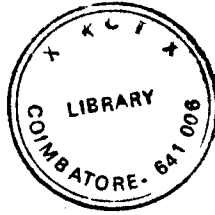


P-3225



**SEMANTIC WEB-A NEW MODEL OF SEMANTIC SIMILARITY
MEASURING IN WORDNET**

PROJECT REPORT

Submitted By

P. GOPALAKRISHNAN

Register No.: 0720300008

*in partial fulfillment for the award of the degree
of*

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

May 2010

KUMARAGURU COLLEGE OF TECHNOLOGY
(An Autonomous Institution Affiliated to Anna University, Coimbatore)
COIMBATORE – 641 006.

Department of Computer Applications

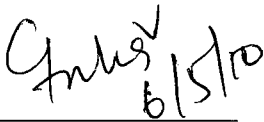
PROJECT WORK

MAY 2010

This is to certify that the project entitled
SEMANTIC WEB-A NEW MODEL OF SEMANTIC SIMILARITY
MEASURING IN WORDNET
is the bonafide record of project work done by

P. GOPALAKRISHNAN

Register No: 0720300008 of MCA (Computer Applications) during the year
2009-2010.

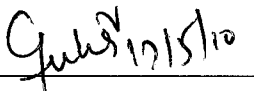

6/5/10

Project Guide

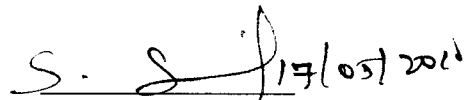


Head of the Department

Submitted for the Project Viva-Voce examination held on 17-05-2010


17/5/10

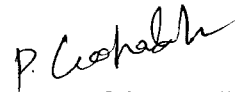
Internal Examiner


17/05/2010

External Examiner

DECLARATION

I affirm that the project work titled '**SEMANTIC WEB-A NEW MODEL OF SEMANTIC SIMILARITY MEASURING IN WORDNET**' being submitted in partial fulfilment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.



(Signature of the Candidate)

P. GOPALAKRISHNAN

Name of the Candidate

0720300008

Register Number

I certify that the declaration made above by the candidate is true



Signature of the Guide,

V. Geetha
AP/MCA

With Name & Designation



Cognizant

This is to certify that **Gopalkrishnan P**, a **MCA** student of **Kumaraguru College of Technology** has done project work in the company on Semantic Web Mining under the guidance of Ganesh Kumar R, as part of the college requirement, between the period January 2010 and May 2010.

Yours sincerely,

for **Cognizant Technology Solutions India Pvt. Ltd.**

J. Sriram

Sriram Iyer
Manager - Human Resources

I accept the terms and conditions of the offer as mentioned above.

Signature: *P. Ustalan*

Date: 12/04/2010

ACKNOWLEDGEMENT

I wish to express my deep unfathomable feeling of gratitude and indebtedness to **Dr.J.Shanmugam, Director**, Kumaraguru College of Technology, Coimbatore for the successful completion of the project work.

I wish to express my sincere thanks to **Dr.S.Ramachandran, Principal**, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

I am very glad to express a special word of thanks to **Dr.A.Muthukumar, Professor and Course coordinator**, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for encouraging me to do this work.

I am very much indebted to **Ms.V.Geetha, Assistant Professor**, Project Coordinator, Department of Computer Applications, Kumaraguru College of Technology, and Coimbatore for her complete assistance, guidance and support given to me throughout my project.

It is my pleasure to express my profound gratitude to **Cognizant Technology Solutions India Pvt. Ltd, Coimbatore** for admitting into this project. I am thankful to **Mr.R.Ganeshkumar** of Cognizant Technology Solutions India Pvt. Ltd, Coimbatore, for this excellent guidance, timely suggestions and constant support in all my endeavors.

Finally, I owe a lot to my beloved parents and family members and to my friends and to my department staffs for their help and co-operation to complete this project successfully.

ABSTRACT

Semantic similarity is a concept in which a set of documents or terms within term lists are assigned a metric based on the likeness of their semantic content. Semantic similarity measures play an increasingly important role in text-related research and applications in areas such as text mining, Web page retrieval, and dialogue systems. Existing methods for computing sentence similarity have been adopted from approaches used for long text documents. These methods process sentences in a very high-dimensional space and are consequently inefficient, require human input, and are not adaptable to some application domains. In this project the implementation focuses directly on computing the similarity between two words.

The semantic similarity of two words is calculated using information from a structured lexical database and from corpus statistics. The use of a lexical database enables our method to model human common sense knowledge and the incorporation of corpus statistics allows our method to be adaptable to different domains.

The algorithms that were implemented as a part of the project are Micheal Lesk algorithm, Wu and Palmer algorithm, Leacock and Chodorow algorithm. The project has been developed using “ASP.NET” as front end and “C#” as code behind language.

TABLE OF CONTENTS

CHAPTER	PAGE
Acknowledgement	iv
Abstract	v
List of figures	viii
1. INTRODUCTION	1
1.1 Introduction to Semantic similarity	1
1.2 Applications of Semantic similarity	3
1.3 Wordnet	4
1.3.1 Database Content	5
1.3.2 Application	6
1.3.3 Problems and limitations	7
1.3.4 Knowledge Structure	8
1.4. Development Environment	10
1.4.1 Hardware Configurion	10
1.4.2 Software Configuration	10
2. TYPES OF SIMILARITY APPROACHES	14
2.1. The path length based similarity measurement	14
2.2. Node based approach	16
2.3. Comparison of the two approaches	18

3. SEMANTIC SIMILARITY ALGORITHMS	19
3.1. Micheal Lesk algorithm	19
3.2. Leacock and Chodorow algorithm	21
3.3. Wu and Palmer algorithm	22
4. TESTING	23
4.1 Unit testing	23
4.2 Integration testing	24
5. CONCLUSION	25
APPENDIX	29
BIBLIOGRAPHY	31

LIST OF FIGURES

S.NO	FIG.NO	FIGURE NAME	PAGE NO
1	1.1	SAMPLE WORDNET STRUCTURE	5
2	2.1	PATH LENGTH SIMILARITY MEASUREMENT	11
3	2.2	NODE-BASED APPROACH	14

CHAPTER 1

INTRODUCTION

This chapter describes the basics of semantic similarity, the development environment and the tools used.

1.1 INTRODUCTION TO SEMANTIC SIMILARITY

Semantic similarity is the process of finding the relation between words in a given context. This process is mainly implemented by using hierarchical based data structures.

Automatic extraction of semantic information from text and links in Web pages is key to improving the quality of search results. WordNet, similarity is a freely available software package that makes it possible to measure the semantic similarity and relatedness between a pair of concepts (or synsets). It provides six measures of similarity, and three measures of relatedness, all of which are based on the lexical database WordNet.

According to some opinions the concept of semantic similarity is different from semantic relatedness because semantic relatedness includes concepts as antonym and meronym, while similarity doesn't. However, much of the literature uses these terms interchangeably, along with terms like semantic distance. Semantic similarity measures have been recently applied and developed in the biomedical fields.

1.2 APPLICATIONS OF SEMANTIC SIMILARITY

Semantic similarity measures have been recently applied and developed in biomedical ontology namely, the Gene Ontology(GO). They are mainly used to compare genes and proteins based on the similarity of their functions rather than on their sequence similarity. These comparisons can be done using some tools freely available on the web.

- ProteInOn can be used to find interacting proteins, find assigned GO terms and calculate the functional semantic similarity of proteins and to get the information content and calculate the functional Semantic similarity of GO terms.

WordNet, similarity implements measures of similarity and relatedness that are all in some way based on the structure and content of WordNet. Measures of similarity use information found in an is-a hierarchy of concepts (or synsets), and quantify how much concept A is like (or is similar to) concept B. For example, such a measure might show that an automobile and boat share vehicle as an ancestor in the WordNet noun hierarchy.

1.3 WORDNET

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold,

- To produce a combination of dictionary and thesaurus that is more intuitively usable, and
- To support automatic text analysis and artificial intelligence applications.

The database and software tools have been released under a BSD style license and can be downloaded and used freely. The database can also be browsed online. WordNet was created and is being maintained at the Cognitive Science Laboratory of Princeton University under the direction of psychology professor George A. Miller.

Development began in 1985. Over the years, the project received funding from government agencies interested in machine translation. WordNet has been supported by grants from the National Science Foundation, DARPA, the Disruptive Technology Office (formerly the Advanced Research and Development Activity), and REFLEX. George Miller and Christiane Fellbaum were awarded the 2006 Antonio Zampolli Prize for their work with WordNet.

1.3.1 Database contents

As of 2006, the database contains about 150,000 words organized in over 115,000 synsets for a total of 207,000 word-sense pairs in compressed form, it is about 12 megabytes in size.

WordNet distinguishes between nouns, verbs, adjectives and adverbs because they follow different grammatical rules. Every synset contains a group of synonymous words or collocations (a collocation is a sequence of words that go together to form a specific meaning, such as "car pool"), different senses of a word are in different synsets. The meaning of the synsets is further clarified with short defining glosses (Definitions and/or example sentences).

Most synsets are connected to other synsets via a number of semantic relations. These relations vary based on the type of word, and include:

- Nouns
 - Hypernyms: Y is a hypernym of X if every X is a (kind of) Y (canine is a hypernym of dog)
 - Hyponyms: Y is a hyponym of X if every Y is a (kind of) X (dog is a hyponym of canine)
 - Coordinate terms: Y is a coordinate term of X if X and Y share a hypernym (wolf is a coordinate term of dog, and dog is a coordinate term of wolf)
 - Holonym: Y is a holonym of X if X is a part of Y (building is a holonym of window)
 - Meronym: Y is a meronym of X if Y is a part of X (window is a meronym of building)
- Verbs
 - Hypernym: the verb Y is a hypernym of the verb X if the activity X is a (kind of) Y (to perceive is an hypernym of to listen)
 - Troponym: the verb Y is a troponym of the verb X if the activity Y is doing X in some manner (to lisp is a troponym of to talk)

- Entailment: the verb Y is entailed by X if by doing X you must be doing Y (to sleep is entailed by to snore)
- Coordinate terms: those verbs sharing a common hypernym (to lisp and to yell)
- Adjectives
 - related nouns
 - similar to
 - participle of verb
- Adverbs
 - root adjectives

While semantic relations apply to all members of a synset because they share a meaning but are all mutually synonyms, words can also be connected to other words through lexical relations, including antonyms (opposites of each other) which are derivationally related, as well.

WordNet also provides the polysemy count of a word, the number of synsets that contain the word. If a word participates in several synsets (i.e. has several senses) then typically some senses are much more common than others.

WordNet quantifies this by the frequency score, in which several sample texts have all words semantically tagged with the corresponding synset, and then a count provided indicating how often a word appears in a specific sense.

The morphology functions of the software distributed with the database try to deduce the lemma or root form of a word from the user's input; only the root form is stored in the database unless it has irregular inflected forms.

1.3.2 Application

WordNet has been used for a number of different purposes in information systems, including word sense disambiguation, information retrieval, automatic text classification, automatic text summarization, and even automatic crossword puzzle generation. The sample Wordnet structure is presented in Fig 1.1.

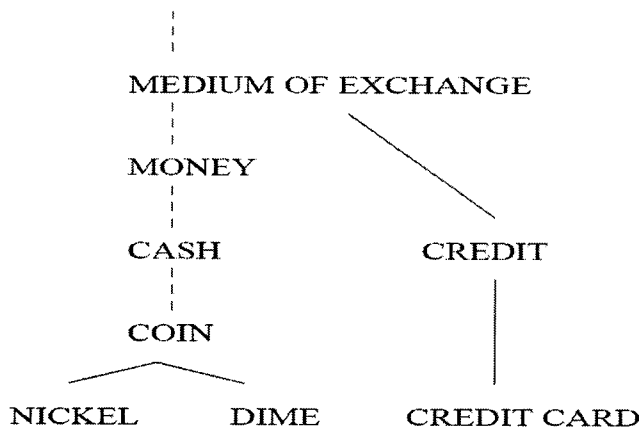


Fig 1.1 Sample WordNet structure

1.3.3 Problems and limitations

Unlike other dictionaries, WordNet does not include information about etymology, pronunciation and the forms of irregular verbs and contains only limited information about usage.

The actual lexicographical and semantic information is maintained in lexicographer files, which are then processed by a tool called grind to produce the distributed database. Both grind and the lexicographer files are freely available in a separate distribution, but modifying and maintaining the database requires expertise.

Though WordNet contains a sufficiently wide range of common words, it does not cover special domain vocabulary. Since it is primarily designed to act as an underlying database for different applications, those applications cannot be used in specific domains that are not covered by WordNet.

In most works that claim to have integrated WordNet into other ontology, the content of WordNet has not simply been corrected when semantic problems have been encountered instead, WordNet has been used as an inspiration source but heavily re-interpreted and updated whenever suitable.

This was the case when, for example, the top-level ontology of WordNet was re-structured according to the OntoClean based approach or when WordNet was used as a primary source for constructing the lower classes of the SENSUS ontology.

WordNet is the most commonly used computational lexicon of English for Word Sense Disambiguation (WSD), a task aimed to assigning the most appropriate senses (i.e. synsets) to words in context. However, it has been argued that WordNet encodes sense distinctions that are too fine-grained even for humans. This issue prevents WSD systems from achieving high performance. The granularity issue has been tackled by proposing clustering methods that automatically group together similar senses of the same word.

1.3.4 Knowledge structure

Both nouns and verbs are organized into hierarchies, defined by hypernym or IS A relationships. For instance, the first sense of the word dog would have the following hypernym hierarchy. The words at the same level are synonyms of each other. Each set of synonyms (synset), has a unique index and shares its properties, such as a gloss (or dictionary) definition.

At the top level, these hierarchies are organized into base types, 25 primitive groups for nouns, and 15 for verbs. These groups form lexicographic files at a maintenance level. These primitive groups are connected to an abstract root node that has, for some time, been assumed by various applications that use WordNet.

In the case of adjectives, the organization is different. Two opposite 'head' senses work as binary poles, while 'satellite' synonyms connect to each of the heads via synonymy relations. Thus, the hierarchies, and the concept involved with lexicographic files, do not apply here the same way they do for nouns and verbs.

The network of nouns is far deeper than that of the other parts of speech. Verbs have a far bushier structure, and adjectives are organized into many distinct

clusters. Adverbs are defined in terms of the adjectives they are derived from, and thus inherit their structure from that of the adjectives.

The goal of WordNet was to develop a system that would be consistent with the knowledge acquired over the years about how human beings process language. Anomic aphasia, for example, creates a condition that seems to selectively encumber individuals ability to name objects, this makes the decision to partition the parts of speech into distinct hierarchies more of a principled decision than an arbitrary one.

In the case of hyponymy, psychological experiments revealed that individuals can access properties of nouns more quickly depending on when a characteristic becomes a defining property.

That is, individuals can quickly verify that canaries can sing because a canary is a songbird (only one level of hyponymy), but require slightly more time to verify that canaries can fly (two levels of hyponymy) and even more time to verify canaries have skin (multiple levels of hyponymy).

This suggests that we too store semantic information in a way that is much like WordNet, because we only retain the most specific information needed to differentiate one particular concept from similar concepts.

1.4 DEVELOPMENT ENVIRONMENT

This section describes about, the hardware and software requirements for the project.

1.4.1 HARDWARE CONFIGURATION

The hardware requirements are,

Processor	: Intel Dual Core Processor
Clock Speed	: 1.73 GHz
Primary Memory	: 1 GB RAM
Hard Disk Drive	: 160 GB

1.4.2 SOFTWARE CONFIGURATION

The software requirements are,

Operating System	: Windows Vista Service Pack 2
Programming Language	: C#.net
Tools Used	: Visual studio 2008, WordNet 2.1.

1.4.3. VISUAL STUDIO 2008

Visual Studio 2008, and Visual Studio Team System 2008 codenamed Orcas, was released to MSDN subscribers on 19 November 2007 alongside .NET Framework 3.5. The codename Orcas is, like Whidbey, a reference to an island in Puget Sound, Orcas Island. The source code for the Visual Studio 2008 IDE will be available under a shared source license to some of Microsoft's partners and ISVs. Microsoft released Service Pack 1 for Visual Studio 2008 on 11 August 2008. The internal version number of Visual Studio 2008 is version 9.0 while the file format version is 10.0.

Common Language Infrastructure (CLI)

The purpose of the Common Language Infrastructure, or CLI, is to provide a language-neutral platform for application development and execution, including functions for exception handling, garbage collection, security, and interoperability. By implementing the core aspects of the .NET Framework within the scope of the CLR, this functionality will not be tied to a single language but will be available across the many languages supported by the framework. Microsoft's implementation of the CLI is called the Common Language Runtime, or CLR.

The .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built in APIs are part of either System or namespaces. These class libraries implement a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all CLI compliant languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library.

The **Base Class Library (BCL)** includes a small subset of the entire class library and is the core set of classes that serve as the basic API of the Common Language Runtime. The classes in mscorlib.dll and some of the classes in System.dll and System.core.dll are considered to be a part of the BCL. The BCL classes are available in both .NET Framework as well as its alternative implementations including .NET Compact Framework, Microsoft Silverlight and Mono.

The **Framework Class Library (FCL)** is a superset of the BCL classes and refers to the entire class library that ships with .NET Framework. It includes an expanded set of libraries, including Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation among others. The FCL is much larger in scope than standard libraries for languages like C++, and comparable in scope to the standard libraries of Java.

Microsoft Visual C# is Microsoft's implementation of the C# programming language specification, included in the Microsoft Visual Studio suite of products. It is based on the ECMA/ISO specification of the C# language, which Microsoft also created. While multiple implementations of the specification exist, Visual C# is by far the one most commonly used in most contexts, an unqualified reference to "C#" is taken to mean "Visual C#."

Some of the advantages of creating C# applications in Visual Studio.NET are

- ✓ Visual Studio.NET is a Rapid Application (RAD) tool. Instead of adding each control to the form programmatically, it helps to add these controls by using Toolbox, saving programming efforts.
- ✓ Visual Studio.NET supports custom and composite controls. Can create custom controls that encapsulate a common functionality that might be used in a number of applications.
- ✓ Visual Studio.NET does a wonderful job of simplifying the creation and consumption of Web Services. Much of the programmer-friendly stuff (creating all the XML-based documents) happens automatically, without much effort on the programmer's side. Attribute based programming is a powerful concept that enables Visual Studio.NET automate a lot of programmer-unfriendly tasks.

.NET FRAMEWORK 3.5

Version 3.5 of the .NET Framework was released on 19 November 2007, but it is not included with Windows Server 2008. As with .NET Framework 3.0, version 3.5 uses the CLR of version 2.0. In addition, it installs .NET Framework 2.0 SP1, (installs .NET Framework 2.0 SP2 with 3.5 SP1) and .NET Framework 3.0 SP1 (installs .NET Framework 3.0 SP2 with 3.5 SP1), which adds some methods and properties to the BCL classes in version 2.0.

CHAPTER 2

TYPES OF SIMILARITY APPROACHES

This chapter describes about the different approaches used to find similarity and comparison of these approaches.

2.1 THE PATH LENGTH-BASED SIMILARITY MEASUREMENT

To measure the semantic similarity between two synsets, hyponym/hypernym (or is-a relations) is used.

A simple way to measure the semantic similarity between two synsets is to treat taxonomy as an undirected graph and measure the distance between them in WordNet. Said P. Resnik. "The shorter the path from one node to another, the more similar they are". The path length is measured in nodes/vertices rather than in links/edges. The length of the path between two members of the same synset is 1 (synonym relations).

Fig 2.1 shows an example of the hyponym taxonomy in WordNet used for path length similarity measurement.

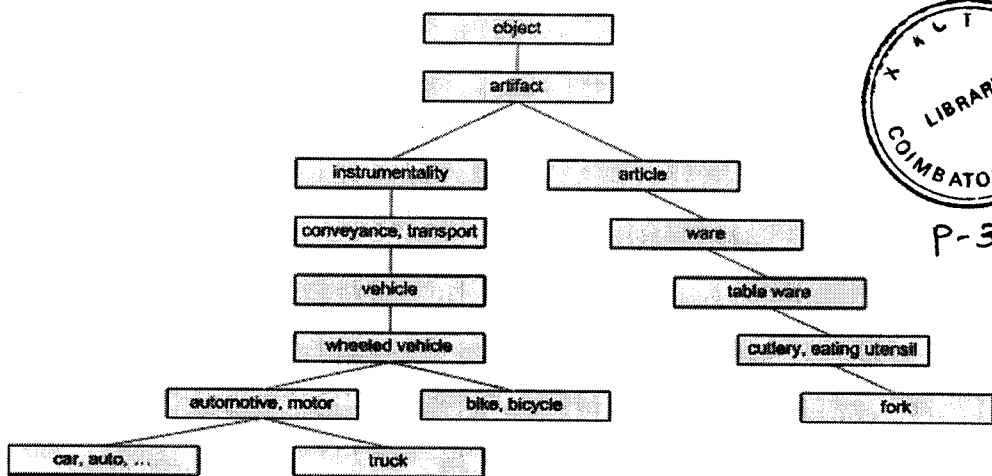


Fig 2.1 Path length similarity measurement

It can be observed that the length between car and auto is 1, car and truck is 3, car and bicycle is 4, car and fork is 12.

A shared parent of two synsets is known as a sub-sumer. The least common sub-sumer (LCS) of two synsets is the sumer that does not have any children that are also the sub-sumer of two synsets. In other words, the LCS of two synsets is the most specific sub-sumer of the two synsets.

Back to the above example, the LCS of {car, auto..} and {truck..} is {automotive, motor vehicle}, since the {automotive, motor vehicle} is more specific than the common sub-sumer {wheeled vehicle}.

The edge based approach is a more natural and direct way of evaluating semantic similarity in a taxonomy. It estimates the distance (e.g. edge length) between nodes which correspond to the concepts/classes being compared. Given the multidimensional concept space, the conceptual distance can conveniently be measured by the geometric distance between the nodes representing the concepts. Obviously, the shorter the path from one node to the other, the more similar.

In a more realistic scenario, the distances between any two adjacent nodes are not necessarily equal. It is therefore necessary to consider that the edge connecting the two nodes should be weighted. To determine the edge weight automatically, certain aspects should be considered in the implementation. Most of these are typically related to the structural characteristics of a hierarchical network.

2.2 NODE-BASED (INFORMATION CONTENT) APPROACH

One node based approach to determine the conceptual similarity is called the information content approach. Given a multidimensional space upon which a node represents a unique concept consisting of a certain amount of information, and an edge represents a direct association between two concepts, the similarity between two concepts is the extent to which they share information in common. The value of the information content of a class is then obtained by estimating the probability of occurrence of this class

in a large text corpus. Following the notation in information theory, the information content (IC) of a concept/class c can be quantified as follows:

$$IC(c) = \log^{-1} P(c)$$

where $P(c)$ is the probability of encountering an instance of concept c .

In the case of the hierarchical structure, where a concept in the hierarchy subsumes those lower in the hierarchy, this implies that $P(c)$ is monotonic as one moves up the hierarchy. As the node's probability increases, its information content or its informativeness decreases.

If there is a unique top node in the hierarchy, then its probability is 1, hence its information content is 0.

This methodology can be best illustrated by examples,

Assume that the similarities between the following classes need to be determined (car, bicycle) and (car, fork).

Fig 2.2 depicts the fragment of the WordNet noun hierarchy that contains these classes. The number in the bracket of a node indicates the corresponding information content value. From the figure it can be seen that the similarity between car and bicycle is the information content value of the class vehicle, which has the maximum value among all the classes that subsume both of the two classes, i.e. $\text{sim}(\text{car}, \text{bicycle}) = 8.30$.

In contrast, the relationship between the car & fork is $\text{sim}(\text{car}, \text{fork}) = 3.53$. These results conform to the perception that cars and forks are less similar than cars and bicycles.

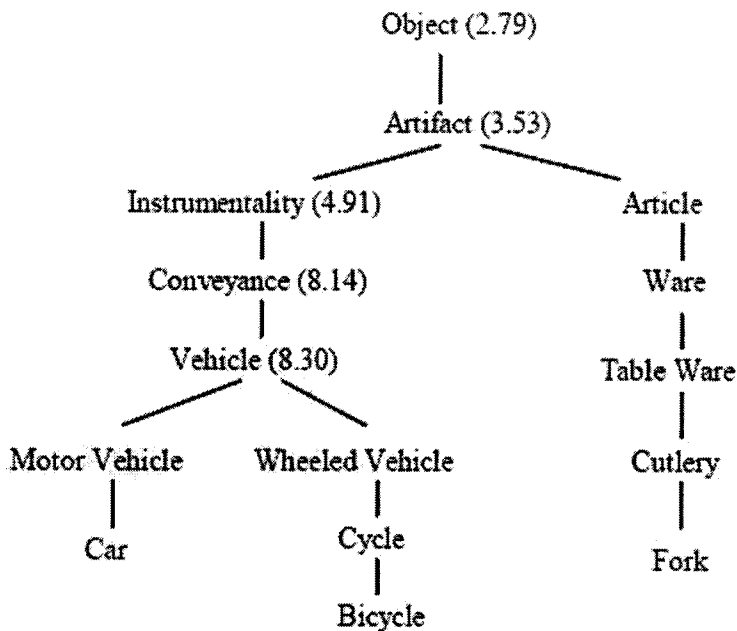


Fig 2.2 Node-based Approach

2.3 COMPARISON OF THE TWO APPROACHES

The two approaches target semantic similarity from quite different angles. The edge-based distance method is more intuitive, while the node-based information content approach is more theoretically sound. Both have inherent strength and weakness.

In addition, we feel that the distance measure is highly dependent upon the subjectively pre-defined network hierarchy. Since the original purpose of the design of the WordNet was not for similarity computation purpose, some local network layer constructions may not be suitable for the direct distance manipulation. The information content method requires less information on the detailed structure of taxonomy. It is not sensitive to the problem of varying link types (Resnik 1995). However, it is still dependent on the skeleton structure of the taxonomy. Just because it ignores information on the structure it has its weaknesses. It normally generates a coarse result for the comparison of concepts. In particular, it does not differentiate the similarity values of any

pair of concepts in a sub-hierarchy as long as their “smallest common denominator” (i.e. the lowest super-ordinate class) is the same.

For example, given the concepts in Fig 2.1, the results of the similarity evaluation between (bicycle, table ware) and (bicycle, fork) would be the same. Also, other type of link relations information is overlooked here. Additionally, in the calculation of information content, polysemous words will have an exaggerated content value if only word (not its sense) frequency data are used.

CHAPTER 3

SEMANTIC SIMILARITY ALGORITHMS

This chapter describes about the various Semantic similarity Algorithms.

3.1 MICHEAL LESK ALGORITHM

A word can have more than one sense that can lead to ambiguity. For example, the word "interest" has different meanings in the following two contexts:

- "Interest" from a bank.
- "Interest" in a subject.

Disambiguation is the process of finding out the most appropriate sense of a word that is used in a given sentence. The major objective of its idea is to count the number of words that are shared between two glosses. To disambiguate a word, the gloss of each of its senses is compared to the glosses of every other word in a phrase. A word is assigned to the sense whose gloss shares the largest number of words in common with the glosses of the other words.

For example, in performing disambiguation for the "pine cone" phrasal, according to the Oxford Advanced Learner's Dictionary, the word "pine" has two senses:

- Sense 1: kind of evergreen tree with needle-shaped leaves,
- Sense 2: waste away through sorrow or illness.

The word "cone" has three senses:

- Sense 1: solid body which narrows to a point,
- Sense 2: something of this shape, whether solid or hollow,
- Sense 3: fruit of a certain evergreen tree.

By comparing each of the two gloss senses of the word "pine" with each of the three senses of the word "cone", it is found that the words "evergreen tree" occurs in one sense in each of the two words. So, these two senses are then declared to be the most appropriate senses when the words "pine" and "cone" are used together.

When computing the relatedness between two synsets s_1 and s_2 , the pair hype-hype means the gloss for the hypernym of s_1 is compared to the gloss for the hypernym of s_2 . The pair hype-hypo means that the gloss for the hypernym of s_1 is compared to the gloss for the hyponym of s_2 .

$$\text{OverallScore}(s_1, s_2) = \text{Score}(\text{hype}(s_1)\text{-hypo}(s_2)) + \text{Score}(\text{gloss}(s_1)\text{-hypo}(s_2)) + \text{Score}(\text{hype}(s_1)\text{-gloss}(s_2)) \dots$$

(OverallScore (s_1, s_2) is also equivalent to OverallScore (s_2, s_1)).

In the example of "pine cone", there are three senses of pine and 6 senses of cone, so we can have a total of 18 possible combinations. One of them is the right one.

The above method allows us to find the most appropriate sense for each word in a sentence. To compute the similarity between two sentences, we base the semantic similarity between word senses. We capture Semantic similarity between two word senses based on the path length similarity.

In WordNet, each part of speech words (nouns/verbs...) are organized into taxonomies where each node is a set of synonyms (synset) represented in one sense. If a word has more than one sense, it will appear in multiple synsets at various locations in the taxonomy. WordNet defines relations between synsets and relations between word senses. A relation between synsets is a semantic relation, and a relation between word senses is a lexical relation. The difference is that lexical relations are relations between members of two different synsets, but semantic relations are relations between two whole synsets.

For instance:

- Semantic relations are hypernym, hyponym, holonym, etc.
- Lexical relations are antonym relation and the derived from relation.

Using the example, the antonym of the tenth sense of the noun light (light#n#10) in WordNet is the first sense of the noun dark (dark#n#1). The synset to which it belongs

is {light#n#10, lighting#n#1}. Clearly, it makes sense that light#n#10 is an antonym of dark#n#1, but lighting#n#1 is not an antonym of dark#n#1; therefore, the antonym relation needs to be a lexical relation, not a semantic relation. Semantic similarity is a special case of semantic relatedness where we only consider the IS-A relationship.

This formula was used to find the similarity, which not only took into account the length of the path, but also the order of the sense involved in this path:

$$\text{Sim}(s, t) = \text{SenseWeight}(s) * \text{SenseWeight}(t) / \text{PathLength}$$

where s and t: denote the source and target words being compared.

SenseWeight: denotes a weight calculated according to the frequency of use of this sense and the total of frequency of use of all senses.

PathLength: denotes the length of the connection path from s to t.

3.2 LEACOCK & CHODOROW ALGORITHM

The relatedness measure proposed by Leacock and Chodorow (lch) is

$$\text{Sim} = -\log(\text{length} / (2 * D))$$

where length is the length of the shortest path between the two synsets (using node-counting) and D is the maximum depth of the taxonomy.

The fact that the lch measure takes into account the depth of the taxonomy in which the synsets are found means that the behavior of the measure is profoundly affected by the presence or absence of a unique root node. If there is a unique root node, then there are only two taxonomies, one for nouns and one for verbs.

All nouns, then, will be in the same taxonomy and all verbs will be in the same taxonomy. D for the noun taxonomy will be somewhere around 18, depending upon the version of WordNet, and for verbs, it will be 14. If the root node is not being used, however, then there are nine different noun taxonomies and over 560 different verb taxonomies, each with a different value for D.

If the root node is not being used, then it is possible for synsets to belong to more than one taxonomy. The relatedness is computed by finding the LCS that results in the shortest path between the synsets. The value of D, then, is the maximum depth of the taxonomy in which the LCS is found. If the LCS belongs to more than one taxonomy, then the taxonomy with the greatest maximum depth is selected (i.e., the largest value for D).

3.3 WU & PALMER ALGORITHM

The Wu & Palmer measure (wup) calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS. The formula is

$$\text{Sim} = 2 * \text{depth}(\text{lcs}) / (\text{depth}(\text{s1}) + \text{depth}(\text{s2})).$$

This means that $0 < \text{score} \leq 1$. The score can never be zero because the depth of the LCS is never zero (the depth of the root of taxonomy is one). The score is one if the two input synsets are the same. wup finds the depth of the LCS of the concepts, and then scales that by the sum of the depths of the individual concepts. The depth of a concept is simply its distance to the root node. The measure path is a baseline that is equal to the inverse of the shortest path between two concepts.

For example, substitute the value in the formula.

where,

depth(lcs) - 7

depth(s1) - 8

depth(s2) - 8

Similarity = 0.875.

CHAPTER 4

TESTING

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. It is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. A unit is the smallest testable part of an application.

In measuring semantic similarity in wordnet , each author can find accuracy by applying their own formula and it tested as individually.

Integration Testing

Integrating testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In semantic similarity, Michael lesk,leacock and chodorow,wu and palmer formula's are applied in main form to get the value from wordnet tool.

CHAPTER 5

CONCLUSION

Two possible implications of this project could be that the results are dependent on the characteristics of a test document and on the characteristics of glosses, which needs to be further investigated. However, the presented approach has several limitations: a small sample, and a big number of fine senses in WordNet, many of which are not that distinguishable from each other.

This project has presented a measure of semantic similarity in an is-a taxonomy based on the notation of Information content. Experimental evaluation was performed with a large independently constructed corpus. For the future enhancements semantic based search can be made and can be implemented in search engines using these algorithms.

APPENDIX

SAMPLE SCREEN DESIGN

The results of the project implementation are shown in the following snapshots.

A.1 HOME PAGE

The home page of the project implementation is shown in the Fig A.1

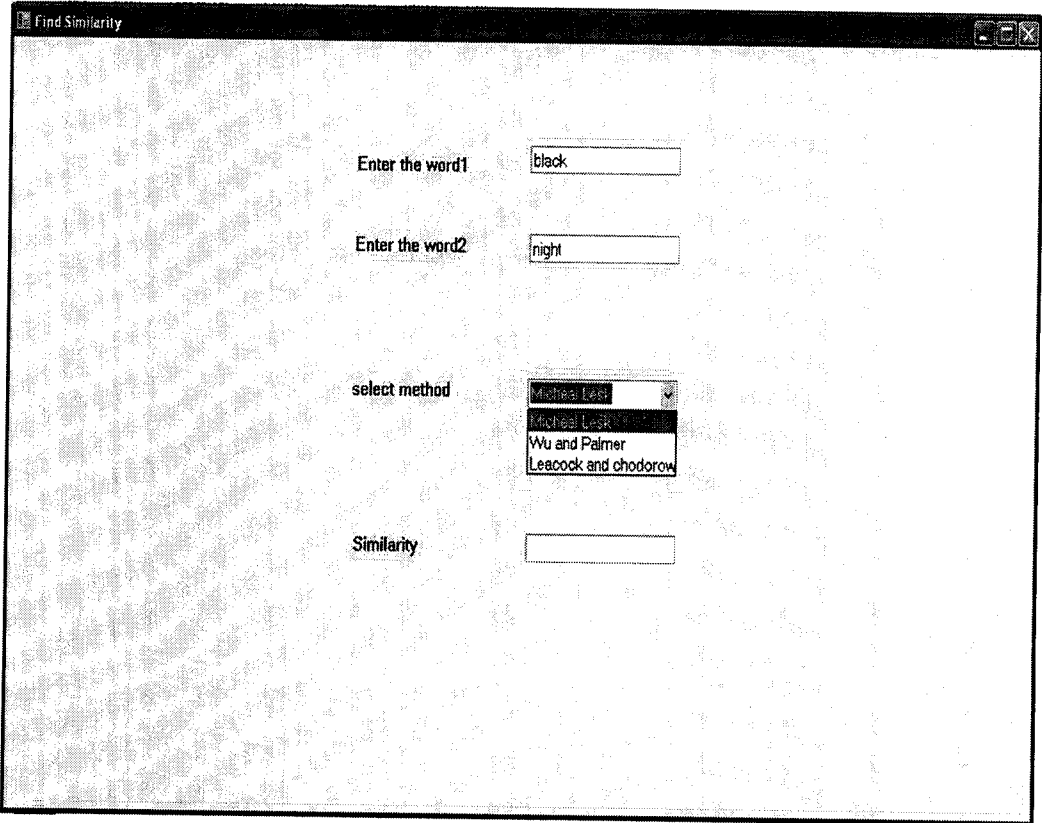


Fig A.1 Home page

A.2 MICHEAL LESK ALGORITHM

The similarity value between the words black and night using Micheal Lesk algorithm is shown in the Fig A.2

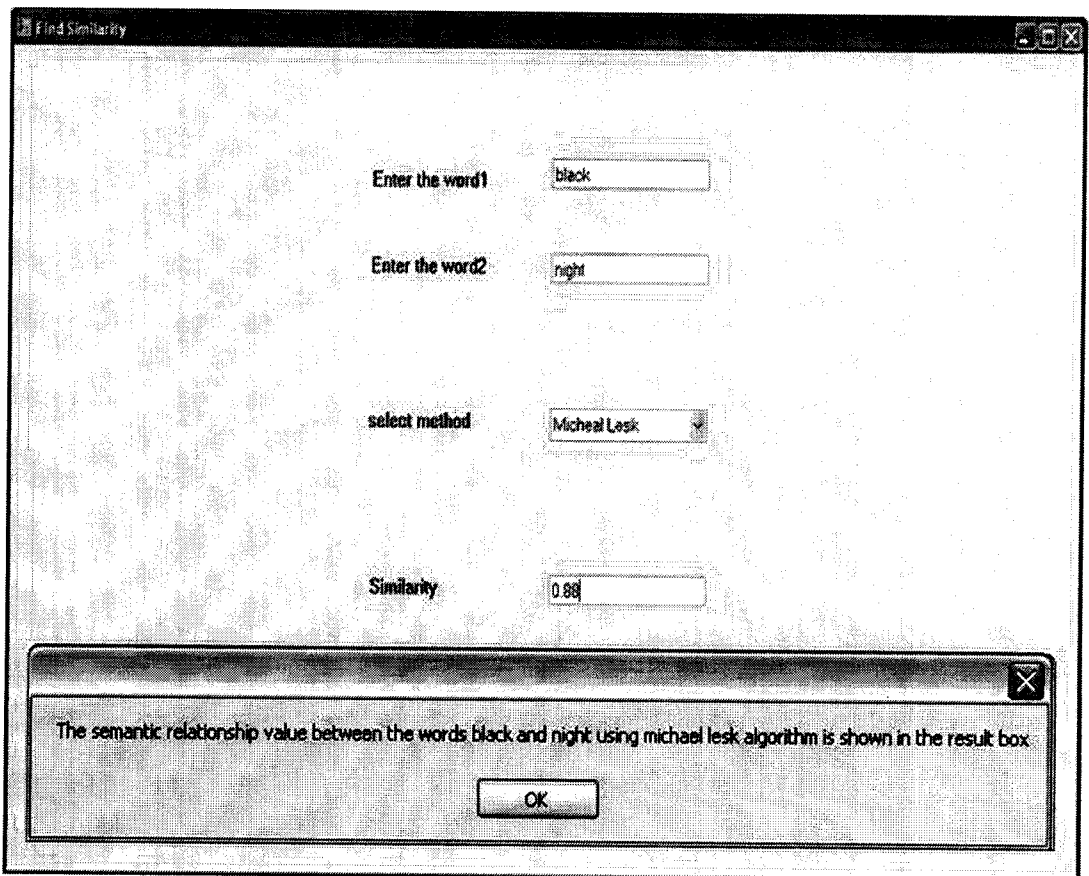


Fig A.2 Micheal Lesk algorithm

A.3. LEACOCK AND CHODOROW ALGORITHM

The similarity value between the words black and night using Leacock and Chodorow algorithm is shown in the Fig A.3

The screenshot shows a window titled "Find Similarity" with the following fields and values:

Enter the word1	black
Enter the word2	night
select method	Leacock and chodorow
Similarity	0.8721666

Below the main window, a message box displays the text: "The semantic relationship value between the words black and night using leacock & chodorow algorithm is shown in the result box." with an "OK" button.

Fig A.3 Leacock and Chodorow algorithm

A.4. WU AND PALMER ALGORITHM

The similarity value between the words black and night using Wu and Palmer algorithm is shown in the Fig A.4

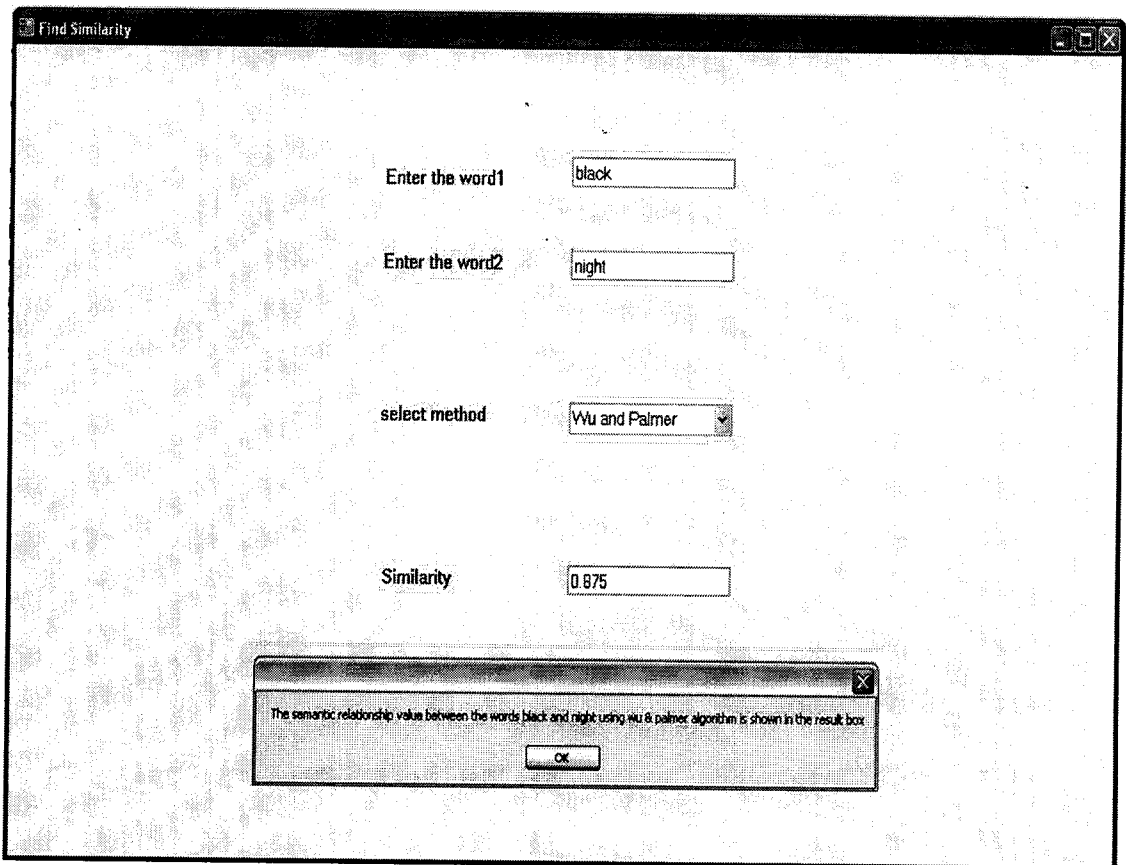


Fig A.4 Wu and Palmer algorithm

BIBLIOGRAPHY

- M. Lesk (1986) 'Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone',.
- D. Lin. (1998) 'Automatic retrieval and clustering of similar words'. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL '98), pages 768-774, Montreal, Canada.
- Resnik, P. (1995), 'Using Information Content to Evaluate semantic Similarity in a Taxonomy', Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol. 1, 448-453, Montreal..
- Wu, Z., & Palmer, M. (1994). 'Verb semantics and Lexical Selection'. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics Las Cruces, New Mexico.
- <http://talisker.d.umn.edu/similarity/measures.html>
- http://en.wikipedia.org/wiki/semantic_similarity
- <http://www.speech.sri.com/people/stolcke/papers/sri-h4-lm/node7.html>
- http://www.semantic-web-book.org/page/GeoS2009_Tutorial
- <http://rss.acs.unt.edu/Rdoc/library/maDB/html/distance.euclidian.html>
- <http://cat.inist.fr/?aModele=afficheN&cpsidt=16895029>
- <http://www.shiffman.net/teaching/a2z/wordnet/>
- <http://www.languageinindia.com/march2002/rajendran3.html>