clusters from a 70 million web page corpus, they added an additional 40,000 synsets. The spring 2009 version[32] has 400 thousand synsets defined. The method is defined in a way to evaluate any relationship, not just hyponyms. As such, it may find use in other ontologies, and the information collected can be used to enrich other WordNet-related projects. The same project also offers versions of WordNet with fewer senses, potentially useful for tasks requiring a coarser-grained taxonomy.

8.2.4 TextRunner and KnowItAll

KnowItAll and the TextRunner (Etzioni et al., 2008) projects are efforts to collect large quantities of specific relations using information extraction techniques. TextRunner in particular utilizes a lightweight parsing process based on combining phrase chunks and a Bayesian classifier trained to recognize plausible combinations. This allows it to make a single pass of a corpus extracting a large set of many different relationships in the same time a traditional information extraction system would require to extract a limited set. As of spring 2009, TextRunner extracted 500 million assertions form 100 million web pages. Soderland and Mahdhani (2007) report on translation of the facts extracted into logical expressions and relations suitable for use in an ontology.

[32] http://ai.stanford.edu/~rion/swn/

8.3 General Methods Applied to Encyclopedic Knowledge Sources

8.3.1 Latent Semantic Analysis and Semantic Vectors

Latent semantic analysis (LSA) was developed by (Deerwester et al., 1990) for information retrieval. LSA reduces the large and sparse term-document matrix that describes the term frequency in documents into two separate matrixes using singular value decomposition. The two matrixes encode term-concept and concept-document information. The processing results in mapping both terms and documents into a concept space of much lower dimensionality, while preserving as much information as possible.

LSA is often used to compare documents and terms in the new semantic vector space. Pilato et al. (2007) provides an example of using LSA for mapping specific natural chunks extracted by a conversational agent into specific terms in the Cyc ontology. The text corpus is based on a merging of Wikipedia article contents and the comments describing each Cyc term used as gloss. Text associated with each Cyc term is used to create a vector for each term. By extending the concepts of the CyN chatbot interpreter (Coursey, 2004), two LSA-specific functional tags were added to the AIML language: *sentenceConcept* provides whole sentence mapping between text and Cyc terms, while *relatedConcept* retrieves closely related Cyc terms based on their LSA similarity.

User: I need information about departments related to the study of plants.

Agent: Do you want information about the BiologyDepartment?

User: Can you show me other related Departments?

Agent: I have found a relation with the Agriculture Department.

User: Can you give me information about it?

Agent: It is a specialization of AcademicDepartment, instances of which conduct teaching and research in Agriculture.

Figure 8.2 Example Chatbot Dialog Using LSA-Cyc Mapping

In a similar line of work, (Bodo et al., 2007) examined the use of Wikipedia and LSA for the purposes of text categorization, but reported negative results when used for the categorization of the Reuters-21578 dataset. Mohler and Mihalcea (2009) in contrast reported positive results using subsets of Wikipedia to train LSA models for the short answer grading task.

Related to LSA is the Semantics Vectors package (Widdows and Ferraro, 2008)[33], which uses Random Projection (RP) instead of LSA to perform a dimensionality reduction, and is related to Sparse Distributed Memory of (Kanerva, 1988). The RP formulation is less computationally intensive, can be performed incrementally, and operates by assigning a random vector for each event (which defines a context), and adjusting the vectors of each item that co-occur in that event towards the context vector for that event. The result is that after repeated association each item that frequently co-occur will have similar vectors, and thus creates a reduced dimensionality co-occurrence vector space.

8.3.2 Explicit Semantic Analysis

Explicit semantic analysis (ESA) (Gabrilovich and Markovitch, 2006) was introduced as a way to determine the relevancy of the Wikipedia articles with respect to a given input text, and as an alternative to LSA. Each article is represented as a vector of the words that occur with it. The dimensional values of the word vectors are assigned using TF-IDF, and quantify the association between the words and the articles. Given a new input the system constructs a new vector to represent it using TF-IDF, and then applies a centroid-based classifier (Han and Karypis, 2000) to rank all Wikipedia articles by relevance. This means that instead of the reduced 300-dimension vector used by LSA, ESA would use two million dimensions. Semantic relatedness is computed using the

[33] http://code.google.com/p/semanticvectors/

Cosine Similarity between two terms being analyzed. ESA has been utilized for text categorization.

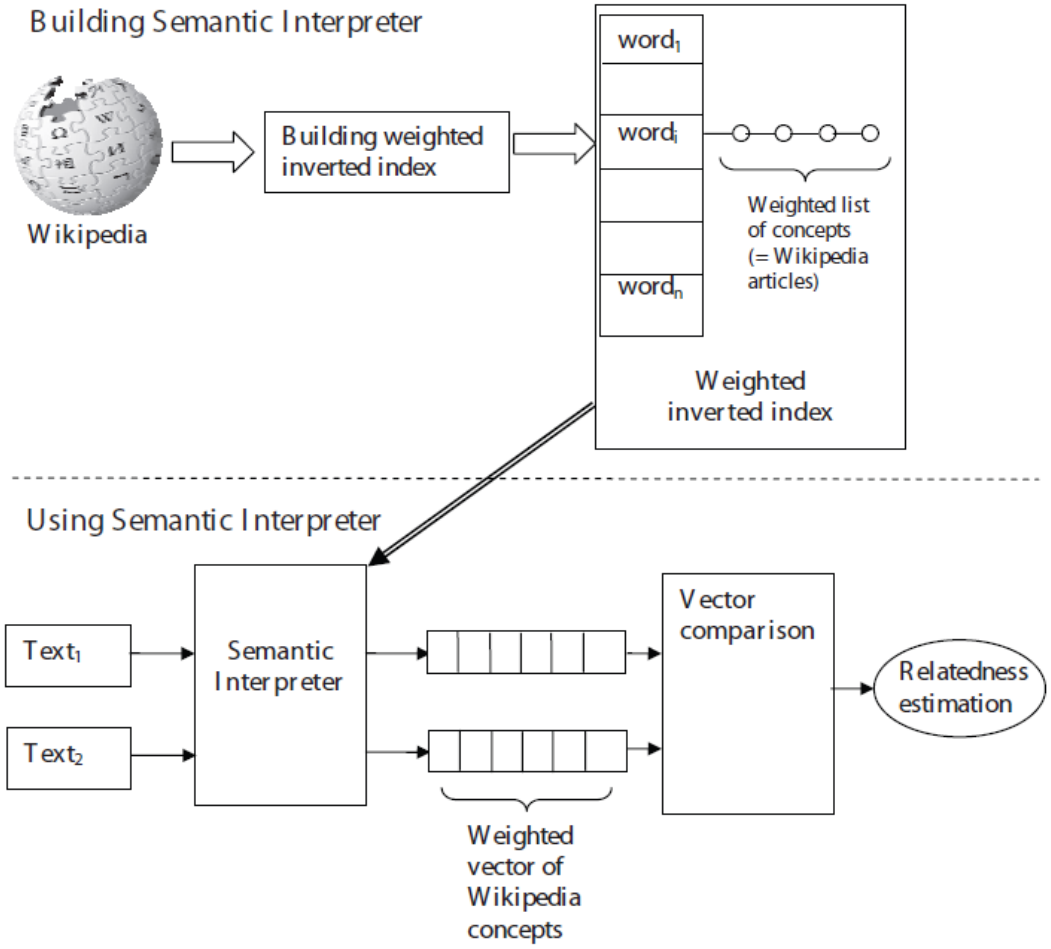

Figure 8.3 The Explicit Semantic Analysis Process

8.3.3 WikiRelate!

In WikiRelate! (Strube and Ponzetto, 2006) adapt the text and graph connectivity based semantic relatedness measures developed for WordNet and applies them to Wikipedia.

The Wikipedia category graph is used to induce a semantic network. Given a word pair, WikiRelate! retrieves the relevant Wikipedia pages. The pages are connected to the appropriate node in the category subgraph and a path is found to connect them.

For their work the best performing metric was the normalized path measure (Leacock and Chodorow, 1998).

Strube and Ponzetto (2006) also examined the use of the informational measure proposed by (Resnik, 1995). The intrinsic information content of a node n is used instead of corpus probabilities:

$$iic(n) = 1 - \frac{\log(hypo(n) + 1)}{\log(C)}$$

where C is the total number of nodes in the hierarchy, and $hypo(n)$ is the number of hyponyms (number of concepts subsumed) of n.

For relational measures based on text-overlap, the text of the first paragraph of the Wikipedia article is substituted for the WordNet glosses in Lesk calculations (Lesk, 1986; Banerjee and Pedersen, 2003).

8.3.4 Wikify!

The Wikify! system generates the automatic annotation of documents with Wikipedia links (Mihalcea and Csomai, 2007) as detailed in Chapter 3. This corresponds to a first stage of topic identification, since it lists Wikipedia articles that can be linked to specific terms in a document. However, Wikify! is purely extractive, and thus limited to the text in the document under analysis and cannot identify important topics or articles in the graph unless they are explicitly mentioned in the input text.

8.3.5 Waikato Topic Indexing Experiments

Medelyan et al. (2008a) provide the dataset for the Computer Science topic identification experiment and their work is closely related to that task.

Medelyan and Witten use the keyphraseness measure for selecting initial articles, however they use a different approach for the WSD problem. Article titles and potential n-grams extracted from the text are case-folded and the parenthetical text used to wrap disambiguating class information are removed from articles. When matching disambiguation pages all articles in the first meaning are used. Those words with unambiguous mapping are used to disambiguate those with more than one mapping. To do this they define two measures, the *similarity* (Milne and Witten, 2008b) and *commonness* to create a score of the final match.

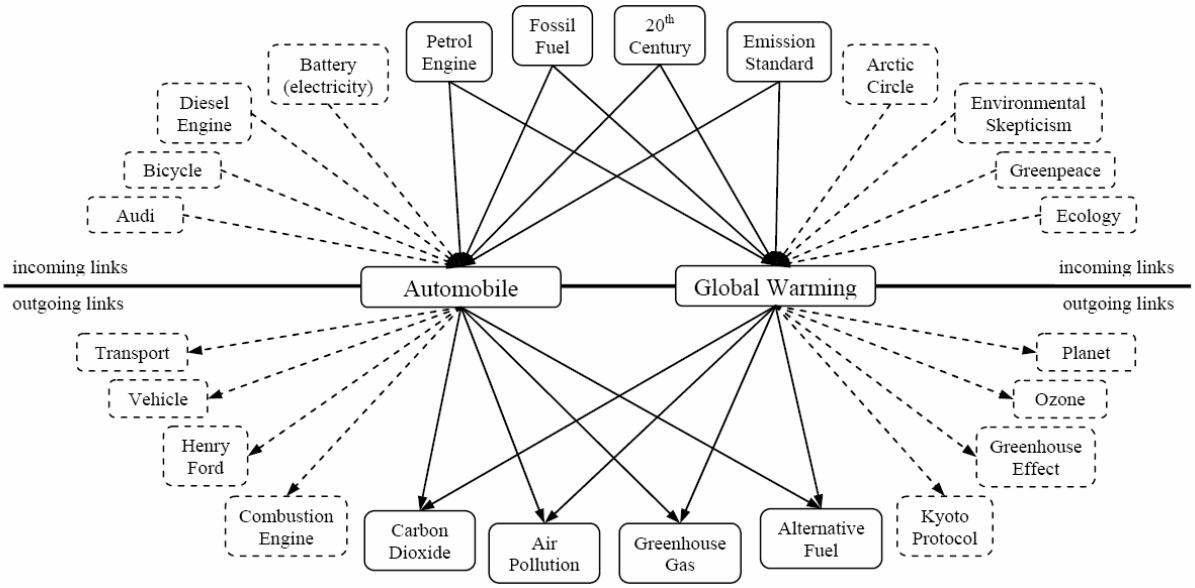

Figure 8.4 Example of Finding the Similarity Between "Automobile" and "Global Warming" Using the Common Wikipedia Links

Similarity is modeled after the Normalized Google Distance of (Cilibrasi and Vitanyi, 2007) and is defined between each possible candidate and all the unambiguous context articles:

$$similarity(x,y) = 1 - \frac{\max(\log(|X|), \log(|Y|)) - \log(|X \cap Y|)}{\log(N) - \min(\log|X|, \log|Y|)}$$

where X is the set of articles x links to, Y is the set y links to, and N is the total number of Wikipedia articles.

The commonness of an article T is derived for the n-gram anchor a as:

$$commonness(a,T) = P(T|a)$$

The final score of a mapping combines the similarity and commonness to find the average similarity of each article with the articles defined to be in the context set:

$$score(a,T) = \frac{\sum_{c \in C} similarity(T,c)}{|C|} * commoness(a,T)$$

where $c \in C$ are the context articles for T. The highest score is chosen as being the most compatible with the unambiguous interpretations. It is unclear what happens if set C is empty.

A training set of documents, with the most relevant Wikipedia articles assigned by humans defined for each, and a Naïve Bayes classifier is trained. Each candidate term is given a feature vector consisting of the TF-IDF, position of first occurrence, length, node degree, and total keyphraseness.

Table 8.1 System Consistency Compared with Human Indexers

METHOD	CONSISTENCY (%)		
	MIN	AVG	MAX
Human indexers	20.3	30.5	38.4
TF-IDF baseline	10.9	17.5	23.5
ML with 4 features	20.0	25.5	29.6
Total keyphraseness	22.5	27.5	32.1
ML with 5 features	24.5	30.5	36.1

8.3.6 Waikato Cyc Mapping

Medelyan and Legg (2008) report on the results of mapping Cyc terms to Wikipedia. Their mapping[34] generates up to 52k mappings. Using a process similar to the topic indexing process, they generate possible mappings between a Cyc term and a set of potential Wikipedia articles and then check the Cyc ontology to filter those mapping that for logical consistency. When compared against a manually prepared set of 9,333 mappings, their method showed a recall rate of 64.0% with a 93.9% precision. Some terms marked as errors were found to be using a more specific mapping than the gold standard annotator may have known at the time (e.g., #$Transport Aircraft → "Transport aircraft" instead of "Cargo aircraft"). Additions to this work can also be found in (Sarjant et al., 2009).

8.3.7 Green Measure

Ollivier and Senellart (2007) describe a method to determine related Wikipedia articles using a Markov chain derived value called the *Green measure*. Instead of the random walk model it uses an analogy related to electrostatic theory (and the potential created by a charge distribution) to Markov chains. The Green measure of a given node can be

[34] http://www.cs.waikato.ac.nz/~olena/cyc.html

thought of as the electric potential created at *x* by a unit charge placed at node *y*. Thus the unit charge placed at various nodes in the graph project a charge distribution across it, and is similar to the distribution that induced by the biased ranking proposed.

Differences exist between the PageRank-based methods tested as a baseline in their work and WikiRank proposed here, since WikiRank can use the content of the article, multiple starting points, and tighter control of the random jump probability via the *Bias* value. Thus the biased PageRank is closer to the Green measure than the unbiased PageRank. Also, instead of a unit charge value associated with each node, the *Bias* value provides finer control and the ability to introduce external knowledge sources.

8.3.8 Wikitology

Wikitology (Syed et al., 2008) and (Finin et al., 2009), is a knowledge base that combines information from Wikipedia, DBpedia and Freebase. The system includes both semi-structured and unstructured text from their component resources. While the KB contains additional semantic relation information, it does not use it for determining semantic relevance.

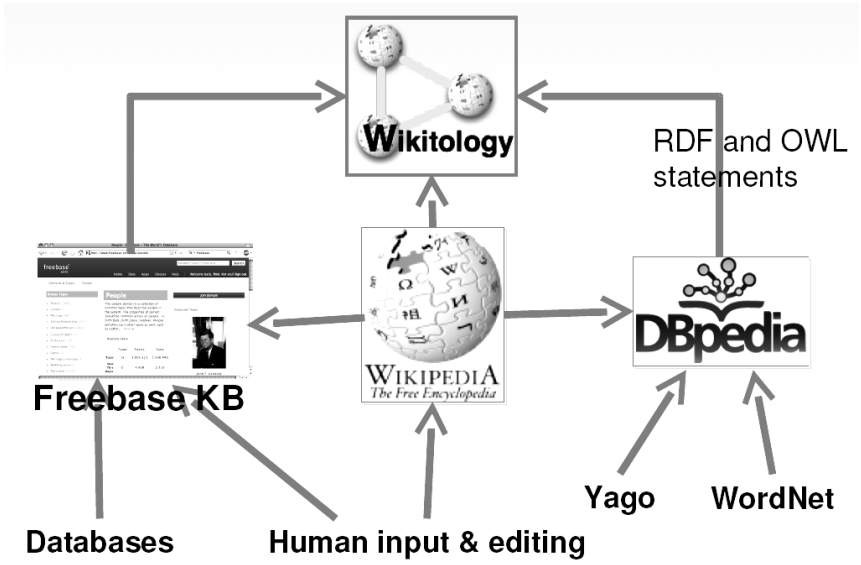

Figure 8.5 The Wikitology Construction Process

Using the knowledge base constructed, three methods of accessing it were tried. The first method involved using a set of documents that were related as a query to an information retrieval system containing Wikipedia articles. Using Cosine Similarity, the top N articles are returned, and for each article the associated Wikipedia categories are extracted, with each category receiving a score based on the number of occurrences and the similarity score associated with each member article. The second method involved the use of the Wikipedia category network to predict related concepts, using the categories listed by the first method and using them as the initial set of activated nodes for a spreading activation system applied to the category graph. After K iterations of spreading activation, the activation level of the categories nodes are presented as a ranked list. The third method used the initial set of matching articles as the start point for spreading activation over the article link graph. The links in the article link graph are filtered such that only articles with a Cosine Similarity above a threshold are visible to the spreading activation algorithm. That is, the spreading activation is applied over a subgraph with only relevant nodes as determined by Cosine Similarity as members.

Positive results were reported using the various methods for topic prediction. One hundred (100) random Wikipedia articles were selected as test cases and were removed from the information retrieval system and the link structure. The system was applied to the content of the test cases to find the related articles and categories for each using the original manual annotation as the gold standard.

For those test articles with an average similarity of at least 0.8, the F-measure was given as 100% for methods one and two, and 80% for method three. When the average similarity was at least 0.5, the F-measure was given as 77% for method one, 61% for method two and 67% for method three.

Wikitology has also been used to provide metadata for information retrieval systems. An accurate category prediction included any subsuming super class up to distance three.

While Wikitology is similar to WikiRank, there are several notable differences. Wikitology uses whole text Cosine Similarity, while WikiRank uses Wikify! to identify specific entities as start points. The spreading activation system is similar to PageRank; however, it only iterates a fixed number of steps, limiting the amount of information it utilizes, and only utilizes a feed-forward term. WikiRank, in contrast, can propagate through the whole graph until a well-defined convergence criteria is met. The *Bias* term in WikiRank allows inclusion of multiple start node functions, including Cosine Similarity, TF-IDF, ESA or LSA. Additionally, the WikiRank processor recognizes visibility, and thus can handle node visibility filtering.

8.3.9 Dataless Classification

Chang et al. (2008) describes a method known as *Dataless Classification* as a learning protocol, being able to produce classifications using no labeled or unlabeled data to train the classifier. The primary assumption is that if L is the label of some classifier C, then Wikipeidia articles that contain L should also contain terms relevant to C. Given a document d set of labels l_i the system picks the category i with the smallest L1 distance between l_i and d. They examined using bag-of-words and ESA as sources of vectors to compare. Note that the vector the document d is compared to is the one generated by just using the name of the desired category as input to ESA. This would be similar to utilizing the textual similarity metrics defined in Chapter 6, using the document and classifier name as inputs. One significant result of the work was that using ESA the

method achieved an accuracy of 94% for the 20 NewsGroup dataset[35] and 96% for the dataset constructed from Yahoo! Answers, which was seen as competitive with supervised algorithms using domain specific training data. They also examined (using semi-supervised learning) using unlabeled data and were able to match the results of a Naïve Bayes classifier trained on 100 labeled examples. The method accuracy was found to be sensitive to how "good" a class name was.

8.3.10 LarKC

The *Large Knowledge Collider* project or LarKC[36] attempts to develop a large infrastructure for reasoning over the semantic web. The system addresses fusing search, reasoning and limited rationality, in an attempt to reason in a Web-scale environment. The goal of the system is to provide an architecture that can handle incomplete and unsound knowledge sources while being capable of utilizing distributed computing, and can utilize multiple computation methods developed in different fields. If successful, the final product of LarKC will be a distributed reasoning platform that utilizes the entire semantic web as an in-depth encyclopedic knowledge source, updated in real-time. Of importance to this task is the ability to wisely allocate resources to retrieve assertions that contribute to a solution along with selection of relevant reasoning methods[37].

8.4 Non Distributional Similarity Measures

Numerous methods have been proposed to measure semantic relatedness. Some of the methods provide a measure for a single pair of terms, while others provide an estimate

---

[35] http://people.csail.mit.edu/jrennie/20Newsgroups/
[36] http://www.larkc.eu/
[37] Inference engine available at http://sourceforge.net/projects/larkc/

for extended passages. Here I will examine work related to returning similarity measures when used with encyclopedic knowledge sources not based purely on probability distribution.

8.4.1 Path-based Similarity Measures

One of the simplest ways to estimate the relatedness of two concepts is to use functions of the measured the distance between them in a semantic network. Given two concepts C_1 and C_2, one can find the shortest path between them and use the semantic distance between them with respect to that graph. Semantic similarity would then simply be the inverse of the semantic distance.

Resnik (1995) pointed out that the unmodified path length measure will be affected by differences in depth at different parts of a taxonomy where some sections may be deeper or more finer gained. Leacock and Chodorow (1998) proposed a normalized path length measure that takes into account the depth of the taxonomy:

$$lch(C_1, C_2) = -log \frac{length(C_1, C_2)}{2D}$$

where D is the maximum depth of the taxonomy (or, in the case of Wikipedia, the category subgraph) and length is the number of nodes in the shortest path that connects C_1 and C_2.

Wu and Palmer (1994) proposed another measure that accounts for the depth of the nodes directly based on the depth of their least common subsumer (*lcs*):

$$wup(C_1, C_2) = \frac{depth(lcs(C_1, C_2))}{depth(C_1) + depth(C_1)}$$

8.4.2 Informational Similarity Measures

As (Resnik, 1995) noted, different depths in a taxonomy may skew path-based measures. The solution proposed was not based on the path, but between concepts based on the information content determined from their probability of occurrence in a corpus. This is an attempt to model relatedness as "the extent to which they share information in common." First using the ISA taxonomy the least common subsume is found and the information content of the *subsumer* is used as a relatedness measure:

$$res(C_1, C_2) = -\log(p(lcs(C_1, C_2)))$$

where p is the probability of the class both concepts are members of. p is computed relative to the frequencies of the corpus:

$$p(c) = \frac{\sum_{w \in W(c)} count(w)}{N}$$

where N is the total number of words in the corpus and $W(c)$ is the set of words that are members of class c. As the subsumer class is higher in the hierarchy it subsumes more subclasses and instances and thus becomes more probable, until eventually the root of the taxonomy covers all other data and its probability becomes one. This means that two concepts that can only be related by use of the root node have a *res* value of 0 (-log(1)=0). The information content (*ic*) of a simple concept c is thus:

$$ic\ (c) = -\log p(c)$$

Lin (1998) extended Resnik's measure by adding a normalization factor based on the information content of the two concepts being compared:

$$Lin(C_1, C_2) = \frac{2 * ic\ (\ lcs(C_1, C_2))}{ic(C_1) + ic(C_2)}$$

Another information based similarity measure is the Jiang & Conrath (Jiang and Conrath, 1997) measure:

$$jnc(C_1, C_2) = \frac{1}{ic(C_1) + ic(C_2) - 2 * ic(lcs(C_1, C_2))}$$

These measures were tested in the Short Answer Evaluation system of (Mohler and Mihalcea, 2009).

CHAPTER 9

DISCUSSION AND CONCLUSIONS

> Wholly new forms of encyclopedias will appear, ready-made
> with a mesh of associative trails running through them,
> ready to be dropped into the memex and there amplified.
> – *Vannevar Bush, As We May Think, 1945*

9.1 Discussion of the Results

In this document I have described the extensions to a system called WikiRank for unsupervised ranking of encyclopedic knowledge which relies on a biased graph centrality algorithm applied on a highly interconnected encyclopedic graph. Overall, existing experiments demonstrate that this process can provide a relatively simple unsupervised method to recover useful human-like associations from knowledge resources such as Wikipedia, and these associations can be applied in meaningful contextual ways towards automatic topic identification and textual similarity tasks.

The research was conducted in several phases, with the results of each phase feeding and informing the next. During basic development the core idea and hypothesis were verified, along with connection to the Wikify! tagging engine. Experiments during this initial phase show that the integration of more encyclopedic knowledge consistently improved performance compared to baselines that did not use encyclopedic associations. This was verified using the topic identification task and showed human competitive results, indicating the algorithm was making meaningful estimations. These estimations were the input used by the similarity metrics and machine learning algorithms in the textual similarity phase of the work, resulting in a general method for constructing encyclopedic knowledge-based text classifiers. Returning to the direct

linkage method, I examined mapping two ontologies through a reference source. To meet the high throughput requirements, parallel processing was utilized in the overall framework of MapReduce. The results exhibited good accuracy, good precision, and the ranking of possibilities tended to reflect the topicality better than surface matchings alone. Finally, I examined recognizing matching ontological entries by reading their descriptions using the methods developed in textual similarity detection. This gave very encouraging results as the classifiers developed exhibited good accuracy, precision, recall, and F-measure.

At the core is an algorithm whose primary output is an estimate of the relative frequency each piece of encyclopedic knowledge is accessed given a focused reader with an infinite amount of time. WikiRank (despite the name) is not limited to just Wikipedia, but should be applicable to any large, highly interconnected knowledge source like those described in Chapter 3 and Chapter 8. That unbiased PageRank is currently used on the full web offers some indication of its potential scalability.

9.2 The Research Answers

Given the research conducted, we can re-examine my original questions and hypothesis:

- *How can one judge how WikiRank performs? What ways exist to verify each hypothesis? What natural basis for comparison (if any) exist?*

  WikiRank performs well when measured against the performance of humans, other systems, and assembled gold standards. When possible I used standard data and test sets along with their reported metrics. This included the Waikato dataset for topic identification, the Short Answer Grading dataset, and the Microsoft Research Paraphrase corpus. For initial

testing, Wikipedia itself provided an annotated corpus with human annotations indicating which categories and articles are appropriate for each text and only required temporary removal from the graph for testing. For the task of ontological similarity the fact that Cyc and YAGO-SUMO both have links to WordNet and Wikipedia allowed the existing WordNet path similarity metric of (Pedersen et al., 2004) to be used as a gold standard, and allowed the creation of the text corpus for the final paraphrases based ontological comparison. When possible either human or the best reported systems were used for comparison. When not available, the most frequent class and random baselines are also available.

- *How will the system react to the noise caused by ambiguity?*

  WikiRank is able to utilize the association between encyclopedia entries to "disambiguate in bulk." There is some sensitivity to the keyRatio used by the Wikify! front-end, with higher values resulting in higher noise. However, in tasks that require high keyRatios, the system is still able to perform well. Note that for paraphrase recognition tasks the tagger can be wrong—as long as it is consistently wrong and a consistent association pattern is generated.

- *Is WikiRank, when combined with sufficient knowledge, broad enough to perform different tasks?*

  WikiRank performed consistently above baselines in deriving the categories of Wikipedia articles when either human or Wikify! annotations were used. When applied to topic identification task of guessing the articles used by human teams to describe computer science texts

WikiRank shows a human competitive consistency of 34.5% compared to the human average of 30.5%.

WikiRank shows good performance at recognizing textual similarity. When applied to the Short Answer Grading task using the JSZKL metric with WikiRank provided a correlation of 0.4676 versus the ESA full Wikipedia score of 0.4681 and a CS focused LSA correlation of 0.4628. The MP5 method can combine metrics to achieve a correlation of 0.4895, which compares favorably with ESA Wikipedia full using relevancy feedback correlation of 0.4893 and other classifiers (see Table B.1) were able to beat the LSA Wikipedia focused on computer science using relevancy feedback correlation of 0.5099. On the Microsoft Research Paraphrase Corpus, WikiRank's performance exceeded all compared accuracy and precision values, and offered a competitive F-measure of 80.9 compared to Pointwise Mutual Information using Information Retrieval value of 81.0 and (Mihalcea et al., 2006) combined metric of 81.3.

On the task of matching ontological elements WikiRank showed positive results. For the task of direct comparison via Wikipedia linkage WikiRank allows the development of classifiers that show high accuracy (92.2%), good precision (81.5%), moderate recall (55.8%) and moderate F-measure (66.2%). Finally it showed very encouraging results for applying the paraphrase approach to recognizing potential ontological matches with the classifiers developed showing high accuracy (86.3%), high precision (94.2%), good recall (84.3%) and a high F-measure (89.0%).

Thus, it appears to be true that WikiRank can be used to perform multiple tasks. The applicability of WikiRank depends on the associations into and

within the encyclopedia used. All tasks performed were done using one version of Wikipedia.

- *Can a conceptually simple general purpose mechanism for applying encyclopedic knowledge to associational tasks be competitive?*

  It does appear to be possible. WikiRank uses biased PageRanking as a simple general-purpose associational inference engine, and systems using it were comparable to those more tailored to their task. When applied to the topic identification task of guessing the articles used by human teams to describe computer science texts WikiRank shows a human competitive consistency of 34.5% compared to the human average of 30.5% placing it between the 86% and 93% percentile of the human participants, with only two human teams out of fifteen teams doing better. In textual similarity tasks WikiRank is proven to provide sufficient information for machine learning algorithms to generate results that compare favorably to other reported systems and in some cases beating them. On the ontology tasks WikiRank allowed the creation of classifiers with high accuracy (86.3% to 92.2%), good precision (81.5% to 94.2%) and for the text-to-text method high F-measure (89.0%). In addition multiple learning methods showed high scores across multiple domains.

Each of the application-level hypotheses were confirmed along with the primary hypothesis and question as highlighted in Table 9.1. WikiRank indeed *is able to utilize the network of an encyclopedic knowledge source as a guide for what concepts are important and relevant to understanding a text.*

Table 9.1 Experimental Summary and Conclusions

PHASE	EXPERIMENT	PURPOSE	RESULT
Basic Development	Biased Ranking of Wikipedia Articles using original links	Evaluate initial biased PageRanking hypothesis in pure form free from Wikification noise	Verified ability to locate relevant terms exceeds baselines. Showed more links and knowledge consistently improves performance. Ranking Engine developed.
	Biased Ranking of Wikipedia Articles using Wikify!	Evaluate ability to operate using Wikify! as a tagger and initial biasing function	Verified ability to locate relevant terms using automatic Wikification exceeds baselines.
Topic Ranking	Topic Identification	Evaluate ability to return article and category rankings using text and compare with human and other systems	Verified ability to provide relevant articles on par with both human and supervised learning systems using an unsupervised algorithm.
Textual Similarity	Short Answer Grading (similarity)	Evaluate the performance of various similarity functions	Showed correlation of similarity functions matches that of existing systems on task.
	Short Answer Grading (WEKA)	Evaluate the ability of machine learning algorithms to utilize similarity of encyclopedic distributions	ML algorithms showed ability to combine similarity functions to exceed performance of pre-existing algorithms.
	MSRPC Paraphrase	Evaluate the performance to recognize text to text similarity	ML algorithms using similarity metrics show ability to match pre-existing algorithms.
Ontological Similarity	Pairwise Ontology Mapping with MapReduce	Evaluate ability to match terms from two different ontologies (Cyc and YAGO-SUMO)	Similarity metrics return semantically relevant results relative to text methods. ML develops classifiers with good accuracy and good precision, and moderate recall. Embedded system in MapReduce framework. Highlight need for better Wikification of ontology terms.
	Text-based Matching between Cyc and WordNet	Evaluate the paraphrase model for recognizing compatible definitions	Utilized paraphrase recognition for ontological purposes. ML systems develop classifiers with good accuracy, precision, recall and a max F-measure of 0.89. Automatic Wikification appears suitable for recognition.

9.3 Associational, Human, and Formal Processing

The system developed shows a form of associational intelligence, in that it is able to identify a subset of relevant terms from a large space that is similar to what a human would select given similar inputs. Its ability to do this is based on both the quantity and quality of the knowledge encoded in the encyclopedia being used. Because Wikipedia has several million articles, given most text the system can find enough relevant entry points in a text to start from. This allows the system to find both specific and general points to base its associations.

While humans may be good at tagging within their specialized field of interest, it becomes more difficult to expect them to select the best possible tag based on detailed understanding of an exponentially expanding ontology and lexicon. While the application of biased ranking of encyclopedic entries is simplistic and shallow, when applied to a constantly updated source like Wikipedia it shows a breadth of coverage that humans lack, and exhibits higher agreement with humans in the computer science annotation task than most humans do. A possible reason for this is the fact the system has immediate access to all the choices, and its method of processing possible associations takes into account the visitation pattern using all possible start points. This shallow total summation of all possible associations using the comprehensive listing appears to compensate for the naturally personalized and in-depth analysis used by humans.

The system is thus able to already know many points of interest, because human editors took the time and effort to make the entries. It is also able to use that knowledge in the form of links to relevant items. WikiRank is able to use the visitation simulation to generalize. Without this combined breath of knowledge of the specific, as well as access

to both generalizing and specializing associations, it is extremely doubtful any system would exhibit similar or superior performance.

One way to view the combined system is as a simple, massive expert system in human associations. It implements a single inference/association process over a large graph. The initial set of anchor-to-article linkages combined with the statistics used by Wikify! provides a form of *a priori* information on human preferences for word sense preferences. The links between articles provide associational information. When measured against knowledgeable humans functioning in a specific domain the system is able to show some level of competency. However, it is an expert system that currently does not utilize finer-grained relations between terms. This is an important area of future research.

An analogy can be drawn to the difference in humans and computers achieving equi-performance in chess (Hsu et al, 1995). Having an admittedly simpler representation of chess, machines achieve similar performance by applying the knowledge they have in the form of evaluating all possible extensions of a given state and extending the process forward in time. Humans, on the other hand, tend to use their initial understanding of the relevant components of the situation to filter the set of possible extensions. They attend to what they *think* their opponent would attend to. In the realm of ranking, the single node in the chess game tree is replaced with a superposition of all possible visitations N steps from the set of start points, based on the links between nodes in the encyclopedic graph. However, this graph was collectively constructed by tens of thousands of editors, and likely has links encoding associations that may not exist in the mental model of any particular human. Campbell (1999) describes a similar process in using human knowledge to guide Deep Blue. In addition, PageRanking to convergence simulates an infinite amount of link surfing. Utilizing Wikipedia provides the system with a form of "shallow omniscience," as it knows *something* about everything but not

very much (it exists as a concept and if it is relevant to the current situation or not). It compensates for its lack of knowledge by its breadth and the fact humans may miss something. The end result is that the ranking process is able to anticipate some entries that some humans will see as relevant, while a human performing the same task may not see the connection or even know that the entry existed as an option. When compared on the evaluation, the two strategies appeared to be evenly matched.

The system performs topic inference by association, in that if many elements closely associated with a topic are deemed relevant then the associated topic is also deemed relevant. This is the nature of the ranking process. In terms of formal inference it may be wrong. However, due to the additive nature of the system if more material is provided it should lead to generalization and having more false positives instead of false negatives. This is desirable for a system that might work as a front end to a formal reasoning system. To achieve higher levels of competency would require integration of domain specific ontologies and possibly a finer-grained set of relationships between nodes. Given a formal system like Cyc, which represents contexts as microtheories, the algorithm can be used to match new text to a textual microtheory description.

9.4 Future Work and Potential Application Areas

The research described in this dissertation is fairly general and can be extended in a number of ways with varying focus and complexity.

9.4.1 Folksonomy Tagging

One option provided might include the ability to determine the relevance of objects tagged with "folksonomies" dynamically constructed by humans with more formal ontologies by simply comparing descriptions. As such it provides a new "gist"-based approach to link implicit semantic methods with more explicit ones. By using text and

tags associated with an object, a system would be able to identify various media (photos, video, blog entries, course materials, virtual objects, etc.) as being semantically similar even across media types. Interestingly enough, the process that humans utilize in tagging folksonomies (jumping to pages with similar tags) is similar to the search process simulated by the biased ranking process. In the topic identification mode, the method can take objects identified by text and "tag" them. This suggests that a dynamic version of Wikipedia is possible whereby a stream of new objects are automatically linked to their most relevant encyclopedic entries.

9.4.2 Broaden Interfacing with Humans

The system can be embedded within a user feedback interface in various ways, both centrally and peripherally. It is fairly easy to construct an interface where a system presents two elements side by side for comparison by a human. The text-driven approach would allow domain and subject matter experts the ability to verify matchings, and provide visualization as to why a match was proposed, as the initial tagging, its links to encyclopedic entries and the ranking of those entries where they make up the dimensions of the semantic vector can be presented. This ability to offer a human-readable interpretation is useful for gaining acceptance and use with a wider audience. The higher potential F-measure would point to its usefulness as a filter for such an active supervised learning interface.

Cross-lingual processing is also possible. Given that Wikipedia is available in a myriad of languages and articles in each are cross linked to articles in others, it should be possible to perform these operations between texts in different languages (Hassan and Mihalcea, 2009). This could be useful for machine translation tasks since one can evaluate the degree that sentences in the two languages contain the same information,

with the added benefit that one could possibly identify any information that may be lost.

There are also potential uses as an assistant to encyclopedic editors. One potentially useful application is to take a description of interest (e.g., from background material) and provide a human reader ready access to relevant background or supporting material (or interested parties). The system can also aid Wikipedia editors by identifying redundancies.

Given the use of the human browsing metaphor, one can utilize methods that improve performance based on monitoring of human browsing behavior. For instance, DirectHit[38] was a search engine that iteratively modified the ranking of search results based on the actual articles accessed by users and the time spent at a linked site. There are numerous ways user collected information can be utilized and integrated into WikiRank. First, one can use the information to better initialize the initial bias values to more closely approximate the actual intent of users when that intent deviates from their explicit query. That is, a user may issue query Q which directly implies some page P1 by the Wikification process, but then they to eventually browse to page P2. In this case one could create a node to accept query Q and link it to P2 and use that when performing initialization. Another area is switch to a weighted version of PageRank and modify the weights between P1 and P2 or create them if non-existent. The use of actual browsing history can implement a form of reinforcement learning over the encyclopedic graph, increasing the association probability between nodes to more accurately reflect actual browsing behavior and other means of implicit feedback.

[38] DirectHit was acquired by Ask Jeeves. See http://en.wikipedia.org/wiki/Direct_Hit_Technologies

9.4.3 Global Knowledge Map

With the advent of large processors, it becomes possible to store a global knowledge map in its entirety along with application-specific ontologies. WikiRank can be utilized as a navigation system for the global map of knowledge—the system can help identify relevant information sources if they are associated with the encyclopedic entries, and can be used to focus any associated reasoning resources. This would potentially allow a system to "know what it doesn't know" and—if links are provided—know where to find it. Also, by being able to utilize text and recognize similarity the system could be used to map input text from various sources into a common reference source. This was demonstrated with paraphrase detection, answer recognition and ontology matching.

Another use is as an estimation function for a classic state space search process. Here an input text would be systematically compared to reference text provided by an ontology, with the similarity functions providing the heuristic estimate values. This would make the search of the global knowledge map explicit.

9.4.4 Integration and Interface with Other Systems and Technologies

An important characteristic of WikiRank and its overall structure is that it could lend itself to easy integration with other technologies. One factor to consider is that currently the ranking only utilizes basically untyped linkages between encyclopedic entries. In the case of Wikipedia, more entries are being directly mapped to formal ontologies. Where links between two entries can be mapped in the ontology, one can potentially identify the relationship using either a directly connected or inferred relationship, or by enumeration of the possible relationships based on link type and light, focused parsing. Given the positive results with unweighted biased PageRank, potentially weighted biased PageRank could be explored with weights assigned based

on relationship. Another idea would be to adjust the weightings based on backward propagation or other machine learning algorithms, and similarly adjusting the weights used for the set construction used by the NGD, Jaccard and DICE metrics.

Cross ontology recognition could lead to cross domain recognition of analogous concepts. Mascardi et al. (2009b) covers a set of algorithms that utilize upper ontologies to find links between terms in different domains ontologies, and evaluated the use of OpenCyc, SUMO-OWL and DOLCE, and found that recall can be significantly improved without degrading precision. They also showed that better performance occurred when OpenCyc and SUMO-OWL are combined. As shown in the chapter on matching ontological terms (Chapter 7), WikiRank can aid in term matching. Another use of the textual method is as a checker for other ontology matching systems. In this role WikiRank can offer an estimate of the likelihood that a proposed mapping is correct.

More efficient means of ontology matching other than the full pairwise testing process that used MapReduce exist. Melnik et al. (2002) described the method of similarity flooding, whereby nodes representing a pairwise mapping between elements of two ontologies are similar if their neighbors in mapping space are similar. In that application, two database schemas are converted to graph structures, initial similarity scores are given to nodes containing a potential matching pair and a fixed point computation similar to ranking computation is used to propagate similarity values from closely matching nodes to structurally linked neighbors. This use of flooding based on potential pairings may offer both enhancement in accuracy and reduction in processing requirements, if the graph of map pairs can be generated inexpensively and a subset of similarity scores would be sufficient for initial processing. Embedded in a larger framework, a system could select for detailed similarity comparisons with WikiRank those nodes that offer the highest information gain for the entire graph.

Another area of exploration would be to utilize other methods to initialize the *Bias* values for WikiRank. In this work the input document is seen as uniform source of weighted links. A better approximation would involve constructing and integrating the graph implied by the text into the whole ranking process. TextRank (Mihalcea et al., 2004) could provide this functionality. WikiWalk (Yeh et al., 2009) utilized ESA (Gabrilovich and Markovitch, 2006) as an analogous bias initialization function. The WSD method outlined in (Agirre and Sora, 2009) would also make an interesting bias initialization processor, especially if embedded in the MapReduce framework to process the required WSD ranking processes in parallel.

While natural language does not consistently or unambiguously name objects, WikiRank is able to use the weighted summation of associations to recognize similar concept descriptions. While formal logical methods are unambiguous, WikiRank is able to operate in spite of the ambiguities. This ability could find use in recognition of web service interfaces, described by a diverse group of authors. Akkiraju et al. (2006) describes using semantic matching of web service interface descriptions to perform composition planning and (Bouillet et al., 2008) describes using folksonomy tag matching semantic web service composition. WikiRank could aid in the ability to recognize semantically equivalent service descriptions.

Possibilities exist to reduce the resource requirements of the algorithm. By monitoring the variance of values assigned to nodes in the graph, the most informative nodes can be identified for a given input corpus. One can then use connectivity analysis to reduce the graph, allowing a domain specific subgraph to be extracted. Evaluation would determine if this subgraph would provide similar performance to that of the full graph for specialized applications. In addition, the performance relative to the full graph could be monitored and an appropriate selection mechanism can be employed. Also, while not utilized in the systems examined, the output of the processing can be utilized

by traditional inverted index text retrieval systems. This could either make retrieval more intelligent through relevant associations, or be used by the algorithms internally to improve scalability.

WikiRank's ability to recognize paraphrases can augment answer extraction algorithms, which often grade the quality of an answer by measuring the support from multiple sources. The system allows variation in the expression of supporting information. In a generate and test format, a system could generate plausible answers and utilize the ability to recognize paraphrases in supporting material to grade the possible conjectures. The inverse of the ability to detect potentially redundant knowledge is the ability to detect novelty. If monitoring a data steam, WikiRank could potentially generate alerts when new input does not match anything else in the stream, or if it matches a predefined standing query. WikiRank could also be explored for use in entailment detection as some of the information-based similarity metrics examined are asymmetric, measuring the ability to generate or infer a target distribution given a source.

Other mergers with knowledge-based systems and methods are possible. An adaptation of the dataless classification method defined by (Chang et al., 2008) could also be explored since WikiRank can produce a compatible semantic vector and has explored using semantic metrics in addition to L1 covered by their work. Similarly, WikiRank can be utilized with the WSD concepts provided by (Agirre and Sora, 2009). And, Wikipedia is not the only encyclopedic knowledge source—other methods of compiling encyclopedic knowledge are possible. MindNet, ConceptNet, and any other ontologies that have a graph structure can be utilized. Systems like TextRunner can read texts and produce similar large scale graphs to which WikiRank could be applied as well.

In Case-Based Reasoning and Memory-Based Reasoning, one indexes similar problems in the past and adapts them to the current situation. WikiRank would allow a system to quickly note the similarities and differences between the current situation and past cases in a flexible manner. Additional knowledge about similarity and difference could be represented and explored explicitly. And finally, the current system hard codes the similarity functions but their use is invoked by a LISP interpreter, which could be modified to perform the basic operations of each function, and thus allow the space of possible similarity functions to be explored.

9.4.5 External Technology Advances

Three technological factors allowed the effective exploration of the biased ranking over an encyclopedia. First, the encyclopedic graph fits in RAM and thus can be efficiently ranked with the current C/C++ based implementation in several seconds. Additional speed improvements can be achieved by the use of fine-grained multi-processors and large memories such as those found in commodity GPU's (Fatahalian and Houston, 2008). Initial tests show that an order of magnitude improvement can be obtained by porting to such platforms, allowing sub-second ranking.

The second factor was the ability to utilize very large scale parallelism in the Hadoop MapReduce framework for the ontology comparison. Without it, additional work would have been necessary for vector indexing, instead of the direct comparison.

Lastly, the availability of Wikipedia, providing an encyclopedia with all its links and information to rank, was critical to this project. While additional encyclopedic sources are available, Wikipedia provides one for free that is constantly updated by a set of motivated editors from a broad range of backgrounds. The trends in all three factors are towards improvement. This means broader coverage knowledge available faster.

Hopefully the end result will be the ability to embed the ability to recognize what a human might find relevant or not in applications, and apply the technique to the web.

9.5 Contributions of this Work

The extensive evaluations performed in this dissertation demonstrated that WikiRank is able to employ the latent semantics in an encyclopedic knowledge source to suggest relevant topics and perform similarity discriminations. Biased ranking provides a controllable spreading activation system that can use the tagging of text for initial entry points to identify additional information that would naturally be entailed. Evaluations show that WikiRank is able to perform competitively in the areas of topic identification and paraphrase detection.

My research verified the "knowledge access heuristic": that knowing an estimate of the amount of time spent examining an object can be used as a proxy for its relative importance in a given context. The amount of knowledge access is estimated by a recursive simulation of visitation frequency. Not only does it offer suggestions for prioritizing processing, characterize contexts by identification of relevant topics, and give estimation of contextual similarity, uses can also be found for the information it provides on the allocation of processing resources given specific entries and what can be inferred by specific knowledge access differences. The only constraint on using the information provided by the system is the encyclopedia being used, the information associated with each entry (such as procedural knowledge, declarative knowledge and links to external resources), and the eventual interpretation process to be applied. Once fully developed, it is hoped that it will find usefulness in future tasks.

This research led to the development of a general framework for paraphrase detection. The method of operation provided has the ability to work with recognition on a wide

range of subjects. The degree to which it operates depends on the wealth of associations provided by the encyclopedic reference source.

The framework developed provides a form of knowledge-based pattern-matching by association. The product that is the result of the association process used is an estimation of the value of various concepts in the current context, which can feed into more formal processes such as word sense disambiguation, or prioritizing concepts to consider in inference processes. The ability to bring more knowledge to bear on a problem can aid in disambiguation, by providing estimation of *a priori* relevancy to parsing and formal reasoning methods.

The extensive examination of the properties of WikiRank demonstrates that it can be integrated with other components. WikiRank has the ability to utilize knowledge encoded in encyclopedic works like Wikipedia for various tasks using the associational structure built up by their editors, and embed it inside larger systems. Wikify! + WikiRank has the ability to process text of varying lengths and identify relevant concepts. The system can provide uniform folksonomy tagging of text associated with data objects (text, audio, video, records, etc.) by using its ability to suggest article and category tags, and linking them to similar objects. Such initial tagging can inform object parsing by more formal systems.

The research addresses the area of broad coverage knowledge bases. In the past, knowledge systems (for the most part) tended to be developed from the specific towards the general. We are starting to find ourselves in the inverse situation, having a broad map of knowledge which we can now increase the resolution as needed in specific areas or find common points such that we can merge existing higher resolution maps (domain ontologies) into the global map.

The core algorithm used to implement biased ranking is simple and uniform, and its operation and termination is dependent on the knowledge graph it processes. In experiments the primary parameter modified for a given task was the keyRatio, or what percentage of available links were actually used. While certain operations are serial, the core algorithm can exploit parallelism in a straightforward manner in both a fine and coarse grain manner, and whole systems can be embedded in a larger framework like MapReduce. Optimization of the computation of PageRank is an area of active research and the system can exploit any development in that area.

Finally, WikiRank provides an existence proof that verifies the primary hypothesis that it is indeed possible to utilize encyclopedic knowledge as a guide for what concepts are important and relevant to understanding a text. Hopefully in the future similar methods will function well using web and semantic web for knowledge bases when processing objects and contexts other than text.

9.6 Conclusions

A large amount of human knowledge covering all of recorded history is represented as natural language text. Humans have used it to represent and communicate a wide range of topics. As evidenced by the creation of Wikipedia (and all of the libraries of earth), collection of knowledge by humans using writing in natural language is relatively easy. Alternatively, collection of equivalent knowledge formally by unambiguous manual encoding has to date been comparatively expensive. The methods outlined here attempt to handle natural language using association to simulate connotations rather than formal definition and deduction from denotation. These experiments were designed in some way to examine and bridge the gap between the two methods, by coupling an initial tagging process (capable of the denotational) with an associative one (simulating connotation). It is this connotational process that allows

natural language to compress and transmit information, albeit in a sometimes lossy manner. It is also the process that the method described in this work tries to simply emulate.

The overall results supports the hypothesis that biased PageRanking coupled with an encyclopedic knowledge source can provide a useful framework for determining the relative importance of the encyclopedic entries in context. When combined with a term identification front end and similarity metrics, the resulting system is competitive with other systems, and where information was available, comparable to human levels of performance. The combination semantic and textual similarity metrics allow existing machine learning methods to flexibly recognize natural language similarity despite surface variations. In short, WikiRank provides a *general-purpose mechanism* where one can straightforwardly add additional knowledge to that can identify relevant concepts by association, and utilize that for enumeration and comparison on the semantic level.

While much still remains to be developed, the system appears to offer useful insights. WikiRank offers conceptual simplicity, the ability to increase performance with advances in hardware, software, and available knowledge, and can exploit (as well as aid) access to the ever increasing web of links between objects. As alluded to in the title, using the links *between* everything WikiRank can assign a value *to* everything based on the current value *of* everything. It remains an area of exploration to see how this ability can be effectively harnessed by both humans and systems with different inference capabilities.

APPENDIX A

MACHINE LEARNING PROCESSING SUMMARY

A.1 SubProject: Text-Text Paraphrase

Description: Microsoft Research Paraphrase Corpus (MSRPC). Wikified, then similarity measured between the two rank vectors. Used in Chapter 6.

keyRatio: 0.2

Train Num Instances:	4075
Test Num Instances:	1722

Analog Grading:

ANALOG METHOD	CORRELATION
LeastMeansSQ	0
LinearRegression	0.4282
Multilayer Perceptron	0.3847
PaceRegression	0.4283
RBFNetwork	0.1989
SimpleLinearRegression	0.4232
Isotonic regression	0.4322
Gaussian Processes	N/A
SMOreg	0.4182
SVMreg	0.4195
Decision Stump	0.3457
MP5	0.4282
REPTree	0.4246
Conjunctive Rule	0.3457
DecisionTable	0.4272
M5Rules	0.4282
ZeroR	0.0000

Binary Classification:

	TRAINING	TEST
Class0	1323	577
Class1	2752	1145

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BayesNet	64.0534	0.775	0.647	0.705
NaiveBayes	66.3182	0.768	0.707	0.736
Logistic	72.4739	0.762	0.852	0.763
Multilayer Perceptron	72.2416	0.764	0.844	0.802
SimpleLogistic	72.8223	0.763	0.857	0.807
SMO	73.1127	0.762	0.867	0.811
Voted Perceptron	59.5819	0.68	0.741	0.709
ADTree	72.3577	0.769	0.835	0.801
BFTree	72.1835	0.745	0.884	0.809
Decision Stump	66.4925	0.665	1.000	0.799
J48	72.1835	0.745	0.884	0.809
J48Graft	72.1254	0.745	0.884	0.808
LMT	72.8223	0.763	0.857	0.807
NBTree	71.3124	0.741	0.873	0.802
Random Forest	69.1638	0.769	0.766	0.768
Random Tree	62.892	0.72	0.724	0.722
REPTree	72.2997	0.753	0.868	0.806
SimpleCart	72.1835	0.745	0.884	0.809
ConjunctiveRule	66.4925	0.665	1.000	0.799
Decision Table	69.6283	0.758	0.798	0.778
JRIP	72.3577	0.769	0.835	0.801
NNge	68.9895	0.735	0.835	0.782
OneR	69.7445	0.732	0.859	0.791
PART	72.1835	0.745	0.884	0.809
Ridor	69.5703	0.697	0.961	0.808
ZEROR	66.4925	0.665	1.000	0.799
RBFNetwork	66.4925	0.665	1.000	0.799

A.2 SubProject: Ontology Vector Classification

Description: Utilize Pairwise similarity comparisons between known good links between Cyc and Wikipedia. Links were found in the respective ontologies to Wikipedia, and ranking was performed to create ranking vectors. Links where both Cyc and YAGO both point to WordNet are used as a gold standard and WordNet::Similarity::Path is used as an evaluation in Chapter 7.

keyRatio: N/A

Train Num Instances:	2702
Test Num Instances:	1314

Analog Grading:

ANALOG METHOD	CORRELATION
LeastMeansSQ	0.2629
LinearRegression	0.7685
Multilayer Perceptron	0.7205
PaceRegression	0.7689
RBFNetwork	0.7033
SimpleLinearRegression	0.7574
Isotonic regression	0.7673
Gaussian Processes	0.7878
SMOreg	0.7677
SVMreg	0.7677
Decision Stump	0.7206
MP5	0.7813
REPTree	0.7722
Conjunctive Rule	0.7206
DecisionTable	0.7753
M5Rules	0.7811
ZeroR	0.1707

Binary Classification:

	TRAINING	TEST
Class0	2358	1133
Class1	344	181

Note:

- Class0 consists of instances where WordNet::Similarity::Path value <0.2
- Class0 1 consists of instances where WordNet::Similarity::Path value >=0.2
- Most frequent class is Class 0 but Precision, Recall and F-measure are for Class 1

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BayesNet	87.74	0.544	*0.685*	0.606
NaiveBayes	89.11	0.587	0.646	0.621
Logistic	91.78	0.876	0.47	0.612
Multilayer Perceptron	91.7	0.833	0.497	0.623
SimpleLogistic	91.55	0.88	0.448	0.593
SMO	90.48	*0.938*	0.331	0.49
Voted Perceptron	91.85	0.885	0.47	0.614
ADTree	92.16	0.815	0.558	*0.662*
BFTree	*92.2374*	0.883	0.503	0.641
Decision Stump	92.16	0.815	0.558	*0.662*
J48	92.16	0.815	0.558	0.662
J48Graft	92.08	0.813	0.552	0.653
LMT	91.5525	0.88	0.448	0.593
NBTree	92.16	0.882	0.497	0.636
Random Forest	91.93	0.844	0.508	0.634
Random Tree	88.965	0.618	0.519	0.565
REPTree	92.0852	0.853	0.514	0.641
SimpleCart	*92.2374*	0.876	0.508	0.643
ConjunctiveRule	92.16	0.815	0.558	*0.662*
Decision Table	92	0.88	0.486	0.626
JRIP	92	0.873	0.492	0.629
NNge	88.58	0.595	0.536	0.564
OneR	91.933	0.878	0.481	0.621
PART	91.7	0.891	0.453	0.601
Ridor	91.32	0.914	0.409	0.565
ZEROR	86.22	0.000	0.000	0.000

A.3 SubProject: CYC - WordNet Paraphrases

Description: Cyc comments and WordNet glosses provided text examples. Cyc's links to WordNet are used to identify gold translations. WordNet::Similarity:Path is used to provide grading. Used in Chapter 7.

keyRatio: 0.2

Train Num Instances:	375
Test Num Instances:	176

Analog Grading:

ANALOG METHOD	CORRELATION
LeastMeansSQ	0.6858
LinearRegression	0.6854
Multilayer Perceptron	0.5151
PaceRegression	0.6788
RBFNetwork	0.5049
SimpleLinearRegression	0.6434
Isotonic regression	0.6653
Gaussian Processes	0.6844
SMOreg	0.6767
SVMreg	0.6768
Decision Stump	0.5200
MP5	0.6854
REPTree	0.6252
Conjunctive Rule	0.5700
DecisionTable	0.6348
M5Rules	0.6854
ZeroR	0.2506

Binary Classification:

	TRAINING	TEST
Class0	140	61
Class1	235	115

Note:

- Class0 consists of instances where WordNet::Similarity::Path value <0.2
- Class1 consists of instances where WordNet::Similarity::Path value >=0.2

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BayesNet	78.4091	0.953	0.7004	0.81
NaiveBayes	77.2727	0.952	0.687	0.798
Logistic	81.25	0.918	0.783	0.845
Multilayer Perceptron	81.8182	0.928	0.783	0.849
SimpleLogistic	81.25	0.927	0.774	0.844
SMO	78.4091	0.953	0.704	0.81
Voted Perceptron	57.3864	0.637	0.809	0.713
ADTree	81.8182	0.919	0.791	0.85
BFTree	*86.3636*	0.942	0.843	*0.89*
Decision Stump	71.022	*0.971*	0.574	0.721
J48	78.4091	0.953	0.704	0.81
J48Graft	78.4091	0.953	0.704	0.81
LMT	80.11	0.935	0.748	0.831
NBTree	82.95	0.938	0.791	0.858
Random Forest	80.68	0.935	0.757	0.837
Random Tree	74.43	0.865	0.722	0.787
REPTree	80.1136	0.864	0.826	0.844
SimpleCart	80.6818	0.955	0.739	0.833
ConjunctiveRule	71.5909	*0.971*	0.583	0.728
Decision Table	78.4091	0.933	0.722	0.814
JRIP	81.25	0.936	0.765	0.842
NNge	85.7655	0.917	*0.861*	0.888
OneR	79.5455	0.944	0.73	0.824
PART	74.7045	0.963	0.67	0.79
Ridor	80.6818	0.966	0.73	0.832
ZEROR	65.34	0.653	1	0.79

APPENDIX B

ADDITIONAL CLASSIFIER PERFORMANCE DATA

Table B.1 Correlation of Regression Classifiers on Short Answer Grading in Chapter 6

ANALOG METHOD	CORRELATION WITH -COSINE = -0.001	CORRELATION WITH -COSINE = -9999	DIFF
LeastMeansSQ	0.3032	-0.0123	-0.3155
LinearRegression	0.3902	0.3005	-0.0897
MultilayerPrecptron	0.4323	0.4051	-0.0272
PaceRegression	0.4018	0.4244	0.0226
RBFNetwork	0.3431	0.3424	-0.0007
SimpleLinearRegression	0.3486	0.3345	-0.0141
Isotonic regression	0.2305	0.2401	0.0096
Gaussian Processes	0.4869	0.4606	-0.0263
SMOreg	0.3619	0.3130	-0.0489
SVMreg	0.3604	0.3126	-0.0478
Decision Stump	0.3539	0.3630	0.0091
M5P	0.4324	0.4895	0.0571
REPTree	0.3444	0.3902	0.0458
Conjunctive Rule	0.2957	0.2850	-0.0107
DecisionTable	0.3999	0.3902	-0.0097
M5Rules	0.4250	0.4874	0.0624
ZeroR	-0.1341	-0.1341	0.0000
PLSClassifier	0.3001	*0.5204*	0.2203
Additive Regression	0.4974	*0.5416*	0.0442

Note: Cosine computation can result in undefined value which was truncated to either -999 or -0.0001. The table provides both and shows the per-classifier performance difference.

Table B.2 Correlation without TRI or LEV in Feature Set for Paraphrase Classifiers in Chapter 6

ANALOG METHOD	CORRELATION
LeastMeansSQ	0.0000
LinearRegression	0.2171
Multilayer Perceptron	0.1291
PaceRegression	0.2144
RBFNetwork	0.1841
SimpleLinearRegression	0.2144
Isotonic regression	0.2225
Gaussian Processes	N/A
SMOreg	0.1314
SVMreg	-0.06665
Decision Stump	0.196
MP5	0.2171
REPTree	0.1583
Conjunctive Rule	0.1966
DecisionTable	0.2050
M5Rules	0.2171
ZeroR	0.0000

Table B.3 Binary Classifier Performance without TRI or LEV in Feature Set in Chapter 6

BINARY METHOD	ACCURACY	PRECISION	RECALL	F-MEASURE
BayesNet	60.163	0.758	0.590	0.663
NaiveBayes	63.705	0.731	0.718	0.725
Logistic	66.493	0.676	0.951	0.791
Multilayer Perceptron	67.364	0.678	0.969	0.798
SimpleLogistic	66.609	0.676	0.956	0.792
SMO	66.493	0.665	1.000	0.799
Voted Perceptron	57.549	0.661	0.741	0.699
ADTree	66.957	0.676	0.966	0.795
BFTree	67.420	0.678	0.974	0.799
Decision Stump	66.493	0.667	1.000	0.799
J48	67.015	0.680	0.951	0.793
J48Graft	67.015	0.680	0.951	0.793
LMT	68.131	0.680	0.955	0.794
NBTree	66.957	0.692	0.907	0.785
Random Forest	62.718	0.700	0.769	0.733
Random Tree	62.311	0.708	0.736	0.722
REPTree	67.654	0.688	0.941	0.795
SimpleCart	67.073	0.681	0.949	0.793
ConjunctiveRule	66.493	0.665	1.000	0.799
Decision Table	66.899	0.679	0.953	0.793
JRIP	66.144	0.688	0.897	0.779
NNge	59.292	0.684	0.721	0.702
OneR	64.053	0.689	0.838	0.756
PART	67.073	0.678	0.960	0.795
Ridor	66.957	0.670	0.990	0.799
ZEROR	66.493	0.665	1.000	0.799
RBFNetwork	66.493	0.665	1.000	0.799

REFERENCES

E. Agirre and A. Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, March 30 – April 3, 2009*, pages 33–41, Athens, Greece, (DBL, 2009).

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June. Association for Computational Linguistics.

R. Akkiraju, B. Srivastava, A. Ivan, R. Goodwin, and T. Syeda-Mahmood. 2006. SEMAPLAN: Combining planning with semantic matching to achieve web service composition. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, pages 37–44, Washington, DC, USA. IEEE Computer Society.

S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. 2007. DBpedia: A nucleus for a web of open data. Aberer, K. et al. (eds.) *ISWC/ASWC 2007, LNCS 4825*, pages 722–35, Berlin Heidelberg, SpringerVerlag.

S. Banerjee and T. Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proc. of IJCAI-2003*, pages 805-810.

Z. Bodo, Z. Minier, and L. Csato. 2007. Text categorization experiments using Wikipedia. In *Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques*, pages 66-72, Cluj-Napoca (Romania).

K. Bollacker, R. Cook, and P. Tufts. 2007. Freebase: A shared database of structured general human knowledge, In *Proceedings of the National Conference on Artificial Intelligence (Volume 2)*, pages 1962-1963, AAAI Press, MIT Press, July 2007.

E. Bouillet, M. Feblowitz, H. Feng, Z. Liu, A. Ranganathan, and A. Riabov. 2008. A folksonomy-based model of web services for discovery and automatic composition. In *IEEE International Conference on Services Computing, 2008. SCC '08*, volume 1, pages 389–396, Honolulu, Hawaii, USA, July. IEEE Computer Society.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Trees.* Wadsworth International Group, Belmont, CA, USA.

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems, 30(1–7).*

E. Buyko, J. Wermter, M. Poprat, and U. Hahn. 2006. Automatically Adapting an NLP Core Engine to the Biology Domain. In *Proceedings of the ISMB 2006 Joint Linking Literature, Information and Knowledge for Biology and the 9th Bio-Ontologies Meeting*, pages 65-68, Fortaleza, Brazil, August.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics, 32(1):13–47.*

M. Campbell. 1999. Knowledge Discovery in Deep Blue. *Communications of the ACM*, 42(11): 65-67.

M. Chang, L. Ratinov, D. Roth, and V. Srikumar. 2008. Importance of Semantic Represenation: Dataless Classification. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 830-835, July.

R. Cilibrasi and P. Vitanyi. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering* 19(3):370-383.

W. Cohen, P. Ravikumar, and S. Fienberg. 2003a. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the Workshop on Data Cleaning and Object Consolidation at the International Conference on Knowledge Discovery and Data Mining (KDD),* pages 39-48, Washington,DC, August.

W. Cohen, P. Ravikumar, and S. Fienberg. 2003b. A comparison of string metrics for matching names and records. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (August 2003),* pages 73–78.

K. Coursey. 2004. Living in CyN: Mating AIML and Cyc together with Program N, Daxtron Laboratories. http://www.cyc.com/doc/white_papers/Cyn_description.pdf

K. Coursey. 2007. WAC: Weka and Cyc, teaching Cyc to learn through self-recursive data mining. In *Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, Honolulu, HI, Jan. Intelligent User Interfaces Conference.

K. Coursey and R. Mihalcea. 2009a. Topic Identification Using Wikipedia Graph Centrality. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational*

Linguistics, Companion Volume: Short Papers, pages 117–120, Boulder, Colorado, June. Association for Computational Linguistics.

K. Coursey, R. Mihalcea, and W. Moen. 2009. Using encyclopedic knowledge for automatic topic identification. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 210–218, Boulder, Colorado, June. Association for Computational Linguistics.

G. de Melo, F. Suchanek, and A. Pease. 2008a. Integrating YAGO into the Suggested Upper Merged Ontology. In *Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence*, 1:190–193.

G. de Melo, F. Suchanek, and A. Pease. 2008b. Integrating YAGO into the Suggested Upper Merged Ontology. Research Report MPI-I-2008-5- 003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August.

J. Dean and S. Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI '04: Proceedings of the 6th symposium on Operating Systems Design & Implementation*, pages 137-150, San Francisco, CA, USA. USENIX Association.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman.1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

W. Dolan, C. Quirk, and C. Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 350-356, Geneva, Switzerland.

O. Etzioni, M. Banko, S. Soderland, and D. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

K. Fatahalian and M. Houston. 2008. A closer look at GPUs. *Communications of the ACM*, 51(10):50–57.

T. Finin, Z. Syed, J. Mayfield, P. McNamee, and C. Piatko. 2009. Using Wikitology for Cross-Document Entity Coreference Resolution. In *Proceedings of the AAAI Spring Symposium on Learning by Reading and Learning to Read*. pages 29-35, AAAI Press, March.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

E. Gabrilovich and S. Markovitch. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Boston.

N. Guarino. 1998. Some ontological principles for designing upper level lexical resources. In *Proceedings of First International Conference on Language Resources and Evaluation*, pages 527–534, Granada, Spain. ELRA – European Language Resources Association.

R. Guha. 1992. Contexts: A formalization and some applications. Ph.D. thesis, Stanford University, Stanford, CA, USA.

E. Han and G. Karypis. 2000. Centroid-based document classification: Analysis and experimental results. In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 424–431, London, UK. Springer-Verlag.

S. Hassan and R. Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1201, Singapore, August. Association for Computational Linguistics.

C. Havasi, R. Speer, and J. Alonso. 2007. ConceptNet 3: A flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September.

T. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the Eleventh International World Wide Web Conference*, May.

C. Hayes and P. Avesani. 2007. Using Blog Tags to Identify Topic Authorities. In *1st International Conference on Weblogs and Social Media, (ICWSM 2007)*, Boulder, Colorado, March.

F. Hsu, M. Campbell, and A. Hoane. 1995. Deep Blue system overview. In *ICS '95: Proceedings of the 9th International Conference on Supercomputing*, pages 240–244, New York, NY, USA. ACM.

A. Huang, D. Milne, E. Frank, and I. Witten. 2009. Clustering Documents Using a Wikipedia-based Concept Representation. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009*, Bangkok, Thailand, April 27-30, 2009, Proceedings, volume 5476 of *Lecture Notes in Computer Science*, pages 628–636, Springer. http://www.cs.waikato.ac.nz/~ml/publications/2009/pakdd09_similarity_cr.pdf

B. Huberman and F. Wu. 2007. The economics of attention: maximizing user value in information-rich environments. In *ADKDD '07: Proceedings of the 1st International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 16–20, New York, NY, USA. ACM.

T. Hughes and D. Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2007)*, pages 581-589, Prague, Czech Republic, June.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pages 19-33, Tapei, Taiwan.

P. Kanerva. 1988. *Sparse Distributed Memory*. MIT Press, Cambridge, MA, USA.

R. Kohavi. 1996. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 202-207, Cambridge, MA 02142. The AAAI Press/The MIT Press.

C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, Chapter 11, pages 265–283, Cambridge, MA. MIT Press.

C. Leacock, M. Chodorow, and G. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

C. Leacock and M. Chodorow. 2003. C-rater: Automated Scoring of Short-Answer Questions. *Computers and the Humanities*, 37(4):389–405

M. Lee, B. Pincombe, and M. Welsh. 2005. An Empirical Evaluation of Models of Text Document Similarity. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society (2005)*, pages 1254–1259, Mahwah, NJ. Erlbaum.

L. Lee. 1999. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25-32, College Park, MA.

L. Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Proc. International Workshop on Artificial Intelligence and Statistics*, pages 65–77.

D. Lenat, A, Borning, D. McDonald, C. Taylor, and S. Weyer. 1983. Knoesphere: Building expert systems with encyclopedic knowledge. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 167–169, Karlsruhe, Germany, August.

D. Lenat. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, November.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference 1986*, pages 24-26, Toronto, June.

V. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848. In Russian. English translation in *Soviet Physics Doklady*, 10(8):707–710, 1966.

H. Lieberman, A. Faaborg, W. Daher, and J. Espinosa. 2005. How to wreck a nice beach you sing calm incense. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*. Pages 278-280, San Diego, California, USA.

C. Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 74-81, Barcelona, Spain, July 25 - 26, 2004.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296-304, Madison, WI.

F. Lin and K. Sandkuhl. 2008. A survey of exploiting WordNet in ontology matching. In *Artificial Intelligence in Theory and Practice II, IFIP 20th World Computer Congress, TC 12: IFIP AI 2008 Stream*, September 7-10, 2008, Milano, Italy, volume 276 of IFIP, pages 341–350. Springer.

J. Lin. 2009. Brute Force and Indexed Approaches to Pairwise Document Similarity Comparisons with MapReduce. In *Proceedings of the 32nd Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009), pages 155–162. Boston, MA, USA. ACM.

H. Liu and P. Maes. 2004. What would they think? A computational model of attitudes. In *Proceedings of the ACM Conference on Intelligent User Interfaces*, pages 38–45, Madeira, Portugal.

H. Liu, and P. Singh. 2004. ConceptNet: A practical commonsense reasoning toolkit. BT Technology Journal 22(4):211–226.

K. Malatesta, P. Wiemer-Hastings, and J. Robertson. 2002. Beyond the short answer question with research methods tutor. In *ITS '02: Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 562–573, London, UK. Springer-Verlag.

V. Mascardi, A. Locoro, and F. Larosa. 2009a. Exploiting Prolog and NLP techniques for matching ontologies and for repairing correspondences. In *2009 Italian Conference on Computational Logic (CILC 2009)*, Ferrara, Italy, June.

V. Mascardi, A. Locoro, and P. Rosso. 2009b. Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 29 June 2009. IEEE Computer Society/IEEE Computer Society Digital Library. http://doi.ieeecomputersociety.org/10.1109/TKDE.2009.154

C. McKinstry, R. Dale, and M. Spivey. 2008. Exploring action dynamics as an index of paired-associate learning. *Psychological Science: A Journal of the American Psychological Society / APS*, 19:22–24, Jan.

O. Medelyan and C. Legg. 2008. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WikiAI'08)*, Chicago.

O. Medelyan, D. Milne, and I. Witten. 2008a. Topic indexing with Wikipedia. In *Proceedings of the Wikipedia and AI workshop at AAAI-08*. AAAI. Associated data http://www.cs.waikato.ac.nz/~olena/wikipedia.html

O. Medelyan, C. Legg, D. Milne, and I.Witten. 2008b. Mining Meaning from Wikipedia. September. http://arxiv.org/abs/0809.4530

S. Melnik, H. Garcia-Molina, and E. Rahm. 2002. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In *Proceedings of the 18th International Conference on Data Engineering (ICDE),* pages 117–128.

R. Mihalcea, C. Corley, and C. Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *21st Conference of American Association for Artificial Intelligence (AAAI-06)*, pages 775–780. Boston, MA, USA. AAAI.

R. Mihalcea and A. Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 233-242, Lisbon, Portugal.

R. Mihalcea, P. Tarau, and E. Figa. 2004. TextRank: bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 404-411, Barcelona, Spain.

G. Miller and W. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6:1-28.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

G. Miller. 1995. WordNet: A lexical database. *Communications of the ACM*, 38(11):39–41.

D. Milne and I. Witten. 2008a. Learning to link with Wikipedia. In *Proceedings of the Seventeenth ACM Conference on Information and Knowledge Management*, pages 509-518, Napa Valley, California, USA.

D. Milne and I. Witten. 2008b. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08).*

S. Mohammad and G. Hirst. 2005. Distributional measures as proxies for semantic relatedness. In submission, http://www.cs.toronto.edu/compling/Publications

M. Mohler and R. Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference (EACL 2009)*, pages 567–575, Athens, Greece. The Association for Computer Linguistics.

E. Mueller. 2003. ThoughtTreasure: A natural language/commonsense platform. http://alumni.media.mit.edu/~mueller/papers/tt.html

The OpenNLP Project. http://www.opennlp.org

M. Porter. 1980. An algorithm for suffix stripping, *Program*, 14(3):130-137.

S. Patwardhan and T. Pedersen. 2006. Using WordNet-Based Context Vectors to Estimate the Semantic Relatedness of Concepts, In *Proceedings of the EACL 2006 Workshop on Making Sense of Sense, Bringing Psycholinguistics and Computational Linguistics Together*, pages 1-8 .

T. Pedersen , S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity-Measuring the relatedness of concepts. In *Proceedings of the Nineethn National Conference on Artificial Intelligence. (AAAI-04).*

G.Pilato, A. Augello, M. Scriminaci, G. Vassallo, and S. Gaglio. 2007. Sub-Symbolic Mapping of Cyc Microtheories in Data-Driven Conceptual Spaces, In *Proc of 11th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2007)* Vietrisul Mare, Italy (12-14 September 2007). In *Lecture Notes in Artificial Intelligence*, Springer-Verlag vol. 4692/2007, pages 156-163.

S. Pulman and J. Sukkarieh. 2005. Automatic Short Answer Marking. *ACL Workshops , Proceedings of the second workshop on Building Educational Applications Using NLP,* pages 9-16, Ann Arbor , Michigan,

Y. Ollivier and P. Senellart. 2007. Finding related pages using green measures: An illustration with Wikipedia. In *Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI 2007),* pages 1427-1433.

R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

C. Quirk, C. Brockett, and W. Dolan. 2004. Monolingual Machine Translation for Paraphrase Generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 142-149, Barcelona, Spain.

D. Ramachandran, P. Reagan, and K. Goolsbey. 2005. First-orderized ResearchCyc: Expressivity and efficiency in a common-sense ontology. In *Papers from the AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*, pages 33-40, Pittsburgh, PA, July.

D. Ramage, A. Rafferty, and C. Manning. 2009. Random walks for text semantic similarity. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 23–31, Suntec, Singapore, August. Association for Computational Linguistics.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 130–142, Philadelphia, May.

A. Ratnaparkhi. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis. Computer and Information Science, University of Pennsylvania, 1998.

P. Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448-453, Montreal, Canada.

G. Salton and C. Buckley. 1997. Term weighting approaches in automatic text retrieval. In *Readings in Information Retrieval* , pages 323-328, San Francisco. CA: Morgan Kaufmann Publishers.

H. Schutze and C.Manning. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

S. Sarjant, C. Legg, and O. Medelyan. 2009. All you can eat ontology-building: Feeding Wikipedia to Cyc. In *2009 IEEE/WIC/ACM International Conference on Web Intelligence (WI-09)*, pages 341-348, Milano, Italy, September.

H. Simon. 1971. Designing organizations for an information rich world. In *Computers, Communications, and the Public Interest*, pages 37–72, Baltimore, MD.

P. Singh. 2002. The Open Mind Common Sense project. KurzweilAI.net, January. Available online from http://www.kurzweilai.net/

P. Singh and W. Williams. 2003. LifeNet: a propositional model of ordinary human activity. *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP) at K-CAP 2003*.

P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Zhu. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/Coop IS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237, London, UK: Lecture Notes in Computer Science, vol. 2519. Springer-Verlag.

R. Snow, D. Jurafsky, and A. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *ACL-44: Proceedings of the 21st International Conference*

on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 801–808, Morristown, NJ, USA.

S. Soderland and B. Mandhani. 2007. Moving from textual relations to ontologized relations. In *AAAI Spring Symposium on Machine Reading*, pages 85-90.

S. Sood and K. Hammond. 2007. Tagassist: Automatic tag suggestion for blog posts. In *International Conference on Weblogs and Social Media*, March.

R. Speer. 2007. Learning common sense knowledge from user interaction and principal component analysis. MS Thesis. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science.

R. Speer, C. Havasi, and H. Lieberman. 2008. Analogyspace: Reducing the dimensionality of common sense knowledge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 548–553. AAAI Press.

M. Strube and S. P. Ponzetto. 2006. Wikirelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the American Association for Artificial Intelligence*, pages 1419-1424, Boston, MA.

Z. Syed, T. Finin, and A. Joshi. 2008. Wikipedia as an Ontology for Describing Documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March.

L. Vanderwende, G. Kacmarcik, H. Suzuki, and A. Menezes. 2005. MindNet: An automatically-created lexical resource. In *Proceedings of HLT/EMNLP on interactive demonstrations,* pages 8-9, Vancouver, British Columbia, Canada, October . Human Language Technology Conference. Association for Computational Linguistics.

T. White. 2009. *Hadoop: The Definitive Guide*. O'Reilly Media, ISBN 978-0-596-52197-4.

P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. 1999. Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. *Artificial Intelligence in Education*, pages 535–542.

P. Wiemer-Hastings, E. Arnott, and D. Allbritton. 2005. Initial results and mixed directions for research methods tutor. In *AIED2005 – Supplementary Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam.

D. Widdows and K. Ferraro. 2008. Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application, Paper presented at the *6th International Conference on Language Resources and Evaluation (LREC 2008)* , Marrakech, Morocco.

I. Witten and E. Frank. 2004. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, CA, USA. 2nd edition. Available at http://www.cs.waikato.ac.nz/ml/weka/

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 133-138.

E. Yeh, D. Ramage, C. Manning, E. Agirre, and A. Soroa. 2009. Wikiwalk: Random walks on Wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 41–49, Suntec, Singapore, August. Association for Computational Linguistics. http://www.stanford.edu/~dramage/papers/wikiwalk-textgraphs09.pdf