THE ACQUISITION OF LEXICAL KNOWLEDGE FROM THE WEB FOR ASPECTS OF SEMANTIC INTERPRETATION

by

HANSEN A. SCHWARTZ
B.S. University of Central Florida, 2004
M.S. University of Central Florida, 2006

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2011

Major Professor:
Fernando Gomez

ABSTRACT

This work investigates the effective acquisition of lexical knowledge from the Web to perform semantic interpretation. The Web provides an unprecedented amount of natural language from which to gain knowledge useful for semantic interpretation. The knowledge acquired is described as common sense knowledge, information one uses in his or her daily life to understand language and perception. Novel approaches are presented for both the acquisition of this knowledge and use of the knowledge in semantic interpretation algorithms. The goal is to increase accuracy over other automatic semantic interpretation systems, and in turn enable stronger real world applications such as machine translation, advanced Web search, sentiment analysis, and question answering.

The major contributions of this dissertation consist of two methods of acquiring lexical knowledge from the Web, namely a database of common sense knowledge and Web selectors. The first method is a framework for acquiring a database of concept relationships. To acquire this knowledge, relationships between nouns are found on the Web and analyzed over WordNet using information-theory, producing information about concepts rather than ambiguous words. For the second contribution, words called Web selectors are retrieved which take the place of an instance of a target word in its local context. The selectors serve for the system to learn the types of concepts that the sense of a target word should be similar. Web selectors are acquired dynamically as part of a semantic interpretation algorithm, while the relationships in the database are useful to stand-alone programs. A final contribution of this dissertation concerns a novel semantic similarity measure and an evaluation of similarity and relatedness measures on tasks of concept similarity. Such tasks are useful when applying acquired knowledge to semantic interpretation.

Applications to word sense disambiguation, an aspect of semantic interpretation, are used to evaluate the contributions. Disambiguation systems which utilize semantically annotated training data are considered supervised. The algorithms of this dissertation are considered minimally-supervised; they do not require training data created by humans, though they may use human-created data sources. In the case of evaluating a database of common sense knowledge, integrating the knowledge into an existing minimally-supervised disambiguation system significantly improved results – a 20.5% error reduction. Similarly, the Web selectors disambiguation system, which acquires knowledge directly as part of the algorithm, achieved results comparable with top minimally-supervised systems, an F-score of 80.2% on a standard noun disambiguation task.

This work enables the study of many subsequent related tasks for improving semantic interpretation and its application to real-world technologies. Other aspects of semantic interpretation, such as semantic role labeling could utilize the same methods presented here for word sense disambiguation. As the Web continues to grow, the capabilities of the systems in this dissertation are expected to increase. Although the Web selectors system achieves great results, a study in this dissertation shows likely improvements from acquiring more data. Furthermore, the methods for acquiring a database of common sense knowledge could be applied in a more exhaustive fashion for other types of common sense knowledge. Finally, perhaps the greatest benefits from this work will come from the enabling of real world technologies that utilize semantic interpretation.

To my wife and family, who have each persevered and not given up through their own struggles which make the problems of this work seem much more attainable.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Fernando Gomez, for sharing his remarkable knowledge and wisdom as well as for his guidance of my research in a clear yet flexible direction. Additionally, I am very grateful to my thesis committee: Annie Wu, Valerie Sims, and Kenneth O. Stanley, who each provided a distinct and helpful perspective. I also appreciate the help and general academic advice from other professors and instructors: Steve Fiore, Charles Hughes, and Arup Guha. Additional discussions with colleagues in Artificial Intelligence helped to inspire progress and to keep the big picture in mind. These colleagues included but were not limited to Christopher Millward, Adam Campbell, Sean Szumlanski, Sameer Joshi, John Tanner, and Nadeem Mohsin. This work was supported in part by the NASA Engineering and Safety Center under Grant/Cooperative Agreement NNX08AJ98A and the ASRC Aerospace Corporation under Grant NAS1003006.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In computational linguistics, the study of meaning is referred to as semantics, and *semantic interpretation* is the process of determining meaning for an entire sentence (Allen, 1994). Algorithms that solve problems in *semantic interpretation* are considered enabling technologies (Resnik, 2006). They are not solutions to real world problems themselves, but they can enable stronger solutions for technologies such as *machine translation*, *information retrieval*, *dialog / spoken-language understanding*, and *question answering* (Ide & Véronis, 1998), as well as twenty-first century applications such as *sentiment/opinion analysis*, *accurate Web search*, and *social network mining*.

Semantics in general has been one of the major areas of study for the field of computational linguistics since its inception. Another major area, syntax, is concerned with studying formal relationships between words (Jurafsky & Martin, 2000). While algorithms performing tasks under the study of syntax, such as part of speech tagging or syntactic parsing, have reached accuracy levels like that of humans, algorithms for semantic interpretation have yet to do so on a broad scale. Tasks under semantic interpretation include *word sense disambiguation*, *semantic role labeling*, and *anaphora resolution* among others. It is widely believed that solutions to these problems lack accuracy due to a *data acquisition bottleneck*, in which systems are limited to correctly annotating sentences similar to those in which the system has been trained to interpret (Mihalcea, 2002; Diab, 2004; McCarthy et al., 2004; Swier & Stevenson, 2004; Gonzalo & Verdejo, 2006). The creation of training data takes many human hours and thus forms a bottleneck for what would otherwise be automatic algorithms.

As a solution to the bottleneck and achieving higher levels of *semantic interpretation* accuracy, this dissertation discusses the automatic acquisition of Common Sense Knowledge (*CSK*). This knowledge includes information we (as humans) use in our everyday life without necessarily being aware of it. Panton et al. (2006), of the Cyc project, define common sense as, "the knowledge that every person assumes his neighbors also possess." Essentially, it is a guideline for the type of knowledge needed to understand the meaning of common natural language. For example, in sentences (1), (2), and (3) the word 'key' has three different meanings. Those meanings could be summarized, respectively, by definitions from *WordNet* (Miller et al., 1993): "1. metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated", "2. something crucial for explaining", and "3. any of 24 major or minor diatonic scales that provide the tonal framework for a piece of music".

(1) *The key was in her pocket.*

(2) *The key was in her mind.*

(3) *The key was in E flat.*

CSK tells us that the type of key corresponding to the first meaning (sense 1), is something that is often kept in one's pocket, and thus the meaning of 'key' in (1) can be understood. For (2), one may utilize *CSK* that the second meaning of 'key' (sense 2) is an idea and that ideas are kept in the mind. Finally, in (3) the meaning of 'key' corresponds with the third meaning of key (sense

3); One may know that this sense of ‘key’ is represented by a letter often followed by ’sharp’ or ’flat’ even if one does not know exactly what that means for music. In general, *CSK* helps one understand the possible meanings or semantics of a sentence.

The primary advantage to using automatically acquired *CSK* is that human hours are not required for creating training data specific to the application. Many current semantic interpretation systems require training samples, example sentences annotated with meanings (semantic annotations) (Gildea & Jurafsky, 2002; Dang & Palmer, 2005; Chan et al., 2007). For example, a system attempting to determine the meaning of ‘key’ in examples (1) - (3) would need many examples of each sense of ‘key’ (and there are more than three senses of ‘key’). A secondary advantage of this automatic approach is that a single type of *CSK* may be used for multiple aspects of *semantic annotation*, where as methods involving training require annotations for each aspect. These advantages summarize the differences between *supervised* algorithms which require examples as training data, and *minimally-supervised* algorithms which do not use annotated examples, but may utilize other sources of information.

Until recently, an approach like that discussed in this dissertation would not be possible. It is the massive growth of the Web combined with an existing knowledge base, the WordNet ontology (Miller et al., 1993), that has enabled the automatic acquisition of *CSK* presented in this work. WordNet provides a mapping of word senses to concepts which are organized into an ontology. For example, WordNet is organized such that the concept corresponding to sense 1 of ‘key’ is a type of *artifact* (defined as “a man-made object taken as a whole”). Due to the massive size of the Web, an algorithm can find many results for something specific, such as the phrase “in a pocket”. Thus, it may be discovered through a Web search that other *artifacts* commonly appear “in a pocket”, such as ‘wallet’, ‘cell phone’, and ‘pencil’. This can provide a strong indication that a ‘key’ in a pocket corresponds to the sense that is an *artifact*.

When one determines the meaning of a word in a sentence, such as ‘key’, he or she is disambiguating among the senses of the word. When computers attempt this algorithmically, the problem is known as *word sense disambiguation* (*WSD*). Among the many aspects of semantic interpretation, *WSD* is focused on most heavily in this work. As will be discussed later in the dissertation, many of the other aspects of semantic interpretation are related to or can even be reduced to a specific type of *WSD*. Additionally, results for current top performing *WSD* systems are significantly below a human baseline and only slightly above a baseline of simply picking the most frequently occurring sense (Edmonds & Cotton, 2001; Snyder & Palmer, 2004; Navigli et al., 2007; Agirre et al., 2010). These ideas make *WSD* a worthwhile problem for evaluating the usefulness of the acquired *CSK*.

The dissertation is organized into 6 chapters, discussing both related works and the contributions of this dissertation. The first chapter, *Related Knowledge Acquisition and Semantic Interpretation*, presents background information and research works that are most related to the research of this dissertation. *Common sense knowledge* is discussed further and other sources of knowledge are presented in Section 2.1. Section 2.2 describes the aspects of *semantic interpretation* that this work is concerned with as well as a review of research studying the use of lexical knowledge for semantic interpretation. Section 2.3 discusses semantic similarity and relatedness measures that are used to compare concepts. These measures are essential for the algorithm applying *CSK* to semantic interpretation.

The major contributions of the research from this dissertation are described in Chapters 3 to 5. Chapter 3, *A Database of Applicable Common Sense Knowledge*, describes a type of *CSK* acquired as explicit relationships. The key novelties within this work include the use generic search phrases with automatically filled constituents to query the Web, and the incorporation a statistical parser to verify that the syntactic structure of results from the Web match an intended structure. After searching the web, the approach also includes an automatic analysis word relationships over WordNet in order to generalize information about concepts or areas of the ontology. Chapter 4, *Web Selectors as a Means to Dynamically Acquire Knowledge*, presents a second type of knowledge which is acquired specifically for a given sentence sentence (sections 4.1 and 4.3: *Web Selectors*). This approach to word sense disambiguation, which is also valid for other types of semantic interpretation, introduces the novel idea of acquiring selectors from the Web as well as the use of selectors in context. Selectors are words which take the place of a given target word within its local context. Finally, Chapter 5 introduces a novel semantic similarity metric as well as evaluates WordNet based semantic similarity and relatedness metrics on tasks of concept similarity, addressing the fact that an extensive evaluation of similarity and relatedness measures for the task of concept similarity had previously never been carried out.

Though the contributions are presented in separate chapters, they share ideas motivating their algorithms. During acquisition both approaches use *context* in a search for knowledge. Consider that one is trying to acquire knowledge about ‘pockets’. Searching just for the word ‘pocket’, results in millions of sample sentences from the Web. However, if the search included context, such as “the * was in her pocket”, the results would be greatly restricted and words returned in place of * may indicate things commonly in a pocket. Both types of *CSK* are validated through the semantic interpretation problem of *WSD*. Additionally, the similarity measure introduced is used within the Web selectors algorithm. The last chapter, numbered 6, brings all conclusions under the context of the ultimate goal to more effectively use knowledge acquired from the Web in problems of *semantic interpretation*.

1.1 Preliminaries

A few basic definitions and assumptions are necessary to fully understand this dissertation. A distinction is made between words and concepts, such that any word has a set of concepts which it could be referring to. In turn, a word sense maps to a concept, and a concept may have multiple word senses which map to it. For example, consider the noun ‘beam’ (words also have parts of speech), which has the seven senses listed below according to WordNet (Miller et al., 1993). Each sense of ‘beam’ refers to a concept described by the proceeding gloss. Additionally, many of the concepts have multiple word senses, such as that of *beam-3*, which also includes *ray-4* and *electron_beam-1*. To see this more clearly, examine Figure 1.1. Note that compound nouns are also considered single words when referring to one concept. Words which contain multiple senses, such as ‘beam’, are termed *polysemous*, while words with only have one sense are termed *monosemous*.

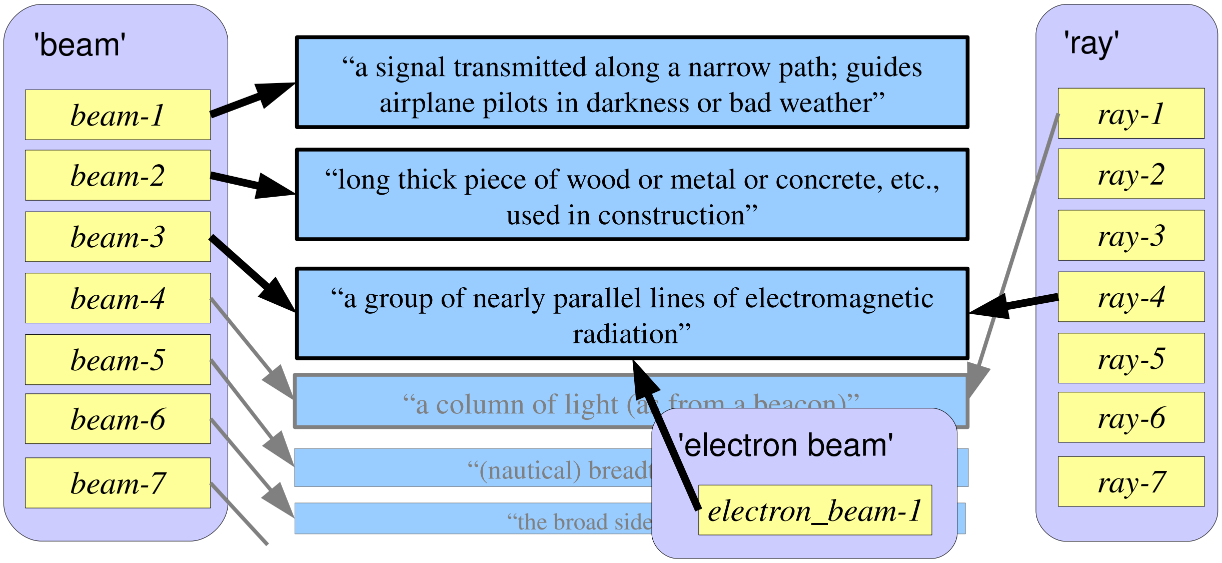

'beam'
beam-1
beam-2
beam-3
beam-4
beam-5
beam-6
beam-7
“a signal transmitted along a narrow path; guides airplane pilots in darkness or bad weather”
“long thick piece of wood or metal or concrete, etc., used in construction”
“a group of nearly parallel lines of electromagnetic radiation”
“a column of light (a
from a beacon)”
“(nautical) breadt
“the broad side
'electron beam'
electron_beam-1
'ray'
ray-1
ray-2
ray-3
ray-4
ray-5
ray-6
ray-7

Figure 1.1: Mapping noun senses of ‘beam’ to concepts, and other noun senses which map to the same concept as *beam-3*.

beam-1	(gloss: “a signal transmitted along a narrow path; guides airplane pilots in darkness or bad weather”)
beam-2	(gloss: “long thick piece of wood or metal or concrete, etc., used in construction”)
beam-3	(gloss: “a group of nearly parallel lines of electromagnetic radiation”)
beam-4	(gloss: “a column of light (as from a beacon)”)
beam-5	(gloss: “(nautical) breadth amid ships)
beam-6	(gloss: “the broad side of a ship”)
beam-7	(gloss: “a gymnastic apparatus used by women gymnasts”)

As was done above, single quotes will surround words: ‘beam’, and word senses (concepts) will appear in the italic form of a word followed by a dash and the sense number: *beam-3*. Many times a word sense will be given, with the intention of representing the concept for which it belongs.

Alternatively, a sense number is not always necessary for describing a concept, so one may also encounter a pattern exemplified here: *bat* (gloss: “a club used for hitting a ball in various games”); In this case it is the concept corresponding with *bat-5* that is being referred to, but it is more important that one understands the concept being referenced rather than the sense number. The glosses are quoted from WordNet (Miller et al., 1993) unless otherwise indicated.

Within this dissertation, the term *semantic interpretation* specifically refers to solving a set of problems in which lexical knowledge can play a part in the annotation of meaning. These problems include *word sense disambiguation* (*WSD*), *semantic role labeling* (*SRL*), *prepositional phrase attachment* (*PP attachment*), *anaphora resolution* (*AR*), and *named entity recognition* (*NER*) . Chapter 2.2 describes these problems and past work in more detail.

As final piece of preliminary information includes definitions for various levels human supervision within semantic interpretation algorithms. These include *supervised*, *minimally-supervised*, and *unsupervised*. *Supervised* specifically refers to algorithms that use a manually created training set of data in which the algorithm learns patterns for annotation (Màrquez et al., 2006). Supervised algorithms include standard machine learning classification algorithms such as support vector machines or maximum entropy learners. Algorithms which use manually created sources of information, but do not require hand-crafted examples as training data, are referred to as *minimally-supervised*. WordNet is a common source of information for minimally supervised algorithms, and one should note the key advantage of these algorithms over supervised algorithms is that there is no need for creating hand-tagged examples for each possible annotation. The term *unsupervised* is reserved for algorithms which do not use any sources of information created by humans other than the corpus with which the algorithm is being run. Unsupervised algorithms typically perform clustering rather than classification as they do not have a source identifying possible annotations (Pederson, 2006). Lastly, this dissertation also uses the term *knowledge-based* when describing algorithms that employ a knowledge source (Mihalcea, 2006). Knowledge-based systems may fall under any category of supervision, but they are typically considered *minimally-supervised*.

2 RELATED KNOWLEDGE ACQUISITION AND SEMANTIC INTERPRETATION

The acquisition of knowledge and its application to semantic interpretation has a long history. This chapter surveys that history broken down into three sections related to the contributions of this dissertation. The first section discusses work in acquiring or creating lexical knowledge sources in many different forms. Next, a discussion of approaches to semantic interpretation which utilize lexical knowledge is presented. Finally, related work in semantic similarity and relatedness metrics is also presented, as an aspect of this dissertation utilizes such metrics in the context of knowledge acquisition.

2.1 Lexical Knowledge Sources

Lexical knowledge is available in many different forms. This work is particularly interested in the subset that can be referred to as *common sense knowledge* (*CSK*). Panton et al. (2006) define common sense as, "the knowledge that every person assumes his neighbors also possess." Essentially, *CSK* is the knowledge we use in our every day life without necessarily being aware of it. It is *CSK* that tells us keys are kept in one's pocket and keys are used to open a door, but *CSK* does not hold that keys are kept in a kitchen sink or that keys are used to turn on a microwave, although all of these ideas are possible. Although the term common sense may be understood as a process such as reasoning, this dissertation is concerned only with the knowledge.

To show the usefulness of *CSK* for problems in computational linguistics, consider the following sentences.

(4) *He put the batter in the refrigerator.*

(5) *She ate the apple in the refrigerator.*

Example (4), deals with lexical ambiguity. There is little doubt for one to determine just what the "batter" is (food/substance used in baking). However, a computer must determine that it is not someone who swings a bat in baseball that is being put into a refrigerator, although it is entirely possible to do (depending on the size of the refrigerator). This demonstrates how *CSK* can be useful in solving *word sense disambiguation*. It is common for food to be found in a refrigerator and 'batter' is easily resolved as a food/substance rather than a person.

CSK can also help to solve syntactic ambiguity. The problem of *prepositional phrase attachment* occurs in sentences similar to example (5). In this case, it is difficult for a computer to determine if 'she' is in the refrigerator eating an apple or if the 'apple' which she ate was in the refrigerator. Like the previous example, the knowledge that food is commonly found in a refrigerator and people are not, leads one to understand that "in the refrigerator" should be attached to the noun phrase "the apple" and not as a modifier of the verb phrase "ate".

2.1.1 Standard Dictionaries

As one of the most prevalent sources of lexical knowledge, dictionary definitions (or glosses) may be regarded as *CSK*. These phrases may describe general knowledge that is important in understanding a language. Additionally, definitions are attached to word senses rather than word forms. In particular, noun glosses often have the form of a *superordinate* followed by *distinguishers*. *Superordinate* is a more general type of entity which the noun sense belongs and *distinguishers* describe unique characteristics of the concept (Miller et al., 1993). For example, consider (6), a gloss from the Longman Dictionary.

(6) trombone: *a metal musical instrument that you play by blowing into it and moving a long sliding tube.* (Summer & Gadsby, 2002)

One can easily determine that there is a relationship between 'trombone' (the *subordinate*) and 'metal musical instrument' (the *superordinate*). This relationship will be described in more detail below (see *hypernymy*). Additionally, the gloss is completed by mentioning distinguishers, 'play by blowing into it and moving a long sliding tube'. These distinguishing characteristics may also be *CSK*.

Standard dictionaries have several significant drawbacks when it comes to common sense. First, the scope of definitions certainly do not provide all necessary information (such as *keys are commonly kept in one's pocket*), and some definitions may be considered expert knowledge rather than *CSK*. Expert knowledge, such as "an amygdala is an almond shaped nuclei located in the cerebral cortex of the brain", is certainly not necessary to understand every day language.

Additionally, the type of knowledge or relationship being described is not made explicit. Consider (6); Although one can use the common pattern of glosses to determine the *superordinate* of ‘trombone’, ‘metal musical instrument’ does not appear in the dictionary itself and ‘instrument’ has 3 different senses according to Summer & Gadsby (2002); The noun tube has five different senses. Therefore, word senses or concepts involved in the relationship are not given. Additionally, the relationship described between ‘trombone’ and ‘tube’ is not clearly understood without the ability to understand language in the first place. In short, dictionaries require users (or computers) to be able to accurately process and understand language in order to retrieve all of the implicit knowledge. Still, these glosses have been used to determine general relatedness between concepts (Lesk, 1986), where relatedness strengths are given between a pair of words or concepts, for which no specific relationship is realized. Such work is described in section 2.3.4.

2.1.2 WordNet

Originally designed as a dictionary that could be searched conceptually, WordNet (Miller et al., 1993) has a unique feature of providing explicit relationships among concepts in what is considered an ontology. Concepts are represented as synsets, a list of word forms which are synonymous with each other, and thus have the same meaning. For example, {*batter, hitter, slugger, batsman*} is a synset with the gloss “(baseball) a ballplayer who is batting”. Below are two important relationships WordNet provides between noun concepts (synsets).

- hyponymy/hypernymy: *An x is a (kind of) y. ("ISA" relationship, x corresponds to the hyponym)*

- meronymy/holonymy: *x is a part of y. ("HASA" relationship, x corresponds to the meronym)*

Note that since these relationships are between concepts, x and y are words which belong to the synsets for which the relationships are defined over. For example, because one accepts "a batter is a baseball player", then {*batter, hitter, slugger, batsman*} is a hyponym of {*ballplayer, baseball player*}. In addition to relationships among concepts (synsets), WordNet also provides antonymy and morphological relations between word forms.

The WordNet noun ontology is designed such that distinguishing features of superordinates are inherited by their subordinates (Miller et al., 1993). Two of these features, *has attribute* (with an adjective) and *has function* (with a verb), are not explicitly given in WordNet, and they were only considered during the accurate creation of the ontology. *Meronymy* represents a third and final feature considered. This idea of inheritance is important, as it allows the inference of generalized knowledge. For example, if one finds many of the hyponyms of *seed-1* (gloss: "a small hard fruit") are commonly found in a jar, then one may be able to conclude that *commonly found in a jar* is a *CSK* feature for *seed-1*. See Figure 2.1 for the hyponyms of *seed-1*.

The importance of the WordNet noun ontology has already extended deep into the field of computational linguistics. Many evaluations of semantic interpretation problems, such as Senseval-2 (Edmonds & Cotton, 2001), Senseval-3 (Snyder & Palmer, 2004), and SemEval-2007 (Navigli et al., 2007), use WordNet as a sense inventory. All of the semantic similarity and relatedness measures that are discussed in Chapter 2.3 use WordNet to some degree, many relying solely on the

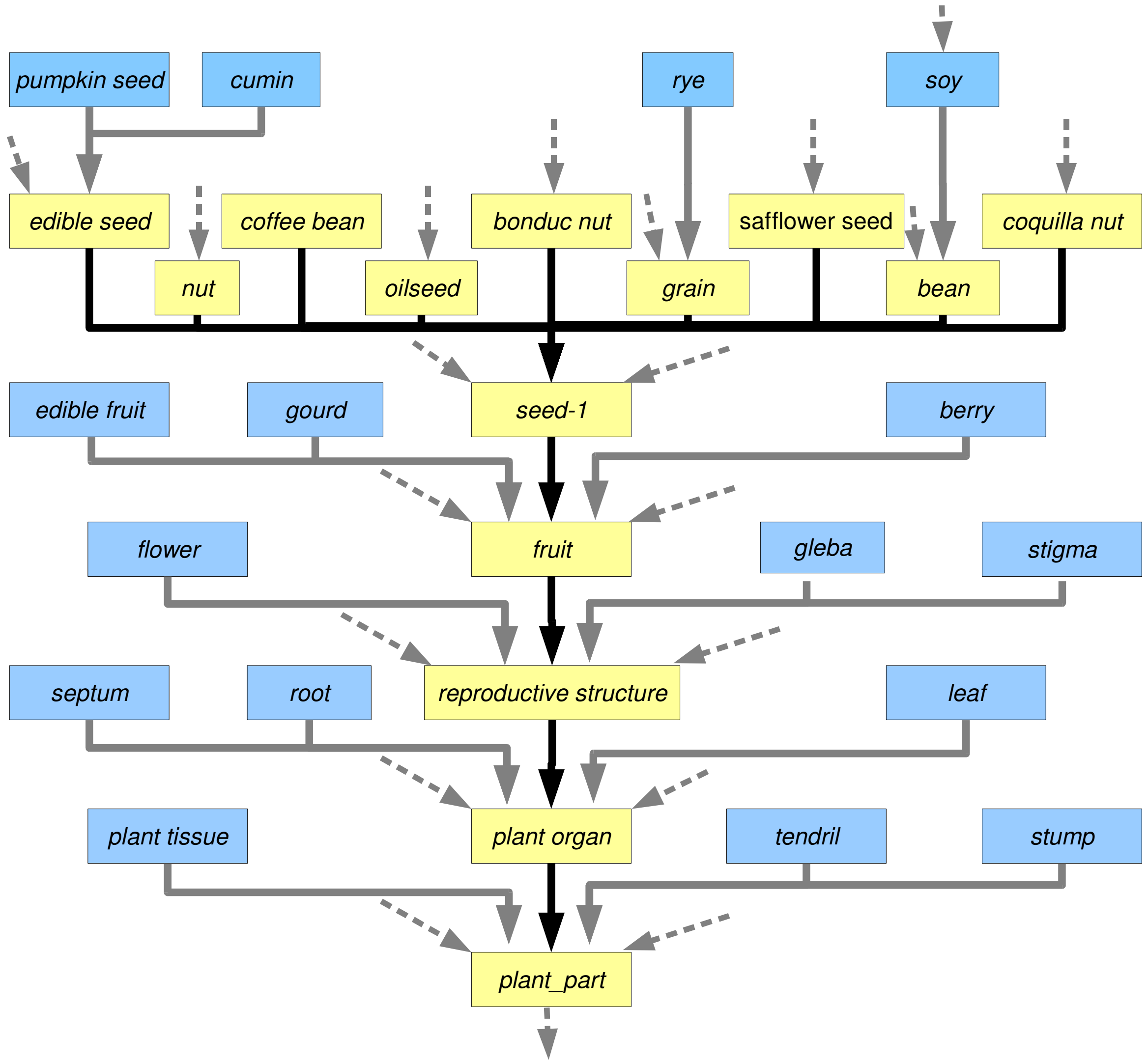


Figure 2.1: A depiction of the WordNet ontology surrounding the concept *seed-1*. Arrows point from *hyponym* to *hypernym*. Select hypernyms and direct hyponyms of *seed-1* are connected by darker lines. Each concept is represented by a word from its synset. Sense numbers are excluded for readability.

graph of the ontology. Additionally, many other sources of lexical knowledge, such as VerbOcean (Chklovski & Pantel, 2004) or the works of Mihalcea & Moldovan (1999), Agirre et al. (2001), Ruiz-Casado et al. (2007), and many others utilize the concepts and relationships of WordNet when creating knowledge.

2.1.3 Large Manually Constructed Knowledge-Bases

A project in progress for over twenty years, Cyc has been acquiring *common sense knowledge* about everyday objects and actions stored in axioms (Lenat, 1995; Panton et al., 2006). The axioms, handcrafted by workers at CYCcorp and now totaling over 3.5 million, represent knowledge rooted in propositions about 328,000 concepts (Curtis et al., 2006). There are three layers of information: the first two, *access* and *physical*, contain meta data, while the third, *logical* layer, stores high level implicit meanings. This representation is much more sophisticated than the relations acquired in this dissertation.

Although Cyc has many uses across the field of Artificial Intelligence, one particularly relevant study focused on its application to word sense disambiguation (Curtis et al., 2006). Documents are turned into *contextualized information structures*, which record information at the word, sentence, and paragraph level. Based on these structures and Cyc's knowledge-base, a concept in Cyc is then chosen as a representative of the sense. Overall, their method performs at 56.7% accuracy on a selection of ambiguous words from Wikipedia sentences. This is higher than a 33.5% random baseline, but other points of comparison are not supplied, and the polysemy of words in Cyc versus standard sense inventories is not provided.

ConceptNet was created based on the OpenMind Commonsense project which utilized an interface on the Web in order to acquire knowledge (Liu & Singh, 2004). Users played games and answered questions about words in order to determine a wide range of relations. In the end, ConceptNet only provides relationships between word forms, an interesting fact considering WordNet actually provides relations between concepts.

As with any handcrafted dataset, many hours must be put into the curation of data. The automatic approach to acquiring common sense presented in this dissertation avoids this curation time. Additionally, only a portion of Cyc is available to the public and its concepts do not map to an existing word sense inventory, so it is difficult to evaluate the knowledge-base in comparison to other sources for problems of lexical semantics.

2.1.4 Web-based knowledge acquisition

The size of the Web is unprecedented when compared to other corpora. It has enabled the success of many systems designed for acquiring knowledge from large sets of text. Hearst (1998) noted that many potentially useful lexical relations, which were missing from WordNet, could be acquired automatically. Earlier, she introduced the idea of using manually built search patterns to find knowledge in large corpora (Hearst, 1992). Search patterns are a big part of the work described in this dissertation as well as a part of nearly all work using the Web as a source for lexical knowledge.

An example pattern for acquiring a hypernymic relationship: *concept1 is-a concept2* is given as example (7).

(7) *concept2* such as *concept1*

In the work presented in this dissertation, manually constructed *search phrases*, or abstract patterns, are used to automatically generate more specific *web queries* by filling constituents based on lists of words. Lexical knowledge is then found through matching *web queries* to text on the web. This process is discussed extensively in Chapter 3. Below, we discuss and compare methods which utilize the Web or other massive corpora for knowledge acquisition.

A recent trend has been the use of *seed* data or relationships in order for contextual patterns to be extracted, which indicate a relationship (Riloff & Shepherd, 1997; Ravichandran & Hovy, 2002; Thelen & Riloff, 2002; Girju et al., 2003; Pantel & Pennacchiotti, 2006; Pasca et al., 2006). Large corpora are then searched for other instances of the extracted patterns, expanding the set of relationships beyond the initial seeds. Figure 2.2 shows the process along with an example for finding hypernymic relationships with ‘trombone’ and ’brass instrument’ as the example seed relationship. In turn, one would find “The trombone (a brass instrument) played ..”, or “Brass instruments, such as trombones, are commonly found ...”. From these phrases, patterns describing hypernymic relationships can be extracted: “*concept1 (a concept2)*” and “*concept2, such as concept1*”, where *concept1* and *concept2* are wildcards matching members of the relationship. Finally, more knowledge is found by searching with these patterns, and the process repeats by then searching for more patterns with the acquired knowledge. Because of this repetitive process, the idea is commonly referred to as *bootstrapping knowledge*, a reference to pulling oneself up by one’s own bootstraps.

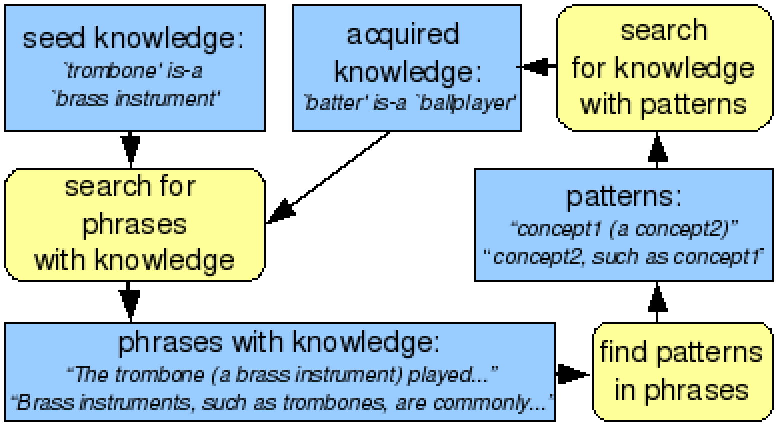


Figure 2.2: The generic process of using seed knowledge to find patterns for acquiring knowledge.

Early work in *bootstrapping knowledge* automatically generated semantic lexicons, lists of words associated with a concept, when given a few example lemmas for a concept (Riloff & Shepherd, 1997; Thelen & Riloff, 2002). Similarly, Ravichandran & Hovy (2002) used the seed approach to generate patterns describing birth years, inventors, discoverers, definitions, why someone was famous, and locations. They applied it successfully in question answering. Ruiz-Casado et al. (2007) presented an automatic approach where seeds were taken from relationships in WordNet, and patterns describing their relationship were discovered over Wikipedia. They were limited to patterns of hypernymy, hyponymy, holonymy, and meronymy, since those were the only ones covered in WordNet. This approach has also been used to show that *CSK* is often explicitly stated in texts Yu & Chen (2010). The advantage of these bootstrapping approaches (in the basic sense) is that only a minimal amount of supervision is needed, that of providing seed relationships. The drawback is that the relationships are less reliable.

To address this reliability drawback, several approaches have been taken to improve automatically acquired patterns. Pasca et al. (2006) developed an algorithm to score and rank both the generated patterns and knowledge acquired in order to produce a precision of 90% on facts about birth years. The Espresso system (Pantel & Pennacchiotti, 2006) took a different approach to improving reliability. It works with noisy broad coverage patterns (referred to as *generic patterns*) by validating them with more precise *reliable patterns*. These reliable patterns are typically longer and known to produce accurate results. Girju et al. (2003) used semantic constraints based on WordNet concepts to remove incorrect instances. These constraints were discovered through a decision tree algorithm on an annotated corpus, such that members of a relationship were expected to belong to certain regions of the WordNet ontology. Most recently, the bootstrapping approach has benefited from coupled-learning, in which multiple constraints are learned at the same time (Carlson et al., 2010). The idea is that one is able to reduce error by learning jointly with other independent functions and outputs rather than a single independent function.

In this work, reliability is important when focused on acquiring knowledge for its application to semantic interpretation. Although much has been done to improve accuracy of approaches based on seed knowledge, they still introduce more complexity and issues such as semantic drift (Komachi et al., 2008). Systems which rely more heavily on manual patterns can focus more on reliability since there is less room for errors to be introduced (there is no step to acquire patterns). These approaches vary quite a bit, but mostly reduce to the idea of using manual patterns presented in (Hearst, 1992).

Agirre et al. (2001) used the web to acquire topic signatures. These signatures were lists of words associated with a concept much like those generated in (Riloff & Shepherd, 1997), but they also include edge weights for the associations. The topic signatures were built by collecting documents related to a concept in WordNet from the WWW using the method of Mihalcea &

Moldovan (1999), which was originally used to create sense tagged corpora (this is discussed more in section 2.2.2). Although Agirre et al. state that topic signatures are not focused on a single application, they were able validate the knowledge through successful application to *word sense disambiguation*. Later, they used a similar approach to automatically acquire example usages for all noun senses in WordNet, and were able to successfully apply the examples to supervised word sense disambiguation (Martinez et al., 2008)

The work on VerbOcean used patterns of phrases in order to search the Web for semantic relations among verbs (Chklovski & Pantel, 2004). The relationships, *similarity*, *strength*, *antonymy*, *enablement*, and *happens-before* included weights (strength of relationship). This knowledge falls into the category of *CSK*, but the specific relationships they acquired were among verb word forms and senses are not resolved. It has been noted that word senses or concepts enable applications more readily than ambiguous words (Pantel & Lin, 2002); Knowledge about noun senses or concepts is a key attribute of our work.

Wikipedia is an attractive source for knowledge acquisition because it is more structured than the Web itself (Ponzetto & Strube, 2007). Gabrilovich & Markovitch (2009) present *Explicit Semantic Analysis* which interprets unannotated corpora through meaning found in Wikipedia concepts rather than direct definitions. Their method improved over the state of the art in text categorization and fragment semantic relatedness. Other works using Wikipedia focused more strictly on semantic relatedness, such as Ponzetto & Strube (2007), who found results in line with human judgments and successfully used their relatedness information for coreference resolution. Additionally, Navigli & Ponzetto (2010) map Wikipedia articles to WordNet concepts and use hyperlinks in Wikipedia to determine relatedness. Szumlanski & Gomez (2010) also use Wikipedia to determine relatedness between WordNet concepts, but they do not rely on the hyperlinks provided in Wikipedia. It is important to note the distinction between works that used structure in Wikipedia, such as that of Ponzetto & Strube; Gabrilovich & Markovitch; Navigli & Ponzetto, and those who only treat Wikipedia as a reliable corpus such as Szumlanski & Gomez. Similar to our work, all of these methods lend themselves to gathering knowledge for concepts rather than words (either a coarse Wikipedia article as a concept or WordNet concepts). On the other hand, the type of knowledge in Wikipedia articles does not always include *CSK*, a thought Gabrilovich and Markovitch acknowledge. For example, the current Wikipedia article on 'keys' mentions nothing about them often being found in one's pocket. Additionally, while Wikipedia is a degree of magnitude larger than a standard dictionary (Gabrilovich & Markovitch, 2009), the Web as a whole is at least a degree of magnitude larger than Wikipedia.

Several SemEval tasks present a good overview of work in noun-noun relationships: SemEval-2007 Task 4: *Classification of Semantic Relations between Nominals* (Girju et al., 2007) and SemEval-2010 Task 8: *Multi-Way Classification of Semantic Relations between Pairs of Nominals* (Hendrickx et al., 2010). Our work is related in that the relationships we acquire are between nominals, though we use an analysis to turn nominal word relationship information into concept relationship information. Additionally, in order to build their corpus Girju et al. queried the web with patterns like that of Hearst's work Hearst (1992). The tasks followed (Girju et al., 2003, 2006) in which a system was trained on positive and negative examples of meronymic (part-whole) relationships in order to classify additional examples of the relationships. These works are concerned with classifying relationships rather than the acquisition or application of relationships. Similar work has characterized relationships in a minimally-supervised fashion in which the Web is searched for verbs, prepositions or coordinating conjunctions connecting noun pairs (Nakov & Hearst, 2008). Note that the relationship classes within these works are not always within the the scope of *common sense knowledge*.

A few other works explore the acquisition of relations from varying points of view. Turney (2008) introduces Latent Relation Mapping Engine (*LRME*). Combining Structure Mapping Theory and Latent Relation Analysis, *LRME* builds mappings between words based on analogy understood through relational similarity and predicate logic. Lapata & Lascarides (2006) use markers such as “after” or “while” to infer temporal relations. Similar to our approach, these works look to leverage vast amounts of unannotated corpora to avoid hand-coding representations. Another approach acquires knowledge represented in propositional form directly from sentence parses (Schubert, 2002). The idea was introduced by Schubert, using gold-standard parses from the Treebank corpora to derive the propositions (Schubert & Tong, 2003). Later Clark and Harrison used an automatic parser to scale the approach to larger amounts of text (Clark & Harrison, 2009). The propositional knowledge is stored in tuples created directly from parses of sentences by matching a fixed set of structures. For example, the system might pull out the noun-verb-noun in the sentence “The men ate the apples” to derive the proposition *(NVN ”men” ”eat” ”apples”)*, interpreted as “men can eat apples”. Comparably, Yates & Etzioni (2009) extract assertions in the form of tuples and used for synonym resolution. Motivated through information extraction, their method creates knowledge of the form *(relation, arg1, arg2)* where *arg1* and *arg2* are typically named entities. Much like other work, these methods gather information regarding ambiguous words rather than concepts, though Yates & Etzioni mention the possibility of improvement through handling polysemous named entities.

2.2 Semantic Interpretation Utilizing the Web

This chapter presents related research in using the Web to aid in various problems of semantic interpretation. While *Common Sense Knowledge* (*CSK*) is a general guideline for the type of knowledge being acquired, the successful application of the knowledge to semantic interpretation is the ultimate goal of this work. In general, 'semantic' refers to meaning, such as defining what a word means in a sentence. When one tries to determine the meaning of an entire sentence (without considering the context of surrounding sentences), this is referred to as *semantic interpretation* (Allen, 1994). For this dissertation, the term *semantic interpretation* specifically refers to solving a set of problems, listed in Table 2.1, in which *CSK* can play a part in the annotation of meaning.

Table 2.1: Abbreviations for problems considered *semantic interpretation* in this work.

WSD	*word sense disambiguation*
SRL	*semantic role labeling*
PP attachment	*prepositional phrase attachment*
AR	*anaphora resolution*
NER	*named entity recognition*

WSD, in particular, is focused on most heavily in this paper. The goal of this problem is to choose the correct sense of a word when given a context (typically a sentence) in which the word is used; That is, the meaning of a word is determined by its context. Consider the word ‘port’ in the sentences (8) and (9).

(8) *They make port from grapes in Portugal .*

(9) *They bought the grapes at the port in Portugal.*

In (8) ‘port’ is referring to a *port-2* (gloss: “sweet dark-red dessert wine originally from Portugal”) while in (9) ‘port’ is *port-1* (gloss: “a place (seaport or airport) where people and merchandise can enter or leave a country”). One can determine these meanings based on how ‘port’ is being used (its context). In these examples, the difference in context is subtle as both sentences contain many of the same words (the exact same nouns). Additionally, both senses of ‘port’ could be associated with the verbs ‘make’ and ‘bought’. This demonstrates an important point, that solving *WSD* requires more that just considering which words occur in the sentence.

Semantic roles, also known as *thematic roles* (Allen, 1994), characterize the arguments of verbs (Jurafsky & Martin, 2000). They were first introduced as *deep roles* by Fillmore (1968). Looking back at the *WSD* examples, when labeling arguments of ‘make’ in (8), one could say ‘they’ is characterized as the *agent* (cause of action), ‘port’ is the *theme* (directly experiences the action), and ’from grapes’ could be labeled *of-stuff* (the thing something is made of (Gomez, 2001)). The prepositional phrase, ‘in Portugal’, would not receive a role as it is an adjunct, and not necessarily an argument of the verb. Generally, *agent* and *theme* occur with most verbs while other roles, such as *of-stuff* are only used to characterize specific verb senses. Many times *SRL* is associating with determining *verbal predicates* (the meaning of a verb dependent on its roles). However, this dissertation uses the notion that a verbal predicate includes a verb sense (Grimshaw, 1990; Gomez, 2004a). Thus, *verbal predicate labeling* is not included as a separate task, since it is covered by verb *WSD*.

For *PP attachment*, one is concerned with deciding which word or phrase is being modified by a prepositional phrase. Although this may be seen as a syntactic problem, it is included under the domain of semantic interpretation as there is a requirement for semantic knowledge in many instances. Consider sentences (10) and (11).

(10) *She ate the grapes in the refrigerator.*

(11) *She ate the grapes at home.*

In (10) the *PP* 'in the refrigerator' is attached to 'the grapes' (The grapes, which she ate, were in the refrigerator). However, in (11) the *PP* 'at home' is attached to 'ate', implying she was at home while eating the grapes.

Anaphora occurs when a word references an entity that was previously introduced in the sentence or corpus (Jurafsky & Martin, 2000). *AR* is the process of determining what entity (also called the *antecedent*) is being referenced. Many times pronouns are doing the referencing, such as in sentences (12) and (13).

(12) *After Hank hit the ball with the bat, it traveled far.*

(13) *After Hank hit the ball, he ran to first base.*

The pronoun 'it' in (12) is referencing 'the ball' (the antecedent), while in (13) 'Hank' is the antecedent of 'he'.

Like *AR*, *NER* is concerned with resolving entities represented by a word. However, in this case, proper names are being labeled with classes rather than a previously identified entity. These named entity classes include *person*, *location*, *organization*, and a class indicating named entities which are not a part of the other three classes: *miscellaneous* (Tjong Kim Sang & De Meulder, 2003). In (8) and (9) 'Portugal' is a named entity which could be labeled as *location* while in (12) and (13) 'Hank' would be labeled as *person*.

All of these problems are highly related. It is suggested that accurate semantic role labels follow from accurate verb sense disambiguation (Schwartz et al., 2008). From the another direction, Dang & Palmer (2005) found semantic roles helped to disambiguate verbs. *PP attachment* can be used to determine whether something should receive a semantic role. *AR* inherently relies on the meaning of other words in the sentence, and *NER* can be generalized as *WSD*, where the possible senses are always one of the 4 classes.

This chapter discusses many related works in using the Web for various semantic interpretation problems. Although this work proposes utilizing the Web for other problems in Semantic Interpretation, *WSD* motivates the algorithms, and thus most of the related works presented in this chapter perform *WSD*. The Web is referred to only because it is the largest set of text widely available. Many of the methods discussed are flexible enough to work with any large unannotated corpus, and they are considered unsupervised or minimally supervised since they do not require hand-tagged training data. Before discussing these methods which use the Web, some traditional knowledge-based semantic interpretation methods are presented (section 2.2.1). The chapter then proceeds by describing semantic interpretation methods which employ the use of the Web for acquiring training samples (section 2.2.2), and then by focusing on systems which use the Web directly in an algorithm (section 2.2.3).

2.2.1 Knowledge-Based Semantic Interpretation

Many algorithms for semantic interpretation have incorporated knowledge-bases as a main source for annotation. Although knowledge-based methods are often considered unsupervised (Yarowsky, 1995), “minimally-supervised” may be a more appropriate term since human created data sources are used. In fact, truely unsupervised methods are distinguished from those in this section in that they do not even rely on human supervision in choosing classes for annotating meaning (they typically perform clustering instead of annotation (Pederson, 2006)). On the other hand, supervised approaches are also distinguished in their use of an annotated corpus in a supervised machine learning algorithm (Màrquez et al., 2006). The term *knowledge-based* is used, following Mihalcea (2006), since all of the methods in this section require some source of knowledge.

Some of the most effective *knowledge-based WSD* methods use simple heuristics based on frequency statistics. They employ variations on an early idea that there is one sense of a word that dominates the others (Mihalcea, 2006, “Zipfian distribution”). Both Gale et al. (1992) and Yarowsky (1995) use this idea as part of a *one sense per discourse* algorithm to disambiguate among words with two possible senses. This idea is that a word (with only two senses) will have same sense for multiple instances within a single text (Gale et al., 1992). Yarowsky (1995) also used *one sense per collocation*, assuming a word maintains meaning when occurring around the same words, and a *bootstrapping* technique discussed in the next section. They were able to achieve 96.5% noun *WSD* accuracy. However, Among more polysemous words (with more than 2 senses), Krovetz (1998) found the *one sense per discourse* did not hold up as 33% of the discourses in Semcor and the DSO have words with multiple senses. Additionally, the frequency information used by these methods requires sense tagged data, which may not be available for a given language or discourse.

McCarthy et al. (2004) addressed the requirement for sense tagged data by creating a method to estimate most frequent sense information without tagged senses. The estimated most frequent sense was called the 'predominant sense'. This method used WordNet similarity measures (discussed in Chapter 2.3) between senses of nouns and dependency neighbors data acquired through the method of Lin (1998a). Essentially the sense of a word most similar to all of the neighbors was considered the predominant sense.

Ultimately, methods which rely only on frequency information within a discourse are held back by the idea that infrequent senses do in fact occur from time to time in the same discourses or even with the same word collocations as frequent senses; See example (14) in which two senses of 'port' appear within the same discourse and collocation (a single sentence).

(14) *She bought the port at the port.*

McCarthy et al. (2004) notes "we do not assume that the predominant sense is a method of WSD in itself". However, predictions of the most frequent sense (*MFS*) over a language commonly serve as a baseline for *WSD* performance evaluations. The MFS baseline is not often overcome;

```
(Clause CL68
    (SUBJ : ((NOUN SURFACE) (NOUN TENSION)
        PHYSICAL_PHENOMENON1 SURFACE_TENSION1 [INANIMATE-CAUSE] )
    (VERB : DRAW ((AUX (WILL)) (MAIN-VERB DRAW DRAW)) <DRAW-FLUID:DRAW8> )
    (OBJ : ((NOUN LIQUID)) LIQUID LIQUID1 [THEME] )
    (PREP : INTO (PREP-NP: ((UDT A) (NOUN CAPILLARY)) TUBE1 CAPILLARY1 [GOAL] ) ) )
```

Figure 2.3: Output of Gomez's Semantic interpreter for the sentence: *Surface tension will draw liquid into a capillary*.

Consider that only 5 of 13 systems that participated in the SemEval-2007 Coarse-Grained English All-Words Task (Navigli et al., 2007) surpassed the baseline.

Gomez's work in semantic interpretation (Gomez, 2001), took a much different approach. Rather than using statistical information, it is based on hand-crafted verbal predicates which contain *selectional restrictions* for semantic roles of the verb (Gomez, 2004a). An example output of the system is given in Figure 2.3. Among the aspects of semantic interpretation we have discussed, this algorithm includes semantic role labeling (maked as '[role]'), verb sense disambiguation along with labeled predicates (marked as '<predicate:sense>'), and noun sense disambiguation (marked with 'senses') for the roles of a verb argument. *Selectional restrictions* are concepts in an enhanced WordNet (Gomez, 2004b), with which arguments of the verb must belong. In Figure 2.3, a selectional restriction for the predicate '<DRAW-FLUID:DRAW8>' may be that the object is a *fluid* (gloss: "continuous amorphous matter that tends to flow and to conform to the outline of its container: a liquid or a gas"). *Liquid-1* is a hyponym of *fluid*, and thus fits the selectional restriction for the predicate corresponding to the verb sense *draw-8*.

Selectional preferences, like *selectional restrictions*, restrict the semantic class with which a word can belong. However, *selection preferences* usually refer to a more general constraint on the meaning of a word in a given context (Mihalcea, 2006). Agirre & Martínez (2001) evaluated a variety of types of selectional preferences in a word sense disambiguation task. The different types can be summarized as *word-word*, *word-class*, and *class-class*, describing to the item in context restricting the meaning of a focused item. For example, *word-class* refers to the idea that a word in context restricts the class to which a focus word belongs. The results showed *class-class* methods perform best overall, while *word-class* methods still achieve a high precision at the expense of recall. However, when Resnik (1997) introduced *word-class* preferences, a major motivation was the lack of class-annotated data required, where as Agirre & Martínez (2001) required semantically annotated text for their *class-class* preferences. The *word-word* preferences perform considerably worse, which is expected since the knowledge contains a lot of ambiguity.

One of the earliest approaches to knowledge based *WSD* was the Lesk algorithm (Lesk, 1986). The algorithm examines a dictionary in order to perform disambiguation by maximizing the number of words in common between definitions. The sense which has the definition with the most words in common with the definitions of other words in the sentences is selected. The original algorithm attempted to find the maximum combination of senses, so if a sentence had one verb and three nouns, it would decide on all the senses at once. A simplified version was found to perform just as strong by considering only one word at a time (Vasilescu et al., 2004). Consider the disambiguation of the word ‘steamer’ in example (15) and the subsequent definitions of ‘steamer’ (16) and ‘port’ (17).

(15) *He walked along the port of the steamer.*

(16) *steamer-1* (gloss: “a clam that is usually steamed in the shell”)

steamer-2 (gloss: “a cooking utensil that can be used to cook food by steaming it”)

steamer-3 (gloss: “a ship powered by one or more steam engines”)

steamer-4 (gloss: “an edible clam with thin oval-shaped shell found in coastal regions of the United States and Europe”)

(17) *port-1* (gloss: “a place (seaport or airport) where people and merchandise can enter or leave a country”)

port-2 (gloss: “sweet dark-red dessert wine originally from Portugal”)

port-3 (gloss: “an opening (in a wall or ship or armored vehicle) for firing through”)

port-4 (gloss: “the left side of a ship or aircraft to someone who is aboard and facing the bow or nose”)

port-5 (gloss: “(computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals”)

Steamer-2 contains the word ‘ship’, which also occurs in a couple definitions of ‘port’. Since this sense of ‘steamer’ has the most words in common among the definitions, it is selected (assuming definitions of ‘walk’ do not have any words in common). A relatedness measure derived from this algorithm is discussed in section 2.3.4.

Another class of knowledge-based algorithms employs the use of semantic relationships among concepts, conceptualized as a graph. The idea behind graph-based approaches is to use graph analysis metrics in order to identify the concept (or sense of a word) most connected to other words in context. The graph analysis techniques may vary quite a bit. Navigli & Velardi (2005) presented *structural semantic interconnects* (*SSI*), graphs of relationships based on WordNet, domain labels, annotated corpora, and collocation dictionaries. The resulting graph contains all the relationships in WordNet as well as a *domain* relationship among synsets, *co-occurrence* (among word senses) relationships extracted from annotated corpora, and *collocation* (among words) from the dictionaries. After first noting monosemous words, *SSI* is explored for links between the disambiguated words and ambiguous concepts, disambiguating the ambiguous word with the most connectivity and then repeating.

Other graph based measures create weighted graphs based on a single sentence or set of context words (Sinha & Mihalcea, 2007; Agirre & Soroa, 2009). The weights in the graph may come from aggregating the number of links or from similarity and relatedness measures (discussed in Section 2.3) over WordNet. Sinha & Mihalcea (2007) use a *PageRank* metric as well as three other metrics that rely on counting links directly (*indegree*, *closeness*, and *betweeness*). The *PageRank* method is appealing since it considers the importance of all vertices (concepts) that are involved in the graph. Those vertices with more connection receive more importance and thus have more influence on the final sense prediction. Agirre & Soroa (2009) use a *personalized PageRank* which helps to avoid a bias for certain vertices. They find the *personalized PageRank* metric to outperform the standard *PageRank* metric. However, both Sinha & Mihalcea (2007) and Agirre & Soroa (2009) were able to achieve the highest, though comparable to each other, results for any unsupervised system on the senseval-2 (Edmonds & Cotton, 2001) and senseval-3 (Snyder & Palmer, 2004) all-words datasets.

Although these methods have found knowledge to be helpful for problems of semantic interpretation, there are some drawbacks. Those with manually constructed data require many human hours to create the annotated data or knowledge bases (the *data acquisition bottleneck* mentioned in the Introduction). The heuristic approaches, such as *one sense per discourse* or the standard *MFS* baseline, are guaranteed to be incorrect when one word appears with multiple senses in the same discourse or sentence. The vast size of the Web presents a great source in order to acquire a large knowledge base or training set automatically. The following sections look into how this has already been done.

In fact, truely unsupervised methods are distinguished from those in this section in that they do not even rely on human supervision in choosing classes for annotating meaning (they typically perform clustering instead of annotation (Pederson, 2006)). On the other hand, supervised approaches are also distinguished in their use of an annotated corpus in a supervised machine learning algorithm (Màrquez et al., 2006). A final term *knowledge-based*, could be used to describe both systems with any level of *supervision* which require a knowledge source (Mihalcea, 2006). Annotated training data is not considered a knowledge source, so system may be *supervised* by not *knowledge-based*.

2.2.2 Acquiring Training Samples from the Web

One reason the *data acquisition bottleneck* exists is because of the human hours needed to produce semantically annotated text. Mihalcea & Moldovan (1999) addressed this by automatically

```
Find all synonyms for a given word sense.
    $synset(\textit{batter-1}) = \{$‘batter’, ‘hitter’, ‘slugger’, ‘batsman’$\}$
Find monosemous synonyms.
    $monosemous(synset(\textit{batter-1})) = \{$‘batsman’$\}$
Search the Web for instances of monosemous synonyms.
    $webexamples = websearch(monosemous(synset(batter - 1))) = \{$
            “The batsman approached the plate.”,
            “The Yankees need to recruit another batsman if they want to beat the White Sox.”,
            “Some say Babe Ruth was the best batsman to grace the field.”,
            ...$\}$
Turn the examples from the Web into training example for the given word sense.
    $annotate(\textit{batter-1}, webexamples) = \{$
            “The batter-1 approached the plate.”,
            “The Yankees need to recruit another batter-1 if they want to beat the White Sox.”,
            “Some say Babe Ruth was the best batter-1 to grace the field.”,
            ...$\}$
```

Figure 2.4: An example of using a monosemous synonym of *batter-1* to extract training examples.

generating sense tagged corpora from the Web. They used WordNet synsets and glosses in order to find lexical phrases unique to the sense of a word. For example, the algorithm would search the Web for instances of a word, w, with a context matching a pre-chosen lexical phrase for a sense, i, of w. The results would in turn become annotated instances of sense i for word w. Additionally, the algorithm may search for a monosemous (containing only one sense) synonym for sense i in order to get training examples. An example of this later idea is given in Figure 2.4. The idea is referred to more generally as using *monosemous relatives*, following the work of Agirre & Martínez (2004), which extended the idea to other monosemous beyond synonyms such as hypernyms and hyponyms.

Another approach to using the Web to help alleviate the *data acquisition bottleneck* is through *bootstrapping* a small set of annotated examples into a larger set. This idea was pioneered as part of the work by Yarowsky (1995), in which one begins with a *seed set* of annotated data, which is

```
Given: examples ($X$), initial seed set of labels for examples ($L^0$) (note $L^0_x$ denotes the label for example $x$)
    for $t = 0$ to $\infty$
        train a classifier ($C^t$) on the labeled examples ($\Lambda^t$), where $\Lambda^t = \{x \in X | L^t_x \neq \perp\}$
        let $p^t_x(j)$ be the probability which $C^t$ predicts label $j$ for example $x$
        foreach example $x \in X$:
            set $top\_label = arg\_max_j[p^t_x(j)]$
            set $L^{t+1}_x = \left\{ \begin{array}{cl} L^0_x & \text{if } x \in \Lambda^0 \\ top\_label & \text{if } p^t_x(top\_label) > \zeta \\ \perp & \text{otherwise } (\perp \text{ is an undefined value }) \end{array} \right\}$
        if $L^{t+1} = L^t$, stop
```

Figure 2.5: The original Yarowsky bootstrapping algorithm (Abney, 2004, “Y-0”).

grown by an iterative process given in Figure 2.5. A classifier trained on the *seed set* tries to label unannotated examples when there is a high probability that the label is correct ($p^t_x(top\_label) > \zeta$). Then, the process repeats with the newly labeled examples in addition to the seed set. As mentioned in the previous section, the Yarowsky method was only evaluated on words with two senses.

Mihalcea (2002) created a *bootstrapping* method that included the earlier idea of searching the web for lexical patterns based on WordNet information. In this case, the *seed set* was used for acquiring examples in addition to being used as the initial labels for the iterative bootstrapping process. Her automatically generated corpora was able to improve *WSD* results on the Senseval-2 all-words task (Edmonds & Cotton, 2001) over that of using a hand-annotated training set by itself. A similar approach has been used to identify extrapositional cases of ‘it’ within sentences Li et al. (2009).

Within *SRL*, *bootstrapping* has been applied successfully as well. Swier & Stevenson (2004) took the *seed set* of semantic roles to be those which have no other options according to a verb lexicon. For example, if a verb only has one possible role for the object, then an example with that verb and role is included in the initial labeled data. With a test corpus of British National Corpus sentences and semantic roles (Clear, 1993), Swier & Stevenson achieved an 87% accuracy, well above a 64% baseline of labeling with the most frequent role for a given slot.

Nakov & Hearst (2005) view the web itself as a training set in an algorithm performing *PP attachment*. They count frequencies of co-occurrences between prepositions and various parts of the phrase: the object of the preposition, a possible noun attachment, or a possible verb attachment. Additionally, if a phrase appears in an alternative form where the attachment is not ambiguous, it is taken into account. Their approach did not use a knowledge base other than the Web, and it was able to achieve results in line with that of other unsupervised approaches to *PP attachment*.

A final use of the Web in creating semantically annotated data is the use of *Wikipedia* as a sense inventory. Mihalcea (2007) did this by considering article topics in Wikipedia as a concepts. When the text of one article contains a hyperlink to another, that link serves a sense annotation for the word(s) which the hyperlink belongs. By manually producing a mapping of Wikipedia articles to WordNet senses, they were able to evaluate the use of Wikipedia as an annotated corpus on the Senseval-2 (Edmonds & Cotton, 2001) and Senseval-3 (Snyder & Palmer, 2004) lexical sample tasks. Their system, a Naive Bayes classifier with features like that of most supervised *WSD* systems, performed at 85% accuracy compared to a 73% MFS baseline. As discussed previously, many systems do not overcome the MFS baseline, so this validated Wikipedia as a sense annotated corpus.

2.2.3 Direct Use of the Web

Rather than acquiring and storing explicit knowledge, some approaches to *semantic interpretation* use the Web in a more direct fashion. These systems may still be considered *knowledge-based*, but the information acquired is usually specific for an instance or a corpus being annotated. Essentially, the Web itself becomes the knowledge source, as data is dynamically acquired during runtime of a semantic interpretation algorithm.

A common approach to using the Web directly is through the use of *monosemous relatives*. Monosemous relatives are words which are similar to a sense of the target word, but which only have one sense. This idea was proposed by Leacock et al. (1998). As mentioned in the previous section, relatives have been used to build sense tagged corpora (Mihalcea & Moldovan, 1999; Mihalcea, 2002; Agirre & Martínez, 2004). These methods queried large corpora with relatives rather than with the context to create annotated training data. In this sense, the data is acquired for word senses, rather than directly for a test instance. In order to acquire knowledge directly for an instance, context must be taken into account.

Martínez et al. (2006) present the *relatives in context* method. A key aspect of this method is the use of context in the Web queries. They produce queries with relatives in place of the target word in a context with a window size of up to 6. This greatly reduces the amount of results returned over that of just searching for instances of a word . Similarly, Yuret (2007) first chooses substitutes and determines a sense by looking at the probability of a substitute taking the place of the target word within the Web1T corpus. The number of hits each query has on the web is then used to pick the correct sense. Both Martínez et al. (2006) and Yuret (2007) incorporate a knowledge-base to

```
Given: sentence ($s$) and target word ($w$)
For each sense, $c$ of $w$:
    Find a set of relatives ($R_c$) of $c$ from a knowledge-base (WordNet).
    Set $hits_c = 0$
    For each $r \in R_c$
        Create $WebQueries$ with $r$ in place of $w$ in $s$ (using a various window sizes).
        Search the Web with $WebQueries$ and add the number of hits to $hits_c$
The chosen sense is the sense of $w$ with the most hits ($arg\_max_c[hits_c]$).
```

Figure 2.6: A generalization of the Martínez et al. (2006) and Yuret (2007) algorithms of using *relatives in context*.

construct queries with pre-chosen relatives. Issues of bootstrapping corpora, such as metaphorical usage, proper nouns in place of regular nouns, and noisy examples (badly formed), are avoided by using relatives in this direct fashion (Martínez et al., 2006). A generalization of the algorithm used by both is presented in Figure 2.6.

A drawback of these approaches is the limitation of selecting relatives before search. It is possible that many instances of the sentence with another word in place of the relative could have been found. These unrestricted words could still give a good indication of the correct sense. The work of this dissertation presented in section 4.3 searches the Web with context, but does not restrict results to pre-chosen relatives. The idea is similar to that of Lin (1997), in which context was searched through a database of dependency relationships. A dependency database was created ahead of time, and this *knowledge-based* method did not actually do any dynamic knowledge acquisition. However, because it is highly related to the dynamic approach, it is presented here.

When one searches with context, but without any pre-chosen relatives, they acquire *selectors*. A *selector* is a word which can take the place of another given word within the same local context (Lin, 1997). When being applied to *WSD*, one compares the selectors with the senses of the original word. Essentially, the target word is disambiguated by usages of other words, rather than usages

Given: sentence (s), target word (t), and a dependency parse database ($dpDB$):

$s =$ *He addressed the strikers at the rally.*, $t =$ ‘striker’

$$senses(t) = \left\{ \begin{array}{ll} 1 & \text{(gloss: “a forward on a soccer team”)} \\ 2 & \text{(gloss: “someone receiving intensive training for a naval technical rating”)} \\ 3 & \text{(gloss: “an employee on strike against an employer”)} \\ 4 & \text{(gloss: “someone who hits”)} \\ 5 & \text{(gloss: “the part of a mechanical device that strikes something”)} \end{array} \right\}$$

Find local context triple (lct) as a triple, from the dependency parse (dp) of s:

$dp(s) = \{address : \{subj \Leftarrow `he', comp1 \Leftarrow `strikers', prep \Leftarrow `rally'\}\}$

$lct(t, dp(s)) = (comp1, address, head)$

Search $dpDB$ for other words with the same lct:

$sels(t, s) = search(dpDB, lct(t, dp(s)))$

$= \{`audience',' letter',' students',' crowd', `question' `Palestine'\}$

(note that we omit mentioning of local context likelihood values in choosing selectors)

Predict the correct sense of t by maximizing similarity (sim) between each selector and the senses of of the target word ($senses(t)$):

$$predictsense(t, s) = \arg \max_{t_i \in senses(t)} \sum_{sel \in sels(t,s)} sim(sel, t_i) = \textit{striker-3}$$

Figure 2.7: An example of the Lin (1997) selector algorithm to disambiguate the word ‘strikers’.

of itself. It does not matter how often a word itself appears in a corpus. Figure 2.7 walks through an example of disambiguating ‘strikers’ in the sentence (18) based on Lin’s algorithm. The local context was defined specifically as a triple containing the type of dependency of the target word, the word in which it is dependent upon, and the position (*head* or *mod*). In the end, the similarity measure (discussed in Section 2.3) finds that many of the selectors have senses similar to *striker-3*.

(18) He addressed the strikers at the rally.

Like selectors, Dligach & Palmer (2008) used *dynamic dependency neighbors*(*DDN*), which are verbs occurring in the same dependency relationship (object) with nouns as a given target verb. The DDNs are actually considered neighbors of a noun, since they are essentially other verbs that take the noun as an object. Still, one may view a *DDN* as a specific type of selector, where the local context is defined as an object with the same noun. In (Dligach & Palmer, 2008), DDNs were successfully used as a feature in a supervised verb sense disambiguation system. They found the DDN feature to outperform other semantic features on select verbs from the OntoNotes project (Hovy et al., 2006) by a small, but significant, margin.

Although Lin (1997) and Dligach & Palmer (2008) were both essentially capturing selectors, their approaches require the parsing of text which is not yet feasible on the Web. Additionally, in both cases, the local context does not take into consideration other words beyond the one in which the target is related. Consider the selectors ‘letter’ and ‘question’ returned for the example of Lin’s method (Figure 2.7). These nouns may also be the head of the subject of a difference sense of ‘address’. This can lead to wrong predictions. If more local context was taken into consideration, such as the prepositional phrase “at the rally”, the selectors may have been less ambiguous. A similar problem could occur with DDNs, if they were based on a highly ambiguous object, which may have a wide variety of DDNs. Within this dissertation, a method is presented of applying selectors to the Web (section 4.3), and a goal of the method is to try to capture as much local context as possible, not just that of one dependency relationship.

2.3 Semantic Similarity and Relatedness Measures

Similarity and relatedness measures have seen wide use for problems of semantic interpretation (Lin, 1997; Leacock & Chodorow, 1998; Stetina et al., 1998; Resnik, 1999; Banerjee & Pedersen, 2002; Patwardhan et al., 2003; Budanitsky & Hirst, 2006; Sinha & Mihalcea, 2007; Schwartz & Gomez, 2008, 2009b; Agirre & Soroa, 2009). These algorithms are used to rate the strength of semantic similarity or relatedness between two concepts. In turn, they can compare concepts of a word in a sentence in order to resolve ambiguities and perform semantic annotations. While other works have applied *similarity* over different terms, such as comparing two *pairs* of words when measuring analogy (Turney, 2008), this section is focused on metric comparing two *single* concepts.

Measures of similarity and relatedness are often broken down in to three categories: *path based*, *information content*, and *gloss based* (Pedersen et al., 2004). *Path based* approaches rely entirely on graphs of relationships, using the idea that concepts closer to each other, according to the length of path between the two, are more similar. The other types of measures may take advantage of graphs, but they are distinguished in that they also take into consideration a concept's *information content* or *gloss*. *Information content* places a value on the amount of information a concept subsumes, while a *gloss* (discussed in section 2.1.1) is a description of a concept that can give clues to other related concepts.

A key notion is that similarity is a specific type of relatedness (Rada et al., 1989; Resnik, 1999; Patwardhan et al., 2003; Agirre & Soroa, 2009). In particular, *similarity* is characterized by the relationships: synonymy, antonymy, and hyponymy, while *relatedness* indicates a general non-explicit relationship. Consider the concepts *ballplayer* (gloss: "an athlete who plays baseball") and *bat* (gloss: "a club used for hitting a ball in various games"). One would easily concede that the concepts are *related*, but a *ballplayer* and *bat* are not similar. On the other hand, *bat* and *stick* (gloss: "an implement consisting of a length of wood") would be considered *related* and *similar*.

Note that concepts are not the same as words (as discussed in section 1.1), and other senses of 'bat' may not be similar to *stick*.

While all methods presented in this section measure *relatedness*, most *path based* and *information content* measures are distinguished as more specifically measuring *similarity*. These similarity measures take advantage of the *is-a* (*hypernym* see section 2.1.1) relationship within the WordNet ontology. In fact, the 'path' in *path based* refers to the path through the WordNet ontology, which in the case of similarity, includes only *is-a* links between concepts. Furthermore, relatedness measures should be able to compare concepts of words from different parts of speech.

All types of similarity and relatedness measures return a value, $strength$, representing the strength of the relation between the two concepts. The strength usually can be normalized to range between 0 and 1 (0 indicating no relatedness and 1 indicating synonymy). For two concepts, c_1 and c_2 (nodes in WordNet), a similarity(S_{meas}) or relatedness measure (R_{meas}) will follow this form:

$$S_{meas}(c_1, c_2) = strength \text{ OR } R_{meas}(c_1, c_2) = strength$$

The *lowest common subsumer* (*lcs*) is central to many similarity measures (Mihalcea, 2006). Given a pair of a concepts, c_1 and c_2, the $lcs(c_1, c_2)$ is the deepest (or lowest) concept which is a hypernym (directly or by transitive closure) of both concepts. In other words, it is a concept in which both c_1 and c_2 are said to be a type of. Since there are often multiple of such subsumers, it is the one closest to c_1 and c_2. Figure 2.8 gives a graphical depiction of the lcs between various concepts. Note that the lcs may be c_1 or c_2, if one subsumes the other.

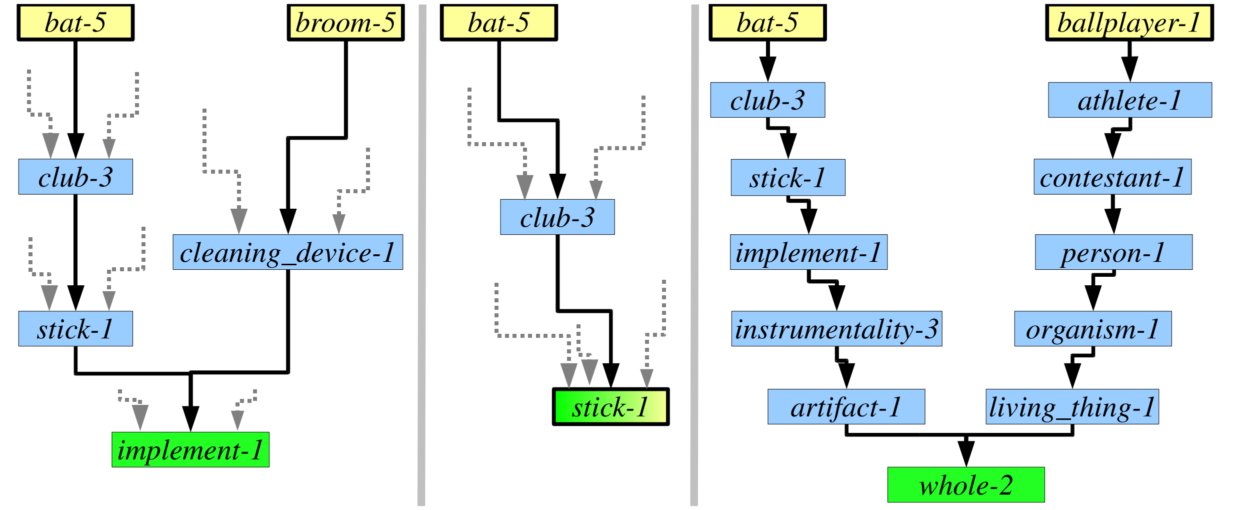


Figure 2.8: Paths to the lowest common subsumer among pairs of concepts: lcs(*bat-5, broom-5*) is *implement-1*, lcs(*bat-5, stick-1*) is *stick-1*, lcs(*bat-5, ballplayer-1*) is *whole-2*.

2.3.1 Path based similarity

Many similarity measures have been created which rely entirely on paths (or edges) in the WordNet ontology. In the simplest form, these measures compute the length of the shortest path between two concepts over the hypernym/hyponym relationship (Rada et al., 1989). A short path represents a strong similarity while a longer path indicates weak similarity. Since these methods rely on counting edges within a graph or ontology, they are also known as *edge based* (Jiang & Conrath, 1997).

A well known problem with early path-based measures is the assumption that the edges between concepts are all uniform (Resnik, 1999). This, *uniformity problem*, presents itself clearly in WordNet as concepts under *organism* (gloss: "a living thing that has (or can develop) the ability to act or function independently") appear with great depth, while concepts in other areas, such as *psychological_feature* (gloss: "a feature of the mental life of a living organism") are much more shallow. For example, *hydrangea* (gloss: “any of various deciduous or evergreen shrubs of the genus Hydrangea”) has a depth of ten, while *fractal* (gloss: “(mathematics) a geometric pattern that is repeated at every scale and so cannot be represented by classical geometry”) has a depth of six from the root of the ontology, *entity*. In this case, it could be argued that the links leading to *hydrangea* represent more subtle differences than the links to *fractal*. Essentially, if one assumes the edges are uniform, then one would be saying a *fractal* is 6 units different than the root (*entity*), while a *hydrangea* is 10 units different. These differences in density also occur within subgraphs of the ontology (i.e. the subconcepts of *whole-2* extend much deeper than those of *location-1*, but both concepts are hyponyms of *object-1*).

Path based approaches to handling the *uniformity problem* rely on a scaling of some sort. Wu & Palmer (1994) did this by considering the depth of the *LCS* between two concepts:

$$S_{WuPalmer}(c_1, c_2) = \frac{2 * depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{2.1}$$

where $depth$ is the length of a path from *entity* to a concept. Essentially, the path length between concepts is scaled by the lcs. This is more clear if the function is rewritten equivalently as:

$$S_{WuPalmer}(c_1, c_2) = \frac{2 * depth(lcs(c_1, c_2))}{2 * depth(lcs(c_1, c_2)) + dist(c_1, c_2)} \tag{2.2}$$

where $dist$ is the number of edges on a path between two concepts. Thus, $dist(c_1, c_2)$ is equivalent to $dist(c_1, lcs(c_1, c_2)) + dist(c_2, lcs(c_1, c_2))$ since $lcs(c_1, c_2)$ is a point in the shortest path between c_1 and c_2.

One aspect that Wu & Palmer (1994) did not consider was the depth of the entire ontology. This can be important as nodes which are closer to the root, relative to the rest of taxonomy, are more general than those near leaf nodes. Leacock et al. (1998) scale the distance between concepts by a term, $maxD = \max_{c \in WN}[depth(c)]$, representing the maximum depth of all concepts in WordNet:

$$S_{LeacockChodorow}(c_1, c_2) = -log(\frac{dist(c_1, c_2)}{2 * maxD}) \tag{2.3}$$

Note that $maxD$ depends only on whether the concepts are nouns or verbs (nouns and verbs have separate ontologies), and $dist(c_1, c_2)$ is at most $2 * maxD$.

As a final observation on *path-based* similarity, some methods of *WSD* presented in section 2.2.1 inherently contain similarity or relatedness measures over a graph (Navigli & Velardi, 2005; Sinha & Mihalcea, 2007; Agirre & Soroa, 2009). Although a similarity between two concepts is not explicitly given by these approaches, they rely on an ontology like WordNet in order to compute distances between many (usually more than two) concepts. The idea is to choose the set of word senses with maximum similarity to each other as a whole, rather simply between any two words.

2.3.2 Path based relatedness

Path-based approaches have also been implemented which are more appropriately categorized as measuring *relatedness*. One such measure was adopted from computing semantic distance over a thesaurus (Hirst & St Onge, 1998). The algorithm relies on *antonymy* in addition to *hypernymy/hyponymy* (*is-a*) relationships, and they assume that more *turns* in the path between two concepts indicates less relatedness. A $turn$ occurs when one relationship is followed by a different type of relationship, such as following a *hypernym* path to a concept and then changing to a *hyponym*. Although it was originally constructed in more abstract terms, for two concepts Mihalcea (2006) describes it as follows:

$$R_{HirstOnge}(c_1, c_2) = C - dist(c_1, c_2) - k * turns(c_1, c_2) \tag{2.4}$$

where C is constant upper bound for relatedness and k is a constant used to weight the penalty of turns. These constants are often set as $C = 8$ and $k = 1$ (Budanitsky & Hirst, 2006). Due to the inclusion of other relationships in addition to *is-a*, this measure is distinguished as a more general *relatedness* measure.

More recently, Yang & Powers (2005) introduced a measure taking *meronymy/holonymy* into account in addition to *synonymy/antonymy* and *hypernymy/hyponymy*. All of these relationships are recorded in WordNet. *holonymy* is a *part-of* or *member-of*relationship (where *meronymy* expresses the opposite relationship: *has-part* or *has-member*). For example *car-1*(gloss: "a motor vehicle with four wheels; usually propelled by an internal combustion engine") has a meronym *bumper-2* (gloss: "a mechanical device consisting of bars at either end of a vehicle to absorb shock and prevent serious damage") among others. They also consider the a relationship through the gloss, where all senses of words contained in the gloss are considered linked through the $gloss$ relationship. Like Hirst & St Onge (1998), Yang & Powers also weight the edge by the type of relationship it represents, and include a threshold over the path distance, γ. First only devising their method for nouns, they later extend it to handle verbs, which meant including relationships that applied to verbs as well (Yang & Powers, 2006). Essentially, this new version uses any type of relationship it can in the WordNet ontology, and is described as follows:

$$R_{YangPowers}(c_1, c_2) = \begin{cases} \alpha_{stm}\alpha_t \prod_{i=0}^{dist(c1,c2)} \beta_{t_i} & , \text{ if } dist(c1, c2) < \gamma \\ 0 & , \text{ if } dist(c1, c2) \geq \gamma \end{cases} \tag{2.5}$$

where α_{stm} is a factor based on stemming, $alpha_t$ is a factor based on the relationship types (t) used in the path, and β_{t_i} a depth factor based on the relationship type. To clarify further, α_{stm} is 1 when there is not stemming needed, and a value between 0 and 1 when stemming is used (they find a value of 0.4 after some initial testing). Yang & Powers (2006) also tune the values for $alpha_t$, $beta_t$, and $gamma$ based on a sample set. The values they discovered for $alpha$ factors were $\{\alpha_{same} = 1, \alpha_{synonym} = 0.9, \alpha_{antonym} = 0.9, \alpha_{holonym} =, 0.85\alpha_{entails} = 0.85, \alpha_{cause} = 0.85, \alpha_{also} = 0.85, \alpha_{similarity} = 0.85, \alpha_{pertanym} = 0.85, \alpha_{derived} = 0.8\alpha_{identity} = 0.7, \alpha_{gloss} = 0.5\}$. For the β factors they found $\beta_{noun} = 0.7$ and $\beta_{verb} = 0.2$. Finally, they also found a threshold of $\gamma = 2$ to be best, indicating that the shortest path between two concepts must be 1. This essentially reduced the approach to simply being based on single-links rather than a path. Another drawback of this approach is that all of these parameters may be tuned to overfit, and will not generalize to other domains. We will apply the $R_{YangPowers}$ measure to other domains in out approach.

2.3.3 Information content based similarity

As an alternative response to the previously mentioned *uniformity problem*, the approaches presented below reduce the reliance on paths in WordNet by calculating an *information content* (*ic*) value for a node (or concept). These measures still use the *is-a* relationship in WordNet, but they do not rely directly on edges to determine the strength of a relationship between concepts. Resnik (1995) introduced *ic* as:

$$ic(c) = -log(p_{cncpt}(c)) \tag{2.6}$$

where $p_{cncpt}(c)$ is the probability that a concept or one of its descendants appears in a corpus. It is computed over instances of all words in a corpus. Assuming one has the probability of a word occurring in a corpus, p_w, to arrive at a probability for concept occurrence, first p_w is spread among its senses (Richardson & Smeaton, 1995):

$$p_{ws}(ws) = \frac{p_w(lemma(ws)}{senses(lemma(ws))} \tag{2.7}$$

where $senses$ returns the number of senses of the word ($lemma$) within the word-sense ws. One can then extend the probability to apply to a synset by summing p_{ws} over all word senses which belong to the synset, $syns$:

$$p_{syn}(syns) = \sum_{ns \in syns} p_{ws}(ns) \tag{2.8}$$

Although a synset is essentially a concept, these functions have yet to consider the descendents of a synset. Below is a recursive function to arrive at p_{cncpt} based on the idea that a concept subsumes all concepts below it (hyponyms, $hypos$) in the WordNet ontology:

$$p_{cncpt}(c) = p_{syn}(syns(c)) + \sum_{h \in hypos(c)} p_c(h) \tag{2.9}$$

To step from ic to similarity measure, one simply uses the *information content* of the *lowest common subsumer*. By using ic rather than distances in the ontology, edge weights are not directly considered and the *uniformity problem* is avoided. Thus, the similarity of two concepts is defined as the amount of information they have in common, given by their lcs (Resnik, 1999):

$$S_{Resnik}(c_1, c_2) = ic(lcs(c_1, c_2)) \tag{2.10}$$

However, simply observing the ic of two concepts' common ancestor (lcs) does not consider the difference in information between the lcs and each concept. With this in mind, Jiang & Conrath (1997) augmented Resnik's function such that the combined ic of the concepts is subtracted from the lcs. Because $(2 * ic(lcs(c_1, c_2))) \leq (ic(c_1) + ic(c_2)) \leq 1$ in the definition of ic, this function ranges from -1 to 0 (0 being maximum similarity):

$$S_{JiangConrath}(c_1, c_2) = 2 * ic(lcs(c_1, c_2)) - (ic(c_1) + ic(c_2)) \tag{2.11}$$

Lin (1998b) defined similarity according to information theory. Although his work focused on more than just concept similarity in a taxonomy, it also resulted in a measure of concept similarity. It is based on scaling the information of both concepts by the information of the lcs. The function follows like that of $S_{WuPalmer}$ or $S_{SchwartzGomez}$, where ic is used in place of $depth$ or nd. A variant of it was used in (Lin, 1997) for noun sense disambiguation:

$$S_{Lin}(c_1, c_2) = \frac{2 * ic(lcs(c_1, c_2))}{ic(c_1) + ic(c_2)} \tag{2.12}$$

An alternative definition of *information content* may be applied to any of these measures. Ambiguity of words leads to error in computation of p_{cncpt} when the assumption is made that all senses of a word should equally receive p_w. To compensate for this a corpus annotated with word senses, such as SemCor, could be used instead (Pedersen et al., 2004). In this case, rather than compute p_{ws} from p_w, p_{ws} is a given, based on frequency information. Although it is important to understand the more complex computation from p_w, this modified realization of *information content* from p_{ws} is used more often in practice.

2.3.4 Gloss based relatedness

Inspiration for *gloss based relatedness* can be traced to a *WSD* algorithm by Lesk (1986). The algorithm finds the number of words in common between dictionary definitions (or glosses) of multiple word senses (or concepts). These words in common are termed *overlaps*. See section 2.2.1 for a description of how the idea is used for disambiguation. As a standalone relatedness measure of two concepts over WordNet, the Lesk algorithm simply returns the overlap from glosses of concepts:

$$R_{Lesk}(c_1, c_2) = |words(gloss(c_1)) \cap words(gloss(c_2))| \tag{2.13}$$

where $words(gloss(c))$ returns the set of words in the gloss for concept c.

Banerjee & Pedersen (2002) extends the standard Lesk measure of relatedness to utilize relationships in WordNet. In addition to a concept's gloss, the glosses of related concepts are included as well. These related concepts are obtained through these relationships: *hypernym*, *hyponym*, *meronym*, *holonym*, *troponym*, and *attribute*. Another extension Banerjee & Pedersen present is consideration for the length of overlap between concepts in addition to the number of overlaps. In turn, matching sequences of words is considered a single *overlap*, and its weight is equal to the number of words in the sequence ($|words(overlap)|$) squared:

$$R_{BanerjeePederson}(c_1, c_2) = \sum_{rg_1 \in rgls(c_1)} \sum_{rg_2 \in rgls(c_2)} \sum_{o \in seq(rg_1, rg_2)} (|words(o)|)^2 \qquad (2.14)$$

where $rgls$ returns the set of related glosses for a concept (including its own gloss) and seq returns all of the sequences two phrases have in common (overlaps). *Overlaps* that do not contain any nouns, verbs, adjectives, or adverbs are thrown out. This measure is also referred to as *adapted Lesk*.

Other measures of relatedness use co-occurrence information in combination with words of glosses. Patwardhan & Pedersen (2006) create *gloss vectors* from co-occurrences and WordNet glosses, and based relatedness on the cosine between any two concepts vectors:

$$R_{PatwardhanPederson}(c_1, c_2) = cos(angle(\vec{v_1}, \vec{v_2})) \qquad (2.15)$$

The gloss vectors, $\vec{v_1}$ and $\vec{v_2}$, contain frequencies of occurrences of all words which appear in a gloss for c_1 and c_2 respectively. The idea extends from a word sense discrimination approach, much like the Lesk algorithm, where *context vectors* were defined for all words within the context of a target word (Schütze, 1998).

2.3.5 Evaluating Measures through Application

Several works have formulated experiments to determine how similarity and relatedness measures stack up to each other in a variety of situations. Some evaluations (Resnik, 1999; Yang & Powers, 2005; Agirre & Soroa, 2009) were based on manually crafted similarity data for words rather than concepts (Miller & Charles, 1991; Rubenstein & Goodenough, 1965). Although the studies based on hand crafted data often found *information-content* measures outperform *path-based* measures (Resnik, 1999; Agirre & Soroa, 2009), this section focuses on studies applying concept *similarity* and *relatedness* in NLP algorithms. Table 2.2 lists the measures categorized as measuring *similarity* or *relatedness* and the type of approach as described previously. The $S_{WuPalmer}$ and $R_{YangPowers}$ are not shown in this review of previous evaluation because they were only evaluated through manually-crafted data. Chapter 5 will present an evaluation including all the measures in Table 2.2 plus another that is a contribution of this work.

One of the first comprehensive evaluations of WordNet semantic relatedness measures involved an application to a spell correction algorithm (Budanitsky & Hirst, 2001, 2006). For a potential misspelling or malapropism (an incorrect spelling of a word that results in the correct spelling of

Table 2.2: Categorization of similarity and relatedness measures.

Similarity - Path Based	
$S_{WuPalmer}$	Wu & Palmer (1994)
$S_{LeacockChodorow}$	Leacock et al. (1998)
Similarity - Information Content	
S_{Resnik}	Resnik (1999)
$S_{JiangConrath}$	Jiang & Conrath (1997)
S_{Lin}	Lin (1998b)
Relatedness - Path Based	
$R_{HirstStOnge}$	Hirst & St Onge (1998)
$R_{YangPowers}$	Yang & Powers (2006)
Relatedness - Gloss Based	
$R_{BanerjeePedersen}$	Banerjee & Pedersen (2002)
$R_{PartwardhanPedersen}$	Patwardhan & Pedersen (2006)

another word), the algorithm determined if any of the senses are related to other words in context (this step is referred to as *suspicion*). When a word does not have any senses related to nearby words, the system determines if any senses of similarly spelled words are related to the other words in context (referred to as *detection*). Budanitsky & Hirst (2006) write, "For example, if no nearby word in a text is related to diary but one or more are related to dairy, we suggest to the user that it is the latter that was intended." Their evaluation was run on 107,233 candidate misspellings (1,408 being actual misspellings), for which two evaluations were run with these measures $S_{LeacockChodorow}, R_{HirstOnge}, S_{Resnik}, S_{Lin}$, and $S_{JiangConrath}$. For the first step, *suspicion*, $S_{JiangConrath}$ performed best but not significantly better than $S_{LeacockChodorow}$ or S_{Lin}. However, after the second step, *detection*, $S_{JiangConrath}$ did show significant improvement over the other measures (Budanitsky & Hirst, 2006).

Patwardhan et al. (2003) developed a Lesk style *WSD* algorithm for nouns in which senses of the target word are compared to senses of the first three nouns on the left and right of the target word. It was previously shown that $R_{BanerjeePederson}$ performed twice as good as R_{Lesk} on Senseval-2 noun data (Banerjee & Pedersen, 2002). In (Patwardhan et al., 2003), the authors focused on the following measures: $S_{LeacockChodorow}$, $R_{HirstOnge}$, S_{Resnik}, S_{Lin}, $S_{JiangConrath}$, and $R_{BanerjeePederson}$. The $R_{BanerjeePederson}$ performed best on the Senseval–2 set of 29 nouns (totaling 1723 instances), followed closely by $S_{JiangConrath}$. Additionally, with the exception of S_{Resnik}, *information content* measures outperformed the two *path-based* measures. In a smaller subsequent experiment, Patwardhan et al. (2003) also found that alternative computations of *information content* did not lead to significant changes in performance.

As part of the introduction to *gloss vectors*, Patwardhan & Pedersen (2006) presented an evaluation in conjunction with five other relatedness measures used in (Patwardhan et al., 2003) (omitting $R_{HirstOnge}$). The rest of the experiment followed like that of (Patwardhan et al., 2003), and found that the $R_{PatwardhanPederson}$ performed just below that of $R_{BanerjeePederson}$, both outscored by $S_{JiangConrath}$. The results are presented in Table 2.3, along with other experiments. Note that the table is meant to show the differences in performance between measures on a single evaluation. The authors do not clearly explain why the accuracies were slightly different between the two relatedness sense disambiguation experiments, but parameters can vary widely even when the task between evaluations are the same.

The evaluations mentioned thus far used metrics for comparing a target word (or senses of a target word) to other words in context. The assumption is that concepts in context are *related*, but as previously mentioned, *relatedness* does not imply *similarity*. Thus, the measures which are more appropriately categorized as measuring *similarity* (those which do not consider relationships

Table 2.3: Results of various application-oriented similarity and relatedness evaluations.

	SC$_{sus}$	**SC**$_{det}$	**NSD**$_{rel1}$	**NSD**$_{rel2}$
$S_{LeacockChodorow}$	0.115	0.184	0.31	0.30
$R_{HirstOnge}$	0.091	0.145	0.32	-
S_{Resnik}	0.075	0.150	0.30	0.30
S_{Lin}	0.110	0.201	0.33	0.36
$S_{JiangConrath}$	0.141	0.254	0.38	0.45
$R_{BanerjeePederson}$	-	-	0.39	0.44
$R_{PatwardhanPederson}$	-	-	-	0.41
units	F1	F1	accuracy	accuracy

SC$_{sus}$: (Budanitsky & Hirst, 2006, *suspicion*), **SC**$_{det}$: (Budanitsky & Hirst, 2006, *detection*), **NSD**$_{rel1}$: (Patwardhan et al., 2003), **NSD**$_{rel2}$: (Patwardhan & Pedersen, 2006),

beyond hyponymy, antonymy, and synonymy) may be at a disadvantage. The evaluation we present in Chapter 5 uses a *WSD* algorithm, where noun senses were compared with senses of words that are found to replace that noun in its context (a task calling for *similarity* comparisons). Past studies have either focused entirely on relatedness or only evaluated judgments over words rather than concepts.

3 A DATABASE OF APPLICABLE COMMON SENSE KNOWLEDGE

This chapter begins discussion of the research contributed to the field as part of this dissertation. In particular an approach for acquiring *common sense knowledge* (*CSK*) from the Web is presented. Common sense knowledge refers to knowledge about the world which people use for understanding and perception in their everyday lives. All of the acquisition methods in this dissertation rely on the idea of searching the Web with context.

The key contributions within this Chapter include novel approaches for both the acquisition and the applicable analysis of knowledge. During acquisition, generic search phrases are used with automatically filled constituents to query the Web, and a statistical parser is incorporated to verify that the syntactic structure of results from the Web match an intended structure. In order to create more applicable knowledge about concepts rather than ambiguous words, we automatically analyze word relationships acquired from the Web over WordNet in order to generalize information about concepts or areas of the ontology. This novel analysis could be applied to data containing word relationships of any kind. Finally, as a secondary contribution, the common sense knowledge base (*CSKB*) created by this process will be made available for others to use as a resource (Schwartz & Gomez, 2009a).

The work presented in this chapter can be broken into two major steps. To summarize, the first step searches of the Web to acquire relationships between ambiguous nouns, while the second major step analyzes word senses or concepts over WordNet to induce knowledge about concepts from ambiguous nouns. Section 3.1 describes a method for acquiring knowledge for the *common sense knowledge* database. The method begins with the automatic construction of Web queries, which retrieve samples of sentences and phrases from the Web. A statistical parser is incorporated to verify that the samples obtained from the Web match the syntactic structure of the query. Through a novel concept analysis over WordNet (Section 3.2), relationship information is induced between a concept and a word rather than between two, often ambiguous, words. Section 3.3 evaluates the usefulness of the acquired knowledge by applying it to the task of word sense disambiguation. Results show that the knowledge can be used to improve the accuracy of a state of the art minimally-supervised disambiguation system.

3.1 Method for Acquiring Common Sense Knowledge

The term *common sense knowledge* is used to refer to the type of knowledge which is used in every day life without necessarily being aware of it, such as that which tells us ‘keys’ are often found in one’s pocket. Although *common sense* may refer to a process, such as reasoning, it is important to note that we are referring only to a type of knowledge. Panton et al. (2006) of the Cyc project define common sense as “the knowledge that every person assumes his neighbors also possess”. The benefits of *CSK* cross into many fields. For example, in computer vision researchers have found that, for object recognition, it is useful to have knowledge describing the context in which an ordinary object may appear (Strat & Fischler, 1991; Rabinovich et al., 2007; Torralba et al., 2010); Hu et al. (2009) found world knowledge to be helpful in clustering text for aggregated search. In this work, *CSK* is used with respect to understanding meaning in natural language.

3.1.1 Common Sense Knowledge Based on Prepositions

The implementation of the system is focused on a type of *CSK* describing what is often found in or on something. For example, one would expect to find coins, a cell phone, or keys in a pocket; food, waiters or a jukebox in a restaurant; thoughts, fear, or pictures in a mind; and books, food, or elbows on a table. This knowledge, expressed as a relationship between entities, could be described as follows.

A relationship *e1***R***e2*, exists between entities *e1* and *e2*
if one finds "*e1* is **R** *e2*," where **R** $\in$ {'in', 'on'}.

To clarify the relationship, a brief linguistic background of prepositions and relationships should be considered. Quirk tells us that prepositions state a relationship between two entities, where one of the entities is typically a constituent of the sentence and the other is the complement to the preposition (Quirk et al., 1985). For example, consider the relationship between 'key' and 'pocket' in the variations of the sentence below.

Table 3.1: Abstract dimensions (**dims**) and corresponding prepositions.

dims	**description**	**prepositions**
1 or 2	*on* surface or line	on, onto, atop, upon, on top of, down on
2 or 3	*in* area or volume	in, into, inside, within, inside of

The key is...

...at the pocket.

...on the pocket.

...in the pocket.

'The key' is the subject of the sentence, while 'the pocket' is a prepositional complement. The preposition 'in' indicates a relationship between 'key' and the prepositional complement 'pocket'. Notice that the meaning is different for each sentence depending on the actual preposition ('at', 'on', or 'in'), and thus *key* relates to *pocket* in three different ways. Although each relationship between *key* and *pocket* is possible, only one would likely be considered *CSK*: *key***in***pocket*.

We use prepositions which indicate a positive relationship given by Quirk et al. (Quirk et al., 1985). There are three types of such relationships: "at a point", "on a line or surface", and "in an area or volume". In particular, we concentrate on the 1 to 3 dimensional relationships given in Table 3.1, denoted *on* and *in* throughout the paper. *At*, the 0 dimensional relationship, occurred far less frequently. The sentences below exemplify the various dimensions.

(19) *on* surface or line *The keyboard is on the table.* *The motion is on the table.*
The beach is on US 1. *A thought is on his mind.*

(20) *in* area or volume *The bank is in New York.* *The request is in the queue.*
The vegetables are in the bowl. *New York is in the playoffs.*

While dimensions are used to clarify the types of prepositions used, the description of the *CSK* (given at the top of this section) is based on language rather than geometry. It would be inappropriate to categorize this knowledge as either spatial relationships or part-whole relationships. One would say it is common to find knowledge in one's head though the relationship *knowledge***in***head* is not physical. Additionally, one finds a waiter in a restaurant very often though the waiter is not attached as a part of the restaurant. One may still argue that these relationships, even when abstract, are 'spatial relationships', but this debate is outside the scope of this work. In the end, this work acquires the general *CSK* of (physical or abstract) entities that are in or on other (physical or abstract) entities, and we show this knowledge is useful for semantically processing natural language.

3.1.2 Acquisition Framework

This section describes the acquisition of nouns (as words) from the Web which are in a relationship with other nouns. A Web search is performed in order to retrieve samples of text matching a *web query* created from a *search phrase* for the relationship. Each sample is syntactically parsed to

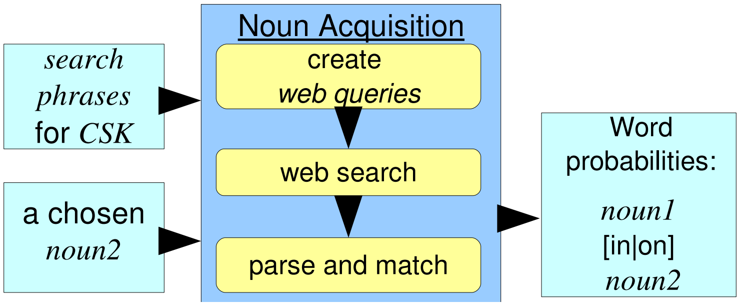


Figure 3.1: The *common sense knowledge* acquisition framework under the assumption one is looking for *noun1s* in a relationship with a given *noun2*.

verify a match with the corresponding *web query*, and the noun(s) filling a missing constituent of the parse are recorded.

The framework, given in Figure 3.1, is very flexible, and it can handle the acquisition of words from other parts of speech. However, to be clear, the explanation focuses on the use of the framework to acquire specific types of relationships between nouns. The process is broken up into three procedures: "*Web query* creation", "Web search", and "parse and match".

Web Query Creation

Web queries are created semi-automatically by defining these parameters of a *search phrase*:

noun1	the first noun phrase
noun2	the second noun phrase
prep	preposition, if any, used in the phrase
verb	verb, if any, used in the phrase.

The verb is statically defined as part of the *search phrase*. Refer to Table 3.2 for a list of all of the *search phrases* used, one of which appears as an example throughout this section:

Table 3.2: *Search phrases* and relationships used for acquisition of a *CSKB*.

relation	**search phrase**	**voice**
on, in	*noun1* is located *prep noun2* *noun1* is found *prep noun2* *noun1* is situated *prep noun2*	passive
on, in	*noun1* is *prep noun2* put *noun1 prep noun2* place *noun1 prep noun2* lay *noun1 prep noun2* set *noun1 prep noun2* locate *noun1 prep noun2* position *noun1 prep noun2*	active
on	hang *noun1 prep noun2* mount *noun1 prep noun2* attach *noun1 prep noun2*	active

place *noun1 prep noun2*

Prepositions are chosen to describe the type of relationship that is sought-after, as described in Section 3.1.1. Those for the **on** and **in** relationships are used to fill the *prep* parameter of a *search phrase*.

$prep$(**on**) = (*on, onto, atop, upon, on top of, down on*)

$prep$(**in**) = (*in, into, inside, within, inside of*)

Determiners and possessive pronouns, selected from the list below, are included by the system when noun parameters are provided. This allows greater accuracy in our search results. The system uses all of these determiners, and those which are not appropriate will return few, if any, results. For example, pocket typically was preceded by ‘my’ or ‘your’, and infrequently preceded by ‘that’.

det = (*the, a/an, this, that, my, your, his, her*)

```
for each $search\_phrase$
   for each $prep(\mathbf{R})$
      for each $det$
         $web\_query$ = create_query($search\_phrase, prep, det, noun2$);
         $samples$ = websearch($web\_query$);
```

Figure 3.2: Pseudocode for process of creating *web queries* from *search phrases* and retrieving samples from the web.

Finally, the undefined parameters are replaced with the wildcard indicator, '*'. These are the words being acquired from the Web. Below is a *web query* created from our example *search phrase* where *noun2* is 'refrigerator', *prep* is 'in', *det* is 'the', and *noun1* is undefined.

place * in the refrigerator

Contextual Web Search

The system uses all combinations of *search phrases*, *preps*, and *dets* to generate *web queries* which are used for retrieving phrases from the Web. The assumption is made that combinations which are not common will return few or no results. Given a *noun2* and a relation, $\mathbf{R} = \mathbf{on}|\mathbf{in}$, the search algorithm can be summarized through the pseudocode given in Figure 3.2.

The searches were carried out through the Google Search API[1], or the Yahoo! Search Web Services[2]. Each *web query* resulting from a *search phrase*, listed in Table 3.2, was run until a maximum of 1000 results were returned or no further results were found. The phrases returned from a search using a *web query* are referred to as $samples$. As part of this process, duplicate $samples$ were removed to reduce the effects of websites replicating the text of one another. Note

[1] no longer supported by Google

[2] http://developer.yahoo.com/search/

that samples usually contain much more text than just the portion that matches the *web query*. Therefore, the chance of an entire *sample* being a duplicate is small, unless one web page was a copy of another. The remaining *samples* are stored so that they can be parsed in order to determine the words taking the place of a missing constituent in the search phrase (the missing constituent is always *noun1* in this work). Below are some random samples returned with our example web query "place * in the refrigerator":

(21) ***Place the pan in the refrigerator***. *In a large mixing bowl, ...*

(22) ***Place the butter in the refrigerator*** *until it hardens. To use, peel the plastic ...*

(23) *Store in a cool, dark* **place; do not store in the refrigerator.** *Will keep up to ...*

(24) *... while fillets are frozen, then* ***place them in the refrigerator to thaw***. *...*

(25) *... completely cooled cake and* ***place cake in the refrigerator*** *until ready to serve. ...*

Within samples (21), (22), (24), and (25) it is being communicated to place 'pan', 'butter', 'them', and 'cake' in the refrigerator. In the case 'them', the system does not resolve pronouns because the it may introduce bias and errors, plus it is not necessary with the size of the Web. Furthermore, sample (23) matches the web query, but it is not communicating to place something in the refrigerator. The next step of the process addresses these issues and more while automatically matching constituents in the *search phrase* with the help of a syntactic parser and WordNet.

parsed *web query*:

```
    (VP (VB place)
        (NP (NN something))
        (PP (IN in) (NP (DT the) (NN refrigerator))))
```

parsed web *sample*:

```
(S1 (S (NP (PRP He))
    (VP (AUX was) (VP (VBN told) (S (VP (TO to)
        (VP (VB place)
            (NP (DT the) (JJ mixed) (NN batter))
            (PP (IN in) (NP (DT the) (NN refrigerator))))))))
```

Figure 3.3: An example of a parsed web query and the parse of the *sample* from the Web which matches the query.

Validation through Parsing and Matching

This step attempts to accurately match the constituents of a Web *sample* describing the relationship:

$$\textit{noun1} \text{ is [on} \mid \text{in] } \textit{noun2}$$

i.e. "the batter is in the refrigerator"

The system parses both the *web query* and the samples returned from the web with Charniak's parser Charniak (2000) in order to ensure accuracy. The word 'something' is inserted in place of the missing *search phrase* parameter '*'. One is then able to determine which part of the parse should match the word(s) retrieved from the Web. Consider Figure 3.3, parses for the *web query*: "place * in the refrigerator", and a *sample* returned from the Web: "he was told to place the mixed batter in the refrigerator".

Notice 'something' appears in place of the '*' in the parsed *web query*. In the parsed *sample*, the head noun(s) which replace '(NN something)' are taken to fill the missing parameter of the *search phrase*. In this case, 'batter' is resolved as *noun1* in the relationship *batter***in***refrigerator*. This is because 'batter' is determined to be the head noun of the matching phrase '(DT the) (JJ

parsed *web query*:

```
    (VP (VB place)
        (NP (NN something))
        (PP (IN in) (NP (DT the) (NN refrigerator))))
```

parsed web *sample*:

```
... (VP (VB place)
        (PP (IN for) (NP (JJ several) (NNS hours)))
        (PP (IN in) (NP (DT the) (NN refrigerator)))))))
```

parsed web *sample*:

```
... (NP (DT a) (JJ cool) (, ,) (JJ dark) (NN place;))))
(VP (AUX do) (RB not) (VP (VB store)
    (PP (IN in) (NP (DT the) (NN refrigerator))))) (. .)))
```

Figure 3.4: Examples of results which are eliminated because they do not match a parsed *web query*.

mixed) (NN batter)'. Words are only recorded if they are present as a noun in WordNet. If the noun phrase contains a compound noun found in WordNet, then the compound noun is recorded (i.e. if 'mixed batter' was in WordNet, it would have been stored instead of simply 'batter').

In addition to verifying a match of appropriate words in a *sample*, the parse can also eliminate bad results. Consider the examples in Figure 3.4.

In the first case, the verb phrase does not match the parse of the *web query* due to an extra PP, and therefore the system does not pull out 'for several hours' as *noun1*. In the second case, the parse of the previously mentioned sample (23), the structure is drastically different. In fact, 'place' is not even being used withe same POS. In practice, one can actually eliminate this result before employing the parser by checking that the punctuation matches the web query. In the previously mentioned sample (24), besides 'them' not matching the correct part of speech within the parse, it would also be eliminated because 'them' is not in WordNet as a noun.

At the end of the noun acquisition phase, we are left with frequency counts of nouns being retrieved from a context matching the syntactic structure of a *web query*. The frequency is represented as $f(noun1, \mathbf{R}, noun2)$ in function (3.1), $p_R(n1, n2)$, which is the probability of a noun *n1* being retrieved with noun *n2* based on a query for the relationship, **R**. Additionally, we define $p_R(n2)$ in function (3.2), the probability of any *n1* being retrieved with noun *n2* based on a query for the relationship, **R**. These values along with the results of the other steps are stored in a MySQL relational database[3]. One could trace a relationship probability between nouns back to the Web *samples* which were matched to a *web query*, and even determine the abstract *search phrase* which produced the web query.

$$p_R(n1, n2) = \frac{f(n1, \mathbf{R}, n2)}{f(\mathbf{R})} \tag{3.1}$$

$$p_R(n2) = \frac{f(*, \mathbf{R}, n2)}{f(\mathbf{R})} \tag{3.2}$$

3.1.3 Knowledge Analysis and Discussion

This section presents samples of noun-noun relationships acquired by the system. Relationships were acquired for the nouns listed in Table 3.3. These nouns represent all possible words to fill the *noun2* parameter of a *search phrase*. Tables 3.4, 3.5, 3.6, and 3.7 show the top *noun1*s for the *noun2*s ‘table’, ‘computer’, ‘pocket’, and ‘life’ respectively. The results for these nouns were

[3] http://www.mysql.com

Table 3.3: Nouns for which relations were acquired ($noun2$ parameters of a *search phrase*).

basket	bike	boat	bookcase	bottle
bowl	building	cabin	cabinet	canoe
car	case	ceiling	child	city
company	computer	country	day	desk
drawer	dresser	eye	floor	government
group	hall	hand	hospital	house
jar	kitchen	life	man	mind
part	person	place	pocket	port
refrigerator	restaurant	road	room	shelf
ship	sink	sofa	story	table
thing	time	tree	truck	van
wall	woman	work	world	year

chosen to cover a broad range of concepts from physical entities to abstractions. Refer to functions (3.1) and (3.2) to understand the values in each table.

First, one should observe the $p_R(n2)$ values seen at the bottom of each table. These represent how likely anything was found **on** or **in** the *noun2*. To interpret the results in Table 3.4, one can see that it is a bit more common to find something **on** a table, than **in** a table. On the other hand, Table 3.6 shows that finding something **in** a pocket is much more common than finding something **on** a pocket. This information (the $p_R(n2)$ values) plays a role in using information theory to analyze concepts over WordNet in the concept analysis phase of the framework (explained in the next section).

Looking into the top *noun1*s themselves provides further insight. Many of the results, such as *cards***on***table*, *information***in***computer*, *money***in***pocket*, and *limit***on***life* are expected. It is perhaps more telling to examine result one might not immediately think of. Some unexpected results were due to oversaturation of a particular topic on the Web. Though duplicates of the exact same *samples* were eliminated, many sentences conveying the same information are phrased differently.

Table 3.4: The top 15 *noun1*s acquired for **on** or **in** a ‘table’.

noun1	$p_{\textbf{on}}(n1, `table')$	***noun1***	$p_{\textbf{in}}(n1, `table')$
food	0.006509	data	0.001689
cards	0.002913	cursor	0.001626
book	0.001914	information	0.000756
key	0.001013	image	0.000662
head	0.000922	text	0.000658
bread	0.000760	result	0.000542
face	0.000705	code	0.000542
box	0.000667	record	0.000491
glass	0.000639	row	0.000482
tray	0.000596	point	0.000471
hands	0.000576	entry	0.000460
money	0.000560	key	0.000446
phone	0.000499	value	0.000417
foot	0.000473	table	0.000411
cup	0.000439	content	0.000410
plate	0.000418	page	0.000381
meal	0.000412	values	0.000365
$p_{on}(`table') = 0.0622$		$p_{in}(`table') = 0.0259$	

Table 3.5: The top 15 *noun1*s acquired for **on** or **in** a ‘computer’.

noun1	$p_{\textbf{on}}(n1, `computer')$	***noun1***	$p_{\textbf{in}}(n1, `computer')$
file	0.004156	card	0.001432
cookie	0.003637	disk	0.000855
software	0.001348	information	0.000824
picture	0.001330	disc	0.000791
spyware	0.001083	dvd	0.000745
virus	0.001071	drive	0.000680
program	0.000982	file	0.000619
image	0.000738	problem	0.000510
information	0.000730	data	0.000447
video	0.000693	hard drive	0.000438
photo	0.000687	virus	0.000353
folder	0.000675	cdrom	0.000331
password	0.000653	ram	0.000315
music	0.000574	hardware	0.000266
hands	0.000528	stick	0.000258
server	0.000524	cookie	0.000254
life	0.000463	diskette	0.000222
$p_{on}(`computer') = 0.0447$		$p_{in}(`computer') = 0.0213$	

Table 3.6: The top 15 *noun1*s acquired for **on** or **in** a ‘pocket’.

noun1	$p_{\textbf{on}}(n1, `pocket')$	***noun1***	$p_{\textbf{in}}(n1, `pocket')$
hand	0.000366	money	0.003146
pressure	0.000319	hand	0.003040
burden	0.000245	phone	0.002371
money	0.000180	cash	0.000729
strain	0.000178	device	0.000641
logo	0.000107	ball	0.000484
dent	0.000095	key	0.000447
hands	0.000087	card	0.000444
phone	0.000081	heart	0.000361
name	0.000051	sheet	0.000340
hole	0.000051	hands	0.000307
key	0.000038	player	0.000298
creed	0.000038	dollar	0.000295
patch	0.000038	ipod	0.000282
stress	0.000036	pedometer	0.000252
design	0.000036	camera	0.000243
hospital	0.000036	coin	0.000233
$p_{on}(`pocket') = 0.0036$		$p_{in}(`pocket') = 0.0327$	

Table 3.7: The top 15 *noun1*s acquired for **on** or **in** ‘life’.

noun1	$p_{\textbf{on}}(n1, `life')$	***noun1***	$p_{\textbf{in}}(n1, `life')$
value	0.003295	goal	0.000907
price	0.001712	god	0.000781
hand	0.001168	people	0.000743
focus	0.000776	there	0.000633
limit	0.000758	love	0.000610
price tag	0.000619	someone	0.000554
years	0.000517	genius	0.000466
spin	0.000424	christ	0.000386
emphasis	0.000418	priority	0.000375
calling	0.000402	order	0.000324
damper	0.000374	much	0.000304
strain	0.000342	purpose	0.000302
call	0.000271	balance	0.000279
contract	0.000271	excitement	0.000250
perspective	0.000253	joy	0.000243
impact	0.000228	soul	0.000230
monetary value	0.000202	person	0.000216
$p_{on}(`life') = 0.0207$		$p_{in}(`life') = 0.0228$	

For example, *hospital***on***pocket* was frequent due to many mentions for the show House, in which ”Princeton Plainsboro Hospital” appears on a clothing pocket. Of course, not many things are found **on** a ‘pocket’, as shown by $p_{on}(‘pocket')$, so $p_{on}(‘hospital', ‘pocket')$ is also low compared to that of more common relationships like $p_{on}(‘money', ‘pocket')$. Additional unexpected results from oversaturation on the Web included *page***in***table*, which was due to text describing how to design a web page by placing the whole page in a table (a table in html), as well as *hand***on***life* which was due to mentions of God’s hand being on someone’s life. Biblical references in general were common; *baby***in***basket* was also strong.

Other results one might question may be related to implementation choices. The decision was made to keep all forms of words found in WordNet. Most plural words are converted to their singular form, but words such as ‘hand’ and ‘hands’ appear in separate forms in WordNet and were thus treated separately during noun acquisition. Examples of this from the tables include *hands/hand***on***pocket* and *disk/disc***in***computer*. For the next step of our processing, which utilizes WordNet concepts, senses of all forms of a word will be considered. Overall, the most frequent relationships found (those that contained the highest pR value) over all possible *noun2*s were *head***on***desk* and *eggs***in***basket*.

A key issue with the system up to this point is the ambiguity among noun-noun relationships. Although our minds have no trouble disambiguating most of the *noun1*s seen here, a computer algorithm may have more trouble. In fact, only four of the thirty *noun1*s listed for being on or in a ‘table’ (shown in Table 3.4) are monosemous according to WordNet: ‘cards’, ‘tray’, ‘data’, and ‘cursor’. It would be a helpful for the system to be able to determine the senses of the nouns, or more broadly, to induce the types of concepts that are often in these relationships. This idea may be seen by observing features of Table 3.4; most nouns listed with the **on** relationship can be conceptualized as physical objects, while those for the **in** relationship are mostly abstractions.

3.2 Method of Analysis over WordNet

This section describes our method of inducing information about concepts from information about nouns. There are two key motivations for the concept analysis. The first is to handle noise created from biased or inaccurate results of the web. The second motivation is an assumption that *CSK* is about concepts rather than ambiguous words. Using the knowledge that keys are kept in one's pocket, we know we are talking about the concept associated with *[key-1]*:"metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated" rather than concepts associated with other senses of 'key' such as *[key-2]*:"something crucial for explaining". Ultimately, we take knowledge of the form: *noun1* is [on | in] *noun2* and induce knowledge of the form below.

concept1 is [on | in] *noun2*

Figure 3.5 shows the general process through the framework of acquiring noun-noun relations from the Web to the analysis inducing concept-noun relations described in this section. Overall, we aim to induce the types of concepts that are often found on or in a given noun.

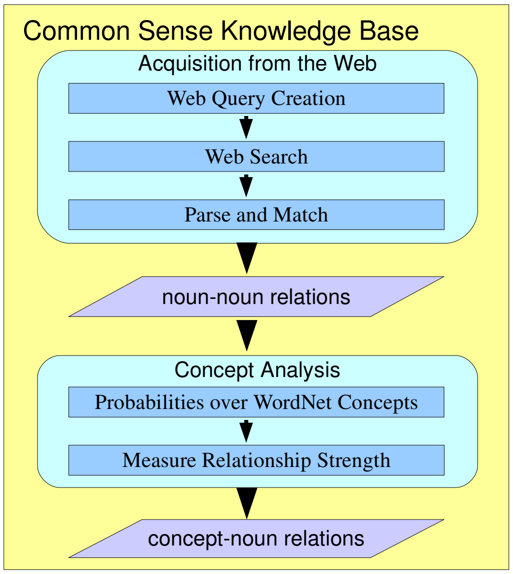


Figure 3.5: The steps taken within the framework for acquiring common sense knowledge.

3.2.1 WordNet Concepts

To begin the concept analysis, probabilities are derived over WordNet concepts from the existing probabilities over nouns. One should recall from Section 1.1 that synsets are the representatives of concepts in WordNet. A synset is a group of word-senses that have the same meaning (Miller et al., 1993). For example, *[batter-1, hitter-1, slugger-1, batsman-1]* is a synset with the meaning "(baseball) a ballplayer who is batting". Recall function (3.1), $p_R(n1, n2)$, the joint probability that *n1* is returned to a query for the relationship, **R**, with *n2*. Below, $P_R^{syn}(c1, n2)$ distributes p_R to WordNet synsets, where $lemma$ represents the word of a noun sense, $n1_s$, $senses$ returns the number of senses of a word, and $c1$ is a concept / synset in WordNet.

$$P_R^{syn}(c1, n2) = \sum_{n1_s \in c1} \frac{p_R(lemma(n1_s), n2)}{senses(lemma(n1_s))}$$

$$P_R^{sub}(c1, n2) = P_R^{syn}(c1, n2) + \sum_{c \in descs(c1)} P_R^{syn}(c, n2) \tag{3.3}$$

$$P_R^{sub}(c1) = \sum_{n2} P_R^{sub}(c1, n2) \tag{3.4}$$

Function (3.3), $P_R^{sub}(c1, n2)$ implements the idea that a concept subsumes all concepts below it (hyponyms) in the WordNet Ontology. In this function, $desc$ returns the set of descendants (all direct or indirect hyponyms) within the WordNet ontology. For example, *[money-3]* is a *[currency-1]*, so $P_R^{sub}($*[currency-1]*$, n2)$ receives $P_R^{syn}($*[money-3]*$, n2)$ among others. Also defined for a single concept parameter in function in (3.4) P_R^{sub} is the probability that $c1$ is in relationship R with any $noun2$.

A couple properties of P_R^{sub} are listed in Figure 3.6. First, the root concept in WordNet, *[entity-1]*, subsumes the probability of all possible $c1$, and therefore it is always in a relationship a $noun2$. Similarly, it could also be said that $P_R^{sub}($*[entity-1]*$, n2) = p_R(n2)$ for all $n2$. Another important property is that a concept's P_R^{sub} is always greater than or equal to that of its descendants. This property means that P_R^{sub} is always biased toward the more general concepts. Although this bias is correct based on the idea that a concept should subsume its hyponyms' probability, P_R^{sub} should not be used directly to measure relationship strength. The next section describes how it is used more appropriately.

$$P_R^{sub}(\textit{[entity-1]}) = 1 = \sum_{n2} p_R(n2) = \frac{f(*, \mathbf{R}, *)}{f(\mathbf{R})}$$

$$P_R^{sub}(c1, n2) \geq P_R^{sub}(c, n2), \forall\, c \in descs(c1)$$

Figure 3.6: Properties of P_R^{sub}.

3.2.2 Measuring Relationship Strength

From equation 3.3, one now has a probability, $P_R^{sub}(c1, n2)$, for the relationship *concept1* is [in | on] *noun2*. However, this probability by itself is not an accurate measure of relationship strength as it favors more general concepts and contains noise from considering multiple senses of ambiguous nouns equally. This can be a problem when a general sense of a word is preferred over a specific sense. Consider two concepts of ‘change’ in Figure 3.7; *[change-8]*: “coins of small denomination regarded collectively” is clearly the better choice for being in a pocket, but *[change-3]*: “the action of changing something” has a much higher P_R^{sub} simply because it subsumes more concepts. The approach turns to information theory to develop a metric for measuring the relationship strength.

Pointwise mutual information (PMI) is commonly used to measure the strength of associations (Church & Hanks, 1989). It computes a ratio between the joint probability of observing two elements and the probability of observing both elements with independence. PMI results in positive values when elements occur together more than expected from chance, zero when they occur together according to chance, and a negative value when the elements occur less frequently than expected. Below, pointwise mutual information is adapted to measure the specific information

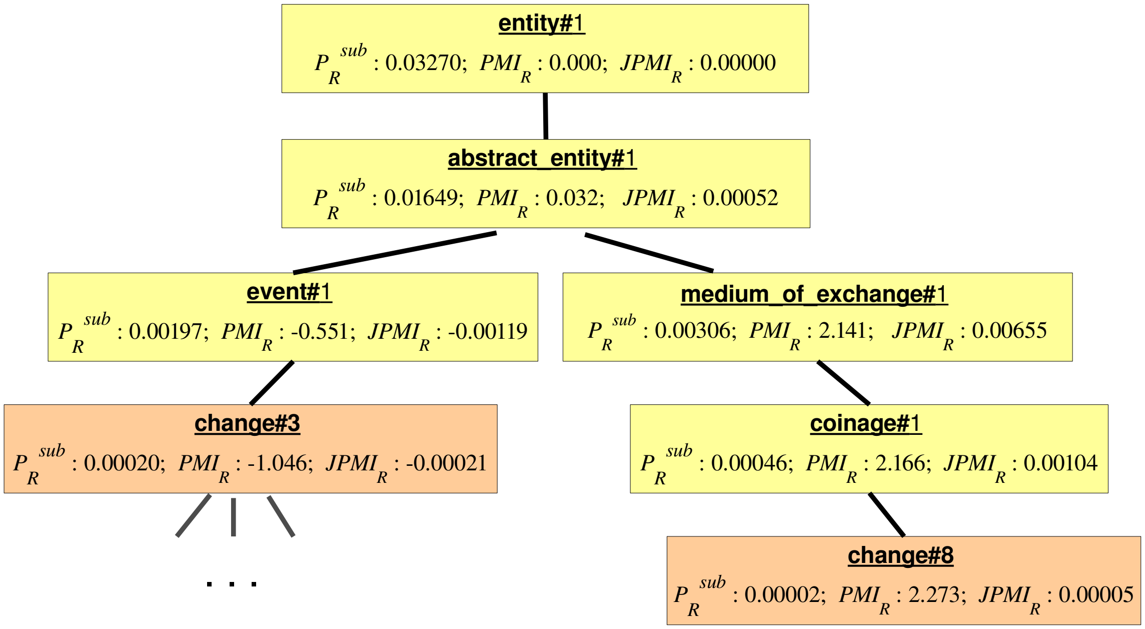

entity#1
P_R^{sub} : 0.03270; PMI_R : 0.000; $JPMI_R$: 0.00000
abstract_entity#1
P_R^{sub} : 0.01649; PMI_R : 0.032; $JPMI_R$: 0.00052
event#1
P_R^{sub} : 0.00197; PMI_R : -0.551; $JPMI_R$: -0.00119
medium_of_exchange#1
P_R^{sub} : 0.00306; PMI_R : 2.141; $JPMI_R$: 0.00655
change#3
P_R^{sub} : 0.00020; PMI_R : -1.046; $JPMI_R$: -0.00021
coinage#1
P_R^{sub} : 0.00046; PMI_R : 2.166; $JPMI_R$: 0.00104
. . .
change#8
P_R^{sub} : 0.00002; PMI_R : 2.273; $JPMI_R$: 0.00005

Figure 3.7: A portion of the WordNet hierarchy with various figures for the relationship: entities in a 'pocket'. A connection between concepts may not indicate direct hypernymy (i.e. not all concepts in the path from concept to entity are shown).

between a WordNet concept, $c1$ and a *noun2*, $n2$.

$$PMI_R(c1, n2) = \log \frac{P_R^{sub}(c1, n2)}{P_R^{sub}(c1) p_R(n2)} \tag{3.5}$$

Conceptually, PMI_R compensates for the bias towards general concepts because concepts which occur with one or only a few $noun2$s are more informative. Since general concepts subsume many others, they are more likely to occur with many $noun2$s.

Finally, while one might expect PMI to be best in application, for the sake of examination, it is also useful to know the informative general concepts that are involved in the relationship; i.e. induce that *[foodstuff-1, food_produce-1]* is commonly found in a 'refrigerator' if many of its subconcepts are are also found in a 'refrigerator'. Therefore, joint probability-weighted PMI is also defined below as $JPMI_R$. In this case, relationship strength is measured as the probability of a concept (and concepts subsumed by the concept) being in the relationship and the amount mutual information. This balances the metric to neither favor the most specific concepts nor the most general.

$$JPMI_R(c1, n2) = P_R^{sub}(c1, n2) * PMI_R(c1, n2) \tag{3.6}$$

Both $PMI_R($*[entity-1]*$, n2)$ and $JPMI_R($*[entity-1]*$, n2)$ are equal to zero for all $n2$. This property respects intuition since *[entity-1]*, the most general concept, should be in relationships exactly according to chance; Knowledge of its existence in a relationship is neither informative nor complementary. This becomes an important property when applying the *CSKB*, as one can simply observe concepts with PMI greater than *[entity-1]*.

These measures are similar to Resnik's information-theoretic approach to compute selectional preferences (Resnik, 1996). He used occurrences of nouns to find classes in WordNet which should subsume the nominal arguments of a verb. However, his method was based on relative entropy, rather than mutual information, which is asymmetric. Asymmetric functions were found to be less appropriate as a measure of relationship strength (i.e. using relative entropy, *[entity-1]* would not end up with a value of zero).

Finally, the idea of inheritance is incorporated. If *[foodstuff-1, food_produce-1]* has a strong value for being *in* a 'refrigerator', then *[egg-2, eggs-1]* should also have a strong value. Thus, in practice we say the strength of a relationship between a concept $c1$ and a noun $n2$ is given by maximum value from all ancestors, $ancs$, of the concept $c1$:

$$RS_{meas_R}(c1, n2) = \max_{c1' \in ancs(c1)} meas_R(c1', n2) \tag{3.7}$$

where $meas_R$ is either PMI_R or $JPMI_R$.

3.2.3 Sample Concept-Noun Relationships

This section presents various sample outputs from the concept analysis. WordNet version 3.0 was used in order to take advantage of the latest updates and corrections to the noun ontology. The possible nouns for the $noun2$ parameter were those given in Table 3.3, while the tables of examples follow the same four nouns we sampled for the noun-noun relationships: ‘table’, ‘computer’, ‘pocket’, and ‘life’.

Tables 3.8, 3.9, 3.10, and 3.11 show the $JPMI$, PMI, and P_R^{sub} values of concepts for the nouns we chose as in our samples. The results are sorted by $JPMI$, showing concepts with the greatest values, and the concept with the least value. In Table 3.8 we also show the value for *[entity-1]*, which is always zero (for other *noun2*s as well), indicating that it is neither informative nor uninformative. Notice the $P_R^{sub}(c1, n2)$ value is equal to the probability that anything is found **on** or **in** a ‘table’ ($p_{on}('table')$ or $p_{in}('table')$ from Table 3.4). Similarly, $P_R^{sub}(n2)$ is always 1 for *[entity-1]*, which subsumes all other concepts in WordNet.

Several ideas are worth noting when examining these tables. Consider concepts found **on** a ‘table’ (Table 3.8). As expected, we see many concepts of *[physical_entity-1]* while *[abstraction-*

Table 3.8: The top *concept1*s, according to $JPMI_R$ found **on** or **in** a 'table'.

concept1	***JPMI***$_R$***(c1, n2)***	***PMI***$_R$***(c1, n2)***	P_R^{sub}***(c1, n2)***	P_R^{sub}***(c1)***
$R =$ **on**				
matter-3	0.018172	1.928070	0.0094	0.0398
food-1, nutrient-1	0.012559	2.733966	0.0046	0.0111
substance-7	0.011516	2.384117	0.0048	0.0149
physical_entity-1	0.010097	0.270794	0.0373	0.4970
solid-1	0.009516	2.600902	0.0037	0.0097
food-2, solid_food-1	0.009487	2.700154	0.0035	0.0087
food-3, food_for_thought-1, intellectu...	0.008303	3.826861	0.0022	0.0025
instrumentality-3, instrumentation-1	0.004925	0.444350	0.0111	0.1310
container-1	0.004264	1.332153	0.0032	0.0204
artifact-1, artefact-1	0.002802	0.154696	0.0181	0.2617
...	...	...	...	...
entity-1	0.000000	0.000000	0.0622	1.0000
...	...	...	...	...
abstraction-6, abstract_entity-1	-0.007279	-0.267157	0.0272	0.5273
$R =$ **in**				
cursor-1, pointer-3	0.008082	4.925300	0.0016	0.0021
communication-2	0.007726	1.532921	0.0050	0.0673
abstraction-6, abstract_entity-1	0.007718	0.444117	0.0174	0.4933
indicator-3	0.007320	4.413976	0.0017	0.0030
data-1, information-4	0.004468	4.411651	0.0010	0.0018
collection-1, aggregation-1, accumulatio...	0.003684	2.481254	0.0015	0.0103
written_communication-1, written_language	0.003568	2.051087	0.0017	0.0162
datum-1, data_point-1	0.003487	4.004151	0.0009	0.0021
information-2	0.003029	2.636957	0.0011	0.0071
message-2, content-2, subject_matter-1...	0.002880	1.570537	0.0018	0.0238
...	...	...	...	...
entity-1	0.000000	0.000000	0.0259	1.0000
...	...	...	...	...
physical_entity-1	-0.005770	-0.639648	0.0090	0.5427

Table 3.9: The top *concept1*s, according to $JPMI_R$ found **on** or **in** a ‘computer’.

concept1	***$JPMI_R$(c1, n2)***	***PMI_R(c1, n2)***	***P_R^{sub}(c1, n2)***	***P_R^{sub}(c1)***
$R =$ **on**				
communication-2	0.015640	1.261513	0.0124	0.1156
software-1, software_program-1, computer..	0.015326	4.126031	0.0037	0.0048
coding_system-1	0.015202	3.954812	0.0038	0.0055
code-3, computer_code-1	0.015190	3.999468	0.0038	0.0053
written_communication-1, written_language	0.013653	2.315626	0.0059	0.0265
writing-4	0.012324	3.114665	0.0040	0.0102
evidence-2	0.006457	2.942864	0.0022	0.0064
indication-1, indicant-1	0.006111	2.759995	0.0022	0.0073
...	...	...	...	...
physical_entity-1	-0.003105	-0.155517	0.0200	0.4970
$R =$ **in**				
memory_device-1, storage_device-1	0.013926	4.941875	0.0028	0.0043
device-1	0.008196	1.768113	0.0046	0.0640
instrumentality-3, instrumentation-1	0.007254	1.189294	0.0061	0.1258
optical_disk-1, optical_disc-1	0.006438	5.232749	0.0012	0.0015
communication-2	0.005884	1.477582	0.0040	0.0673
magnetic_disk-1, magnetic_disc-1, disk...	0.004933	5.288089	0.0009	0.0011
recording-3	0.004665	4.890266	0.0010	0.0015
artifact-1, artefact-1	0.004513	0.613446	0.0074	0.2262
...	...	...	...	...
physical_entity-1	-0.001351	-0.128009	0.0106	0.5427

6, abstract_entity-1] has the lowest $JPMI$. The opposite is true for the relationship **in** with abstractions being more toward the top *[physical_entity-1]* at the bottom. It is not always the case that things split so clearly between the physical and abstract concepts. Take, for example, concepts found **in** a ‘pocket’ (Table 3.10). Many of those at the top are abstractions, such as *[medium_of_exchange-1, monetary_system-1]* , while there are also physical entities such as *[electronic_equipment-1]*. In the end, it was very uncommon to find a psychological feature in a pocket (which is also a subordinate of abstraction) so we have types of abstractions at the both the top and the bottom.

Table 3.10: The top *concept1*s, according to $JPMI_R$ found **on** or **in** a ‘pocket’.

concept1	***JPMI***$_R$***(c1, n2)***	***PMI***$_R$***(c1, n2)***	P_R^{sub}***(c1, n2)***	P_R^{sub}***(c1)***
$R =$ **on**				
abstraction-6, abstract_entity-1	0.000489	0.220977	0.0022	0.5273
logo-1, logotype-1	0.000476	4.415179	0.0001	0.0014
trademark-2	0.000475	4.401528	0.0001	0.0014
cognition-1, knowledge-1, noesis-1	0.000331	0.748193	0.0004	0.0733
medium_of_exchange-1, monetary_system-1	0.000311	2.245012	0.0001	0.0081
standard-1, criterion-1, measure-5, to...	0.000307	2.214618	0.0001	0.0083
marker-2, marking-1, mark-2	0.000295	2.384777	0.0001	0.0066
system_of_measurement-1, metric-3	0.000292	2.082897	0.0001	0.0092
...	...	...	...	...
object-1, physical_object-1	-0.000428	-0.378613	0.0011	0.4085
$R =$ **in**				
medium_of_exchange-1, monetary_system-1	0.006548	2.141022	0.0031	0.0212
standard-1, criterion-1, measure-5, to...	0.006493	2.121870	0.0031	0.0215
system_of_measurement-1, metric-3	0.006374	2.079083	0.0031	0.0222
measure-2, quantity-1, amount-3	0.004686	1.119645	0.0042	0.0589
currency-1	0.004668	2.346209	0.0020	0.0120
instrumentality-3, instrumentation-1	0.004498	0.681613	0.0066	0.1258
telephone-1, phone-1, telephone_set-1	0.004096	3.982789	0.0010	0.0020
electronic_equipment-1	0.004072	2.709799	0.0015	0.0070
...	...	...	...	...
psychological_feature-1	-0.00229	-0.70585	0.0032	0.162

One might also consider examining the differences between $JPMI$, PMI, and P_R^{sub}. Keep in mind that we introduced $JPMI$ to negate the fact that PMI highly favors specific concepts (as we may want in application). Furthermore, P_R^{sub} is biased toward general concepts. $JPMI$ provides a logical choice for sorting when examining the results, so one gets an idea of the types of concepts in the relationship. From Table 3.9, we see a somewhat general concept at the very top: *[communication-2]*. However, when looking at the PMI values, we see that *[software-1, software_program-1, computer..]* carried much more information. Additionally, recalling Figure 3.7, both PMI and $JPMI$ found *[change-8]*:“coins of small denomination...” stronger for being in a ‘pocket’ when compared with *[change-3]*: “the action...”. P_R^{sub} favors *[change-3]*. One can see

Table 3.11: The top *concept1*s, according to $JPMI_R$ found **on** or **in** ‘life’.

concept1	***JPMI***$_R$***(c1, n2)***	***PMI***$_R$***(c1, n2)***	P_R^{sub}***(c1, n2)***	P_R^{sub}***(c1)***
$R =$ **on**				
abstraction-6, abstract_entity-1	0.010393	0.618902	0.0168	0.5273
worth-2	0.005842	3.527883	0.0017	0.0069
quality-1	0.005178	2.232304	0.0023	0.0238
attribute-2	0.005115	1.126399	0.0045	0.1003
value-2	0.004885	3.480612	0.0014	0.0061
time_period-1, period_of_time-1, period	0.003410	2.672110	0.0013	0.0097
psychological_feature-1	0.003368	0.620400	0.0054	0.1703
fundamental_quantity-1, fundamental_measur...	0.003360	2.616400	0.0013	0.0101
...	...	...	...	...
physical_entity-1	-0.005450	-1.323202	0.0041	0.4970
$R =$ **in**				
abstraction-6, abstract_entity-1	0.007236	0.465676	0.0155	0.4933
attribute-2	0.005301	1.103654	0.0048	0.0980
psychological_feature-1	0.004907	0.776551	0.0063	0.1617
cognition-1, knowledge-1, noesis-1	0.003654	1.048979	0.0035	0.0738
state-2	0.002846	1.236812	0.0023	0.0428
content-5, cognitive_content-1, mental...	0.002825	1.178329	0.0024	0.0465
person-1, individual-1, someone-1, som...	0.002258	0.616498	0.0037	0.1047
spiritual_being-1, supernatural_being-1	0.002153	2.361831	0.0009	0.0078
...	...	...	...	...
physical_entity-1	-0.005101	-0.643804	0.0079	0.5427

that PMI tends to favor more specific concepts such as *[coinage-1]*. Furthermore, one can also see from Figure 3.7 how inheritance would work, where, in the case of $JPMI$ *[change-8]* would inherit from *[medium_of_exchange-1]* since it has the maximum $JPMI$ over all its ancestors. While it is clear that P_R^{sub} is biased too heavily toward the general concepts, we leave it up to the application to make a judgment on whether PMI or $JPMI$ is best as a measure of relationship strength.

3.3 Evaluation

The evaluation focuses on the applicability of the acquired *CSKB*. It was chosen to apply the knowledge to the task of *word sense disambiguation* (*WSD*), annotating the correct sense of an ambiguous word within a sentence. *WSD* is a fundamental task of semantics in the field of natural language processing. A description of the *WSD* system and experimental corpus follows.

3.3.1 Disambiguation System

The *CSKB* is not intended to be used by itself for disambiguation. It would be far from accurate to assume the sense of a noun can be disambiguated simply by observing its relationship with one other noun in the sentence. For example, one of the test sentences incorporated the relationship *note***in***pocket*. Multiple senses of note are likely to be found in a pocket (i.e. the senses referring to "a brief written record", "a short personal letter", or "a piece of paper money"). In other cases, a relationship may not be found for any sense of a target word. Therefore, the knowledge is intended to be used as a reference, consulted by a disambiguation system.

The knowledge is integrated into a state of the art "all-words" word sense disambiguation algorithm. These algorithms are considered minimally supervised, because they do not require specific training data that is designed for instances of words in the testing data. In other words, these systems are designed to handle any word they come across. The *CSKB* can supplement such a system, because the data can be acquired automatically for an unlimited number of nouns, assuming limitless web query restrictions.

The basis of the disambiguation system is the publicly available GWSD system (Sinha & Mihalcea, 2007). Sinha and Mihalcea report higher results on the Senseval-2 (Edmonds & Cotton, 2001) and Senseval-3 (Snyder & Palmer, 2004) datasets than any of the participating minimally-supervised systems. Additionally, GWSD is compatible with WordNet 3.0 and its output made it easy to integrate the knowledge. Sense predictions from four different graph metrics are produced, and knowledge is incorporated as another prediction within a voting scheme.

GWSD is supplemented by included suggestions from the *CSKB* for relationships found in a sentence. First, potential relationships are discovered by matching the phrase "$in|on$ *det noun2*" within the sentence, anywhere after the target noun (taken as $noun1$). Recall $RS_{meas_R}(c1, n2)$ is the relationship strength given by a metric mentioned previously (functions (3.5) and (3.6); PMI_R and $JPMI_R$). As exemplified previously from the relationship *note***in***pocket*, in some cases one would like suggestions of multiple senses and in others none. With this in mind, our suggestions are based on two criteria:

$$RS_{meas_R}(c1, n2) > RS_{meas_R}(\textit{(entity-1), n2)}$$

$$RS_{meas_R}(c1, n2) > \max_{c \in senses(c1)} RS_{meas_R}(c, n2) * mp$$

These criteria, respectively, insure that each suggestion is informative and that it is not notably weaker than the top suggestion. If no senses match this criteria than no senses are suggested. The variable mp represents a minimum percentage of the maximum strength over all senses if $c1$, $senses(c1)$. In our experiments, mp is set to 0.75.

Table 3.12: List of nouns in our testing corpus which fill the $noun2$ constituent in a *search phrase*.

basket	boat	bookcase	bottle	bowl
cabin	cabinet	canoe	car	ceiling
city	desk	drawer	dresser	floor
house	jar	kitchen	pocket	refrigerator
road	room	shelf	ship	sink
sofa	table	truck	van	wall

Considering the role of the *CSKB* as a reference, in some cases one would like suggestions of multiple senses and in others none. The $P_c(cncptA, \mathbf{R}, nounB)$ value is found for each sense of a target noun instance in the corpus, ($cncptA$ is the WordNet concept that corresponds to a sense of the target noun). The $nounB$ is chosen by matching the phrase "$in|on$ *det nounB*" within the sentence. The system suggests all senses with a P_c value greater than 0.75 of the maximum P_c value over all senses. If no senses have a P_c value then no senses are suggested.

During voting, tallies of predictions (from GWSD) and suggestions (from the *CSKB*) are taken for each sense of a noun. Ties are broken by choosing the lowest sense number among all those involved in the tie. Note that this is different than choosing the most frequent sense (i.e. the lowest sense number from *all* senses), in that only the top predicted senses are considered. This same type of voting is used with and without the *CSKB* suggestions.

3.3.2 Experimental Corpus

A goal of this dissertation is to acquire data which can be applied to semantic interpretation problems. This experiment focused particularly on the difficult problem of *word sense disambiguation*. Due to the lack of sense tagged data, there was no existing annotated corpus with instances of all the nouns in Table 3.12 as prepositional complements. This was not surprising considering one of the reasons that minimally supervised approaches have become more popular is that they do not require hand-tagged training data (Mihalcea, 2002; Diab, 2004; McCarthy et al., 2004).

A corpus of sentences from Wikipedia was created which contained the phrase "$in|on$ *det lemma*", where *det* is a determiner or possessive pronoun, *lemma* is a noun from Table 3.12, and $in|on$ is a preposition for either relationship as described in Section 3.1. Sentence (26) is an example from the corpus where the knowledge from 'pocket' can be applied to disambiguate 'key'.

(26) *Now Tony's **key** to the flat is in the pocket of his raincoat, so on returning to his flat some time later he realizes that he cannot get inside.*

The corpus[4] contained a total of 342 sentences, with one target noun annotated per sentence. The target nouns were selected in order to fill the *noun1* constituent in the relationship *noun1***R***noun2*, and they were assigned all appropriate WordNet 3.0 senses. Considering the fine-grained nature of WordNet (Ide & Wilks, 2006), 26.3% of the instances were annotated with multiple senses. The corpus was also restricted to only include polysemous nouns, or nouns which had an additional sense beyond the senses assigned to it.

Inter-annotator agreement was used to validate the corpus itself. Because the corpus was built by an author of the work, a non-author to re-annotate the corpus without knowledge of the original annotations. This second annotator was told to choose all appropriate senses just as did the original

[4] available at: http://eecs.ucf.edu/~hschwartz/CSK/

Table 3.13: Experimental corpus data for each relationship (***on, in***).

	on	**in**	**both**
insts	131	211	**342**
agree	0.799	0.808	**0.805**
$\mathbf{F}_h$	0.847	0.919	**0.892**
$\mathbf{F}_{rnd}$	0.282	0.272	**0.276**
$\mathbf{F}_{MFS}$	0.710	0.678	**0.690**

insts: number of annotated instances; **agree**: inter-annotator agreement; **F** values: h: human annotation, rnd: random baseline, MFS: most frequent sense baseline.

annotator. Agreement was calculated as:

$$\textbf{agree} = \left(\sum_{i \in C} \frac{|S1_i \cap S2_i|}{|S1_i \cup S2_i|} \right) \div 342$$

where $S1$ and $S2$ are the two sets of sense annotations, and i is an instance of the corpus, C.

The agreement and other data concerning corpus annotation can be found in Table 3.13. As a point of comparison, the Senseval 3 all-words task had a 75% agreement on nouns Snyder & Palmer (2004). A second evaluation of agreement was also done. The non-author annotations were treated as if they came from a disambiguation system, and the original annotations were used as a key. This gives a human upper-bound of performance, and it is shown as $\mathbf{F}_h$ in Table 3.13. Just as the automatic system handled tie votes, when the second annotator chose multiple senses for one word, the lowest sense number among those chosen was used. F scores here and within the results were calculated as $\mathbf{F} = 2 * \frac{precision*recall}{precision+recall}$. As is standard for word sense disambiguationSnyder & Palmer (2004); Edmonds & Cotton (2001), precision was calculated as the number correct out of the number attempted, while recall was the number correct out of all instances. In our experiments, precision was equal to recall (indicating all instances were attempted) unless otherwise indicated.

Table 3.14: F-scores on our experimental corpus without(**w/o**) and with(**w/**) the *CSKB*.

	meas	**on**	**in**	**both**	***ties***
w/o *CSKB*	-	0.626	0.687	**0.664**	*37*
w/ *CSKB*	PMI	0.634	0.744	**0.702**	*106*
	$JPMI$	0.626	0.744	**0.699**	*60*

ties: number of instances where tie votes occurred.

3.3.3 Results

The primary purpose of the evaluation was to validate that the *CSK* is applicable to the semantics of natural language processing. Therefore, the results are presented from running the disambiguation system on the experimental corpus with and without suggestions from the *CSKB*.

Table 3.14 shows the F scores when running the disambiguation system with and without the suggestions from the *CSKB*. The incorporation of the *CSKB* significantly improved results when using either the PMI and $JPMI$ measures of relationship strength. For the top F score, there was an error reduction of 11.3%. While running GWSD, it was discovered that the *indegree* metric by itself actually performed stronger on our corpus than the combination of all four graph metrics. This was not surprising considering Sinha and Mihalcea Sinha & Mihalcea (2007) found the indegree metric by itself to perform only slightly below a combination of metrics on Senseval data. Therefore, results using the *indegree* metric by itself are also presented in Table 3.15, with and without the *CSKB*. In this case using the *CSKB* measured with PMI resulted in our best results with an F-score of 0.744 and a 20.5% error reduction over the GWSD with the *indegree* metric alone. Additionally, the results with the *CSKB* exceed the F_{MFS} baseline of 0.690, a point which GWSD alone did not pass.

Table 3.15: F-scores when applying GWSD with only the *indegree* graph metric.

	meas	**on**	**in**	**both**	***ties***
w/o *CSKB*	-	0.634	0.697	**0.673**	*0*
w/ *CSKB*	PMI	0.687	0.773	**0.740**	*168*
	$JPMI$	0.649	0.768	**0.722**	*84*

A couple additional experiments were run with regards to how the system handled ties. Note that for the results above, the chosen sense was taken from the predictions and suggestions during ties. When the most frequent sense is chosen for instances that had ties, the top F score dropped to 0.731 using the *indegree* graph metric with the *CSKB*. Additionally, a test with no predictions made for tie votes found a precision of 0.770 on the 134 instances that did not have a tie for top votes (also using the indegree metric with the *CSKB*). Thus, the system was more likely to miss instances with ties, indicating those instances were more difficult but that the *CSKB* was still helpful. All results supported our goal of acquiring *CSK* that was applicable to natural language processing.

3.4 Discussion

Experiments found the acquired *CSKB* to be useful when incorporated into a word sense disambiguation system. The experiments focused on a type of *CSK* which is applicable to natural language processing and describes what is often found on or in something. The approach searched the Web with queries constructed automatically by filling parameters of predetermined search phrases. Through a unique aspect of the approach, results from the Web were matched with constituents of search phrases by incorporating a statistical parser, which eliminated Web phrases that did not contain the intended syntactic structure.

Under the assumption that *CSK* describes information about concepts rather than information about ambiguous words, the system also performs a novel concept analysis over WordNet. The analysis, based on applying information-theory over the WordNet ontology, sought to automatically induce the types of concepts that are often found in or on something. Thus, an attempt was made to take *CSK* about ambiguous words and turn it into knowledge about concepts. Although samples of the results of the concept analysis seemed to follow intuition, our goal was to acquire applicable common sense knowledge. Therefore, the resulting knowledge was evaluated through its application in a state of the art word sense disambiguation system.

The evaluation found that integrating our acquired *CSK* into an existing disambiguation system significantly improved results, with a 20.5% error reduction for the top results. While this paper focused broadly on examining and validating our approach to acquire common sense knowledge as a whole, further research may investigate specific aspects of the approach. Examples of such work may include quantifying the helpfulness of the parser, searching for other types of *CSK*, or using the concept analysis that was introduced to analyze data describing other lexical relationships. One should keep in mind, that *noun2* is not disambiguated, so these concepts may be for any sense of 'pocket'. Still, a few concepts appear which may not belong *in* any of the senses of 'pocket'. *Person-1*, for example, is very common to appear with a high P_c, because many nouns for objects, have senses that are also a description for a type of person, such as *computer-2* (gloss: "an expert at calculation (or at operating calculating machines)"). Finally, a vast amount of future work lies in the application of common sense knowledge to additional problems in language and other domains.

4 WEB SELECTORS AS A MEANS TO DYNAMICALLY ACQUIRE KNOWLEDGE

A goal of the methods to acquired knowledge within this dissertation is that the knowledge can be successfully applied to semantic interpretation problems. In this chapter the focus is on acquiring knowledge dynamically from the Web for a given target word and sentence as part of a word sense disambiguation approach. Section 2.2.3 introduced the term selectors when discussing the work of Lin (1997). Selectors are words which take the place of an instance of a target word within its local context. They serve for the system to essentially learn the areas or concepts of WordNet that the sense of a target word should be within (Schwartz & Gomez, 2008).

4.1 Method for Acquiring Web Selectors

As a general definition, selectors are words which take the place of an instance of a target word within its local context. In (Lin, 1997), dependency relationships over a small corpus were used for matching local context to find selectors. However, the task of producing a dependency parse database of the Web is not currently practical. In turn, one must search for text as local context. For example, in sentence (27) the local context for ‘strikers’ would be composed of “he addressed the” and “at the rally.”.

(27) *He addressed the strikers at the rally.*

This dissertation introduces the idea of using selectors of other words in a sentence in addition to selectors of the target, the word being disambiguated (Schwartz & Gomez, 2008). Words taking the place of a target word are referred to as *target selectors* and words which take the place of other words in a sentence are referred to as *context selectors*. *Context selectors* are classified further based on their part of speech:

noun context selectors nouns which are found to replace other nouns of the sentence.

verb context selectors verbs which are found to replace other verbs in the sentence.

adjective context selectors adjectives which replace other adjectives in the sentence.

pro context selectors nouns which replace pronouns and proper nouns.

In (27), if ‘striker’ was the target word, the *verb context selectors* would be verbs replacing ‘addressed’, the *noun context selectors* would be nouns replacing ‘rally’, and the *pro context selectors* would be nouns replacing ‘he’.

In practice, selectors are acquired for all appropriate parts of speech. Whether the selectors are used as *target selectors* or *context selectors* depends on the target word with which they are being applied. Thus, one process can be used to acquire all *noun*, *verb*, *adjective*, and *adverb selectors*. Additionally, noun selectors can be acquired for pronouns and proper nouns (referred to as “pro” selectors). These are nouns found to replace a pronoun or proper noun within their local context, and are only used as context selectors since a pronoun is never a target.

i Shorten to a size of 10 words.
ii Remove end punctuation, if not preceded by *.
iii Remove front punctuation, if not proceeded by *.
iv Remove determiners (*the, a, an, this, that*) preceding *.
v Remove a single word.

Figure 4.1: The steps taken in order to truncate a query for Web selectors.

The first step in acquisition is to construct a query with a wildcard in place of the target. In the running example, with ‘address’ as the target, the query is “he * the strikers at the rally.” Yahoo! Web Services[1] provides the functionality for searching the web for phrases with wildcards. Selectors are extracted from web search results by matching the words which take the place of the wildcard. All words not found in WordNet under the same part of speech as the target are thrown out as well as phrases longer than 4 words or those containing punctuation. WordNet is also used to determine if the phrase is a compound and the base morphological form of the head word. Results containing head words not found in WordNet are filtered out. Finally, the list of selectors is adjusted so no single word takes up more than 30% of the list.

The Web is massive, but unfortunately it is often not large enough to find results when querying with a whole sentence as context. Therefore, the query is truncated and the search is repeated until a stop condition is met. The steps in Figure 4.1 are followed where the final step is repeated until a goal for the number of selectors was reached or the query becomes too short.

When removing a single word, the algorithm attempts to keep the * in the center. Figure 4.2 demonstrates the loop that occurs until a stop condition is met: enough selectors are found or the query has reached a minimum size. Since a shorter query should return the same results as a longer query, the selectors from longer query results are filtered out of the shorter results. It is important

[1] http://developer.yahoo.com/search/

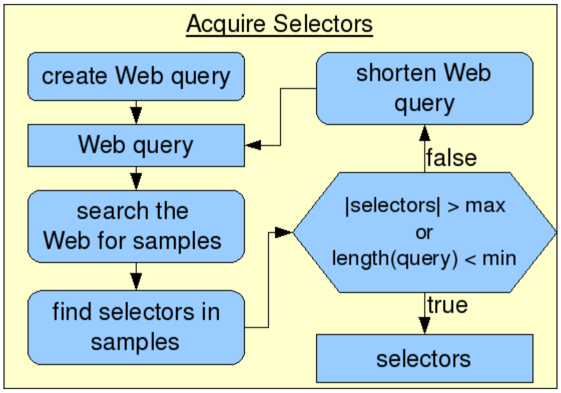

Acquire Selectors
create Web query
Web query
search the Web for samples
find selectors in samples
shorten Web query
false
|selectors| > max or length(query) < min
true
selectors

Figure 4.2: The iterative process of acquiring selectors and truncating Web search queries.

that the criteria to continue searching is based on the number of selectors and not on the number of samples, because many samples fail to produce a selector. The truncation follows the idea of using a decreasing window size as context (Martínez et al., 2006; Yuret, 2007).

4.2 Knowledge Analysis and Discussion

Figure 4.3 lists selectors retrieved for sentence (27). In the first couple sets of selectors, the context is larger, and therefore less selectors are retrieved. However, those selectors do seem to be similar to the correct sense of 'striker', *striker-3* (gloss: "an employee on strike against an employer"). As the the query gets shorter, the number of selectors increases, but the similarity between the selector and the target word becomes less strong, such as with 'member' and 'council'. In some cases, the similarity is not clear, such as with 'Saturday'.

Results can vary quite a bit from sentence to sentence, so an examination of actual selectors is a bit limited. Therefore, statistics are provided on the occurrences of selectors acquired for all sentences in the SemEval 2007 coarse-grained all-words task (Navigli et al., 2007). Listed as the

*He addressed the * at the rally*

crowd:1

*He addressed * at the rally*

student:1, supporter:2

*He addressed * at the*

Council:1, Muslim:1, Saturday:1, Ugandan:1, analyst:2, attendee:20, audience:3, class:2, consumer:1, council:1, delegate:64, diplomat:2, employee:2, engineer:1, fan:1, farmer:1, globalization:1, graduate:5, guest:2, hundred:3, investor:1, issue:1, journalist:9, lawmaker:11, legislator:1, member:6, midshipman:1, mourner:1, official:2, parliamentarian:1, participant:17, patient:1, physician:18, reporter:8, sailor:1, secretary:1, soldier:3, staff:3, student:20, supporter:8, thousand:3, today:2, trader:1, troops:2, visitor:1, worker:1

*He * the strikers at the*

treat:2

*He * the strikers at*

get:1, keep:1, price:1, treat:1

Figure 4.3: Lists of selectors for the target words ‘striker’ and ‘address’ returned by corresponding web queries.

column headings of Table 4.1, selectors are acquired for five parts of speech (*pro* is actually a combination of two parts of speech: pronoun and proper noun). The data in Table 4.1 is based on results from acquiring selectors for our experimental corpus. The information presented is described in the bottom portion of the table.

The selector acquisition data provides useful insights. In general, *% w/ sels* was low from being unable to find text on the Web matching local context (even with truncated queries). The lowest *% w/ sels*, found for *pro*, was expected considering only nouns which replace the original words are used (pronouns acquired were thrown out since they are not compatible with the relatedness measures). There was quite a variation in the *sels/inst* depending on the type, and all of these numbers are well below the upper-bound of 200 selectors acquired before the algorithm stops searching. It turned out that only 15.9% of the instances hit this mark. This means that most instances stopped acquiring selectors because they hit the minimum query length (5 words). In

Table 4.1: Various statistics on the acquired selectors for the SemEval07 Task 7 broken down by part of speech.

	noun	***verb***	***adj.***	***adverb***	***pro***
insts	1108	591	362	208	370
% w/ sels	54.5	65.8	61.0	57.2	27.0
sels/inst	36.5	51.2	29.5	17.7	15.9
unique/inst	11.6	13.1	8.4	4.1	5.6
insts/sent	4.5	2.4	1.5	0.8	1.5

insts instances which the algorithm attempts to acquire selectors
% w/ sels percentage of instances for which selectors were acquired
sels/inst average number of selectors for an instance (over all *insts*)
unique/inst average number of unique selectors for an instance (over all *insts*)
insts/sent average instances in a sentence

fact, the average web query to acquire at least one selector had a length of 6.7 words, and the bulk of selectors came from shorter queries (with less context from shorter queries, the selectors returned are not as strong). The combination of quantity and quality issues presented above is referred to as the *quality selector sparsity* problem.

Although quality and quantity were not ideal, when one considers data from the sentence level, things are more optimistic. The average sentence had 10.7 instances (of any part of speech listed), so when certain selector types were missing, others were present. As explained previously, the *target selector* and *context selector* distinction is made after the acquisition of selectors. Thus, each instance is used as both (exception: *pro* instances were never used as *target selectors* since they were not disambiguated) . Employing this fact, more information can be discovered. For example, the average noun was disambiguated with 36.5 *target selectors*, 122.9 *verb context selectors* (51.2 *sels/inst* * 2.4 *insts/sent*), 44.3 *adjective context selectors*, 14.2 *adverb context selectors*, and 23.9 *pro context selectors*. Still, with the bulk of those selectors coming from short queries, the reliability of the selectors was not strong.

Note two differences between selectors as *CSK* and the database of knowledge presented in section 3.1, *CSKB*. The first difference is that the relationship between a selector and its local context is not explicitly recorded, while the *search phrases* for the *CSKB* were constructed to find specific relationships. Second, *Web selectors* are dynamic. As language evolves and changes on the Web selectors will change as well. This is a benefit, but also a drawback in that it takes time to acquire selectors during runtime. With the *CSKB*, the acquisition is done ahead of time, and applications simply lookup the knowledge within the database rather than perform a Web search.

4.3 Method to Utilize Web Selectors in Disambiguation

This section examines a semantic interpretation method using selectors acquired from the Web. As explained first in section 2.2.3, selectors describe words which may take the place of another given word within its local context. These words are used in an algorithm to perform noun, verb, adjective, and adverb word sense disambiguation. The overall process of acquiring selectors and applying them to *WSD* is shown in Figure 4.4. Results over SemEval2007 (Navigli et al., 2007) find noun sense disambiguation accuracy above a most frequent sense baseline. Verb, adjective, and adverb disambiguation accuracies were slightly below the most frequent sense baseline, while well above a random baseline and the average system participating in SemEval. Further experiments found that, for noun and verb sense disambiguation tasks, each type of context selector can assist target selectors in disambiguation. Finally, these experiments also help to draw insights about the future direction of similar research.

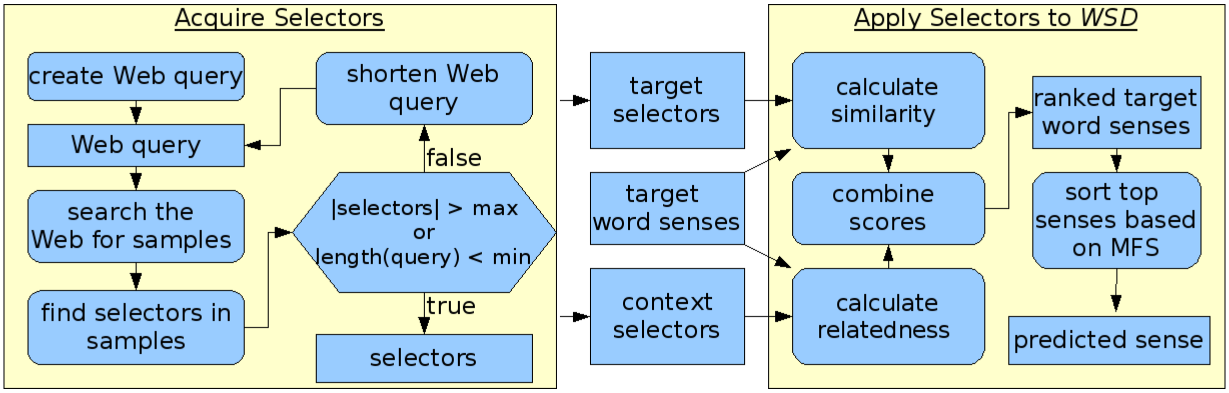


Figure 4.4: The overall process undertaken to disambiguate a word using Web selectors.

Section 4.1 described the difference between *target* and *context* selectors. *Similarity* is used to measure the relationship between a target word and its *target selectors*, while *relatedness* measures the relationship between a target word and *context selectors* from other parts of the sentence. Thus, the use of selectors in disambiguating words relies on a couple assumptions:

1. Concepts which appear in matching syntactic constructions are *similar*.

2. Concepts which appear in the context of a given target word are *related* to the correct sense of the target word.

This idea of distinguishing similarity and relatedness has an extensive history Rada et al. (1989); Resnik (1999); Patwardhan et al. (2003); Budanitsky & Hirst (2006), but most algorithms only find a use for one or the other. Essential the implementation of this assumption for context selectors follows this path: the target word is *related* to the context word, which is *similar* to its context selector. Through transitive closure and since relatedness encompasses similarity, the target word sense is then *related* to the *context selector*.

The correct sense of a target word is chosen based on a combination of the strength given from similarity and relatedness measures over WordNet and the probability of a selector occurring within the local context. A *similarity* measure was used with *target selectors* while a *relatedness* measure was used with *context selectors*. The process of combining values can be seen in Figure 4.5.

After acquiring selectors as described in section 2.2.3, the occurrences of selectors can be converted to a probability of a selector, w_s appearing in a web query, q, represented as function 4.1.

$$p_{sel}(w_s, q) \tag{4.1}$$

Function 4.2, based on Resnik's *word similarity* (Resnik, 1999), is used to find the max similarity or relatedness between a concept and a word (specifically between a sense of the target word, c_t and a selector, w_s):

$$maxsr(c_t, w_s) = \max_{c_s \in w_s}[meas(c_t, c_s)] \tag{4.2}$$

where c_s is a sense of the selector and $meas$ is a similarity or relatedness measure.

The senses of the target word are compared with each selector. For a given sense of the target word, c_t, the similarity or relatedness from a selector and query is computed as defined in function 4.3.

$$SR(c_t, w_s, q) = \frac{p_{sel}(w_s, q) * maxsr(c_t, w_s)}{senses(w_s)} \tag{4.3}$$

where $senses(w_s)$ is the number of senses of the selector.

As the queries get shorter, the accuracy of the selectors becomes weaker. For example, one of the sentences from the test corpus is truncated to the following web query "PHP and Python *

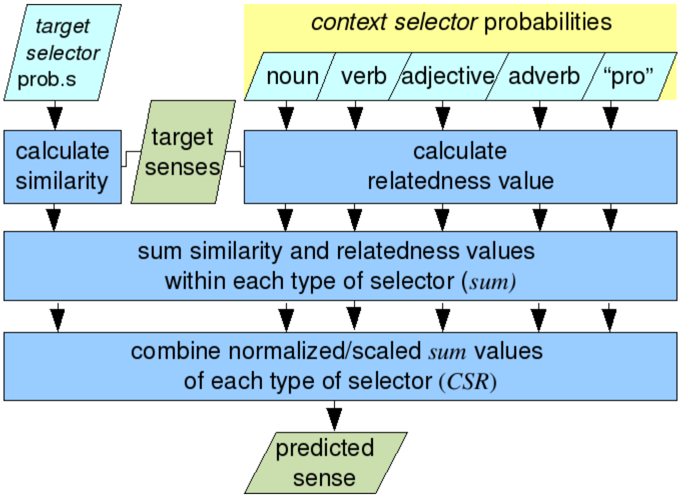

target selector prob.s
context selector probabilities
noun
verb
adjective
adverb
“pro”
calculate similarity
target senses
calculate relatedness value
sum similarity and relatedness values within each type of selector (sum)
combine normalized/scaled sum values of each type of selector (CSR)
predicted sense

Figure 4.5: General flow in applying selectors to word sense disambiguation. Note that the target selectors may be any part of speech.

the”, which produces selector occurrences for ‘protect’: (‘be’: 6, ‘code’: 7, ‘help’: 1, ‘import’: 2, ‘store’: 2, ‘use’: 2). The most frequent of these is ‘code’, which as a verb is not similar to the correct sense of ‘protect’. In turn, the SR value from selectors is scaled by a ratio of the web query length, wql, to the original sentence length, sl. This scaling is applied when the *SR* values for one target word sense are summed in function 4.4:

$$sum(c_t, T) = \sum_{q \in qs(T)} \sum_{w_s \in sels(q)} SR(c_t, w_s, q) * \frac{wql}{sl} \tag{4.4}$$

where $qs(T)$ represents the set of queries for a selector type, T, and w_s ranges over all selectors found with q, denoted $sels(q)$.

The general approach of disambiguation is to find the sense of a target word which is most similar to all target selectors and most related to all context selectors. This follows the assumptions initially given about selectors. Thus, similarity and relatedness values from different selector types, represented as $Types$, must be combined. By aggregating the normalized $sums$ from all types of selectors, one achieves a combined similarity/relatedness for a given target word senses as defined in function 4.5:

$$CSR(c_t) = \sum_{T \in Types} scale(T) * \frac{sum(c_t, T)}{\max\limits_{c_i \in w_t}[sum(c_i, T)]} \tag{4.5}$$

where w_t represents the set of all senses belonging to the target word, $Types$ spans over the set: (*target, noun context, verb context, adjective context, adverb context, pro context*), and $scale(T)$ is a coefficient used to weight each type of selector. This term is important in this work, because an experiment explores the impact of various selector types.

The top sense is then chosen by looking at the CSR of all senses. For some situations, specifically when other senses have a score within 5% of the top CSR, the difference between concepts is very small. In these cases, the concept with the lowest sense number in WordNet is chosen from among the top scoring senses.

4.4 Evaluation

Four experiments were performed in order to validate the Web selectors method of *WSD*. The first explores various similarity and relatedness measures for noun *WSD*. The second experiment is focused on finding results for disambiguating all parts of speech. The third experiment tests the validity of the Web selectors algorithm on a domain *WSD* task. Finally, experiments are done in order to study the impact of each type of selector in the accuracy of the disambiguation results.

Three of the four experiments (1, 2, and 4) utilized the corpus from the SemEval 2007 Task 7: coarse-grained English all-words. The sense inventory was created by mapping senses in WordNet 2.1 to the Oxford Dictionary of English (Navigli et al., 2007). The corpus was composed of five documents with differing domains resulting in 2269 annotated word instances. The Web selector algorithm runs on fine-grained WordNet senses, but evaluation is done by checking if the predicted fine-grained sense maps to the correct coarse-grained sense. Many issues associated with fine-grained annotation, such as those brought up in (Ide & Wilks, 2006) are avoided through the use of this corpus. The third experiment utilized the SemEval 2010 Task 17: All-Words Word Sense Disambiguation on a Specific Domain, which used sense inventories from OntoNotes (Agirre et al., 2010). In this case, Web selectors was a participating system of SemEval 2010 (Schwartz & Gomez, 2010). Results for all experiments are presented as precision (P), recall (R), and F1 value (defined based on precision and recall as $F1 = 2 * \frac{P*R}{P+R}$).

The similarity and relatedness measures spanned the following set, defined in Section 2.3: $S_{WuPalmer}$ (Wu & Palmer, 1994), $S_{LeacockChodorow}$ (Leacock et al., 1998), $R_{HirstOnge}$ (Hirst & St Onge, 1998), S_{Resnik} (Resnik, 1999), S_{Lin} (Lin, 1997), $S_{JiangConrath}$ (Jiang & Conrath, 1997), $R_{BanerjeePederson}$ (Banerjee & Pedersen, 2002), and $R_{PatwardhanPederson}$ (Patwardhan & Pedersen, 2006). Since the relatedness measures used for *context selectors* require comparison across multiple parts of speech, only $R_{BanerjeePederson}$ and $R_{PatwardhanPederson}$ were used for relatedness as these were the only two measures which supported this. All of the measures were used at some point for *target selectors*. Additionally, $S_{SchwartzGomez}$ (Schwartz & Gomez, 2008) is used which is presented in Chapter 5 as a contribution of this dissertation. The WordNet::Similarity package provides a flexible implementation of the other measures (Pedersen et al., 2004). WordNet::Similarity was configured for for WordNet version 2.1, the same version used to annotate the chosen experimental corpus.

4.4.1 Experiment 1: Noun Sense Disambiguation

Out of 2269 noun, verb, adjective, or adverb instances in the SemEval 2007 Task 7, this experiment was concerned with disambiguating the 1108 noun instances from the 245 sentences in the corpus . These noun instances represent 593 different words. Information about the acquired selectors was given in section 4.2. For this section, the bottom of Table 4.2 shows the random baseline as well as a baseline using the most frequent sense (MFS) heuristic. As previously mentioned, many supervised systems only perform marginally better than the MFS. For the SemEval task, only 6 of 15 participating systems performed better than this baseline on the nouns (Navigli et al., 2007), all of which used MFS as a back off strategy and an external sense tagged data set.

The tests in this section use a $scale(T)$ value of 1 for *target selectors*, a value of 0.5 for noun and verb *context selectors*, and a value of 0.1 for adjective and pro *context selectors* (see function 4.5). This weights the scores that come from target selectors equal to that of noun and verb context selectors, while the adjective and pro selectors only play a small part. These choices were based on early tests with a small portion of the corpus, and intended to get the best results for noun *WSD*. Additionally, adverb *context selectors* were not used. Note that experiment 4 presents an extensive

Table 4.2: Performance of using Web Selectors for noun sense disambiguation with various similarity and relatedness measures.

	$\mathbf{R}_{BanerjeePederson}$	$\mathbf{R}_{PatwardhanPederson}$
$\mathbf{S}_{SchwartzGomez}$	**80.2**	78.6
$\mathbf{S}_{WuPalmer}$	78.7	78.6
$\mathbf{S}_{Resnik}$	78.6	79.3
$\mathbf{S}_{Lin}$	78.5	79.2
$\mathbf{S}_{JiangConrath}$	78.0	78.1
$\mathbf{R}_{BanerjeePederson}$	78.4	80.0
$\mathbf{R}_{PatwardhanPederson}$	78.6	78.9
$\mathbf{BL}_{MFS}$	77.4	(baseline)
$\mathbf{BL}_{rand}$	59.1	(baseline)

Results are F1 values (precision = recall). Similarity measures for *target selectors* are row headings while relatedness measures for *context selectors* are column headings. Baselines: $\mathbf{BL}_{MFS}$ = most frequent sense, $\mathbf{BL}_{rand}$ = random choice of sense.

set of tests with various $scale(T)$ values for all types of *context selectors* (including adverb) over the disambiguation of all parts of speech.

Table 4.2 shows the results when using various similarity measures for the *target selectors*. For the *context selectors*, gloss-based measures were selected due to the need for handling multiple parts of speech. The web selectors method performs better than the MFS baseline. The path-based measure, $S_{SchwartzGomez}$, is defined in Chapter 5. This similarity measure along with the gloss based $R_{BanerjeePederson}$ relatedness measure gave the best results. Note that the path-based and information content measures, in general, performed equally. The table also shows the gloss-based $R_{BanerjeePederson}$ and $R_{PatwardhanPederson}$ relatedness measures used in place of similarity measures. The idea was that one measure could be used for both target selectors and context selectors. The bottom of Table 4.2 shows the measures performed nearly equally. The experimental runtime of the path-based and information content measures was roughly one-fourth that of the gloss-based measures, so they are usually preferred.

Table 4.3: Results for the Web Selectors method with restrictions on a minimum number of target selectors and context selectors.

tMin	cMin	A	P	R	F1
0	0	1108	80.2	80.2	80.2
4	0	658	84.4	50.1	62.9
16	0	561	85.2	43.1	57.2
0	10	982	81.1	71.9	76.2
0	40	908	81.3	66.6	73.3
4	10	603	85.4	46.4	60.1
8	20	554	85.3	42.6	56.9
12	30	516	86.4	40.2	54.9
16	40	497	86.5	38.8	53.5

A: Number attempted, **P**: Precision, **R**: Recall, **F1** values; **tMin**: minimum number of target selectors, **cMin**: context selectors.

Table 4.3 presents results from tests where annotations were only to instances with over a minimum number of target selectors (**tMin**) and context selectors (**cMin**). Steps of four were used for **tMin** and steps of ten were used for **cMin**, reflecting a ratio of roughly 2 target selectors for every 5 context selectors. It was more common for an instance to not have any target selectors than to not have context selectors, so results are presented with only a **tMin** or **cMin** as well. The main goal of these tests was simply to determine if the algorithm performed better on instances from which more selectors were acquired. This was the case as the precision improved at the expense of recall from avoiding the noun instances that did not have many selectors.

Table 4.4 shows the results when modifying the method in a few ways. All these results use the path-based $S_{SchwartzGomez}$ similarity measure and the gloss-based $R_{BanerjeePederson}$ relatedness measure. The results of Table 4.2 included the first sense heuristic used as a back-off strategy for close calls, when multiple senses have a score within 0.05 of each other. Therefore, an results without this heuristic are presented as **noMFS**, and indicate the method still performs strongly.

Table 4.4: Results of tests noun *WSD* tests with modifications to the algorithm.

WS	**noMFS**	**1SPD**
80.2	79.6	79.8

All tests used $S_{SchwartzGomez}$ and $R_{BanerjeePederson}$. **WS**: Web Selectors baseline, **noMFS**: WS with no use of most frequent sense, **1SPD**: WS with use of one sense per discourse.

Table 4.5: Comparison of noun F1 values with various participants on the SemEval2007 coarse-grained all-words task.

WS	**MED**	**UPV-WSD**	**NUS-PT**	**SSI**
80.2	71.1	79.33	82.31	84.12

WS: Web Selectors, **MED**: median of all participating systems, **UPV-WSD**: (Buscaldi & Rosso, 2007), **NUS-PT**: (Chan et al., 2007), **SSI**: (Navigli & Velardi, 2005)

Another test implemented one sense per discourse (Gale et al., 1992), reported as **1SPD**. The experimental corpus had five documents, and for each document the most commonly predicted sense was calculated and used for all occurrences of the word within the document. This strategy did not seem to improve the results.

A results comparison of the Web selectors method (**WS**) to other systems participating in the SemEval task is given in in Table 4.5. These results include the median of all participating systems (**MED**), the top system not using training data (Buscaldi & Rosso, 2007, **UPV-WSD**), and the top system using training data (Chan et al., 2007, **NUS-PT**). The best performance reported on the nouns for the SemEval coarse-grained task, was actually from a system that was used to help annotate the data in the first place by the authors of the task (Navigli & Velardi, 2005, **SSI**). All systems performing better than the $\mathbf{BL}_{MFS}$ used the MFS heuristic as a backoff strategy when unable to output a sense (Navigli et al., 2007). Also, the systems performing better than **WS** (including **SSI**) used more sources of sense annotated data.

Table 4.6: Results as F1 Values over all parts of speech for *WSD*.

BL$_{Rand}$	**MED**	**WS**	**BL**$_{MFS}$
53.43	70.21	**76.02**	78.89

WS: the Web Selectors system , **BL**$_{Rand}$: random baseline, **BL**$_{MFS}$: MFS baseline, **MED**: median system performance at SemEval07 task 7 (Navigli et al., 2007)

4.4.2 Experiment 2: Disambiguation of all Parts of Speech

This experiment applies the Web selectors algorithm in a straight-forward manner over all parts of speech in order to validate the algorithm's scalability to large sets of words and all parts of speech. Tests were done with $R_{BanerjeePederson}$ as the *relatedness* measure for *context selectors*. An important characteristic of this measure is that it can handle multiple parts of speech. For *target selectors* we sought to use measures over the WordNet ontology in order to most closely measure *similarity*. An information-content (IC) measure, S_{Resnik}, was used for target selectors of nouns and verbs. However, because IC and path-based measures do not work with all parts of speech, we used the adapted Lesk algorithm as an approximation of *similarity* for adjectives and adverbs. Note that finding the best relatedness or similarity measure was outside the scope of this experiment.

The results are compared with baselines and other disambiguation algorithms. Unless stated otherwise, all results are presented as F1 values. For SemEval2007, all systems performed better than the random baseline of 53.43%, but only 4 of 13 systems achieved an F1 score higher than the MFS baseline of 78.89% over all parts of speech (Navigli et al., 2007).

Table 4.6 lists the results of applying the Web selector algorithm described in this work in a straight-forward manner, such that all $scale(T)$ are set to 1. The Web selectors system performs better than the median system in the SemEval07 task, but it is a little below the MFS baseline.

Table 4.7: Results as F1 Values of top performing systems for the SemEval07 Task07.

UPV-WSD	**NUS-PT**	**SSI**
78.63	82.50	83.21

UPV: (Buscaldi & Rosso, 2007), **NUS-PT**: (Chan et al., 2007), **SSI**: a task organizer's system (Navigli & Velardi, 2005)

Table 4.8: Results as F1 values of the Web Selectors system by parts of speech.

	N	**V**	**A**	**R**
MED	70.76	62.10	71.55	74.04
WS	**78.52**	**68.36**	**81.21**	**75.48**
BL$_{MFS}$	77.44	75.30	84.25	87.50
insts	1108	591	362	208

N: noun, **V**: verb, **A**: adjective, **R**: adverb). *insts*: disambiguation instances of each part of speech. For other keys see Table 4.6.

A comparison with top systems is seen in Table 4.7. Overall results were just below that of the top system not utilizing training data (Buscaldi & Rosso, 2007, **UPV-WSD**), and a little over 6 percentage points below the top supervised system (Chan et al., 2007, **NUS-PT**).

The results are broken down by part of speech in Table 4.8. Adjective disambiguation was the furthest above the median point of reference, and noun disambiguation results were above the MFS baseline. On the other hand, the adverb disambiguation results appear weakest compared to the baselines. Note that the previous experiment reported a noun sense disambiguation F1 value of 80.20% on the same corpus (Schwartz & Gomez, 2008). Current results differ because the previous experiment used different $scale(T)$ values as well as a custom noun similarity measure.

4.4.3 Experiment 3: Domain Word Sense Disambiguation

This section studies the application of the Web Selectors word sense disambiguation system on a specific domain. The system was primarily applied without any domain tuning, but the incorporation of domain predominant sense information was explored. In the previous experiments, the Web Selectors system was applied to text of a general domain. However, the system was not directly tuned for the general domain. The system may perform just as strong for domain WSD since the selectors, which are the core of disambiguation, can come from any domain present on the Web. Therefore, this experiment explores the application of the Web Selectors WSD algorithm to an all-words task on a specific domain, the SemEval 2010: Task 17 (Agirre et al., 2010).

This study utilized the implementation of the Web Selectors system from the previous experiment that was also presented in (Schwartz & Gomez, 2009b). The incorporation of a part of speech tagger was a necessary addition to the existing system. Previous evaluations of Web Selectors relied on the testing corpus to provide part of speech (POS) tags for content words. In the case of SemEval-2010 Task 17, words were only marked as targets, but their POS was not included. The system used the POS tags from the Stanford Parser (Klein & Manning, 2003). The Stanford Parser was chosen since the dependency relationship output was also useful for our domain adaptation. A modification was made to the POS tags given the knowledge that the testing corpus only included nouns and verbs as targets. Any target that was not initially tagged as a noun or verb was reassigned as a noun, if the word existed as a noun in WordNet (Miller et al., 1993), or as a verb if not.

Overall, the Web Selectors system is not explicitly tuned to the general domain. Selectors themselves can be from any domain; They are completely dependent upon the local context which is found anywhere on the Web (assumed to contain the same domain being tested on). However, sense tagged data may be used indirectly within the system. First, the similarity and relatedness measures used in the system may rely on SemCor data (Miller et al., 1994). Also, the system breaks ties by choosing the most frequent sense according to WordNet frequency data (based on SemCor). These two aspects of the system can be seen as tuned to the general domain, and thus, they are likely aspects of the system for adaptation to a specific domain.

This experiment focused on domain-adapting the tie breaker aspect of the Web Selectors system. The system defines a tie occurring when multiple sense choices are scored within 5% of the top sense choice. In order to break the tie, the system normally chooses the most frequent sense among the tied senses. However, it would be ideal to break the tie by choosing the most prevalent sense over the testing domain. Because sense tagged domain data is not typically available, Koeling et al. (2005) presented the idea of estimating the most frequent sense of a domain by calculating sense prevalence scores from unannotated domain text.

Several steps are taken to calculate the prevalence scores. First, a dependency database is created, listing the frequencies that each dependency relationship appears. In this case, the Stanford Parser (Klein & Manning, 2003) was used on the background data provided by the task organizers. From the dependency database, a thesaurus is created based on the method of (Lin, 1998a). In this approach, the following relationships from the dependency database were considered (typed dependency names listed in parenthesis):

subject (*agent, csubj, subjpass, nsubj, nsubjpass, xsubj*)

direct object (*dobj*)

indirect object (*iobj*)

adjective modifier (*amod*)

noun modifier (*nn*)

prepositional modifier (any preposition, excluding *prep_of* and *prep_for*)

Finally, a prevalence score is calculated for each sense of a noun or verb by finding the similarity between it and the top 50 most similar words according to the automatically created thesaurus. Based on Koeling et al., the similarity measure of $S_{JiangConrath}$ was used.

The main results of the experiment are given in Table 4.9. The first set of results (**WS**) was a standard run of the system without any domain adaptation, while the second set (**WS**$_{dom}$) was from a run including the domain prevalence scores in order to break ties. The results show that the domain adaptation technique did not lead to improved results. Overall, **WS** results came in ranked thirteenth among twenty-nine participating system results.

This study found that using the prevalence scores alone to pick a sense (i.e. the 'predominant sense') resulted in an F score of 0.514 (***PS*** in Table 4.9). Koeling et al. (2005) found the predominant sense to perform significantly better than the first sense baseline (*1sense*: equivalent to most frequent sense for the English WordNet) on specific domains (32% error reduction on a finance domain, and 62% error reduction on a sports domain). Interestingly, there was no significant error reduction over the *1sense* for this task, implying either that the domain was more difficult to adapt to or that our implementation of the predominant sense algorithm was not as strong as that used

Table 4.9: (**P**)recision, (**R**)ecall, and (**F**)-score of various runs of the system on the Task 17 data.

	P	**R**	**F**	$\mathbf{P}_n$	$\mathbf{P}_v$
rand	0.23	0.23	0.23		
1sense	0.505	0.505	0.505		
WS	0.447	0.441	0.444	.446	.449
$\mathbf{WS}_{dom}$	0.440	0.434	0.437	.441	.438
PS	0.514	0.514	0.514	.53	.44

$\mathbf{P}_n$ and $\mathbf{P}_v$ correspond to precision results broken down by nouns and verbs.

by Koeling et al.. In any case, this lack of significant error reduction over the *1sense* may explain why our $\mathbf{WS}_{dom}$ results were not stronger than the **WS** results. In $\mathbf{WS}_{dom}$, prevalence scores were used instead of *1sense* to break ties.

A few figures were computed to gain more insights on the system's handling of domain data. Noun precision was 0.446 while verb precision was 0.449. It was unexpected for verb disambiguation results to be as strong as nouns because the previous experiment using Web Selectors found noun sense disambiguation clearly stronger than verb sense disambiguation on a coarse-grained corpus. Ideally, *WS* results for noun disambiguation would have been stronger than the the *1sense* and *PS* results. In order to determine the effect of the POS tagger (parser in this case) on the error, we determined 1.6% of the error was due to the wrong POS tag at (0.9% of all instances). Lastly, Table 4.10 shows the precision scores for each of the three documents from which the English testing corpus was created. Without understanding the differences between the testing documents it is difficult to explain why the precision varies, but the figures may be useful for comparisons by others.

Several aspects of the test data were unexpected for the system. Some proper nouns were considered as target words. Our system was not originally intended to annotate proper nouns, but

Table 4.10: Precision scores based on the three documents of the English testing corpora (‘en1’, ‘en2’, and ‘en3’).

	$\mathbf{P}_{en1}$	$\mathbf{P}_{en2}$	$\mathbf{P}_{en3}$
WS	0.377	0.420	0.558
$\mathbf{WS}_{dom}$	0.384	0.415	0.531

it was adjusted to treat proper nouns simply as nouns. To be sure this treatment was appropriate, a test determined resulted in a precision of 0.437 and recall of 0.392 when proper nouns were excluded. One would expect the precision to increase at the expense of recall if the proper nouns were more problematic for the system than other instances. Unfortunately, another unexpected aspect of the data was not handled correctly by the system. The system only considered senses from one form of the target word according to WordNet, while the key included multiple forms of a word. For example, the key indicated *low_tide-1* was the answer to an instance where our system had only considered senses of ‘tide’. In a similar example when the system ran across the proper noun ’Banks’, it only considered senses of ’Banks’ (which has just one sense) instead of ’bank’ which has 10 senses. It was determined that for 10.2% of the instances that were incorrect in the **WS** results, the correct sense was not even considered as a possible prediction due to using an inventory from only one form of the word. Since this issue mostly applied to nouns, it may explain the observation that the noun disambiguation performance was not better than the verb disambiguation performance as was expected.

4.4.4 Experiment 4: Impact Analysis

This section explores the influence of each context selector on the disambiguation algorithm. This is done by changing the value of $scale(T)$ in CSR (function 4.5). Examining Table 4.11 reveals precision results when disambiguating instances with *target selectors*, based only on the target word's similarity with target selectors. This serves as a bearing for interpreting results of context selector variation.

The tests are concerned with determining how well each type of *context selector* complements the *target selectors*. Accordingly, $scale(target)$ was set to 1, and $scale(T)$ for all other context types were set to 0. In order to limit external influences, words with only one sense in WordNet or instances where the CSR was zero (indicating no selectors) were not disambiguated. Additionally, examples were only tested if they had at least one target selector and at least one selector of the specific type being examined. This restriction ensures avoidance of some of the *quality selector sparsity* problem described in section 4.2. Nevertheless, results are expected to be a little lower than experiments 1 and 2 as other types of selectors are ignored and monosemous words according

Table 4.11: Precision when disambiguating with *target selectors* only.

wsd	**prec.** %	*insts.*
N	64.08	348
V	52.86	227
A	77.36	106
R	58.39	56

All instances contain target selectors and multiple senses in WordNet. *insts.*: number of instances disambiguated.

Table 4.12: Instance occurrences used for disambiguation when experimenting with all types of context selectors.

wsd	***noun***	***verb***	***adj.***	***adverb***	***pro***
N	272	186	120	84	108
V	211	167	110	80	103
A	97	78	50	40	34
R	47	44	30	17	26

The types of context selectors are listed as columns. The rows represent the four parts of speech disambiguated.

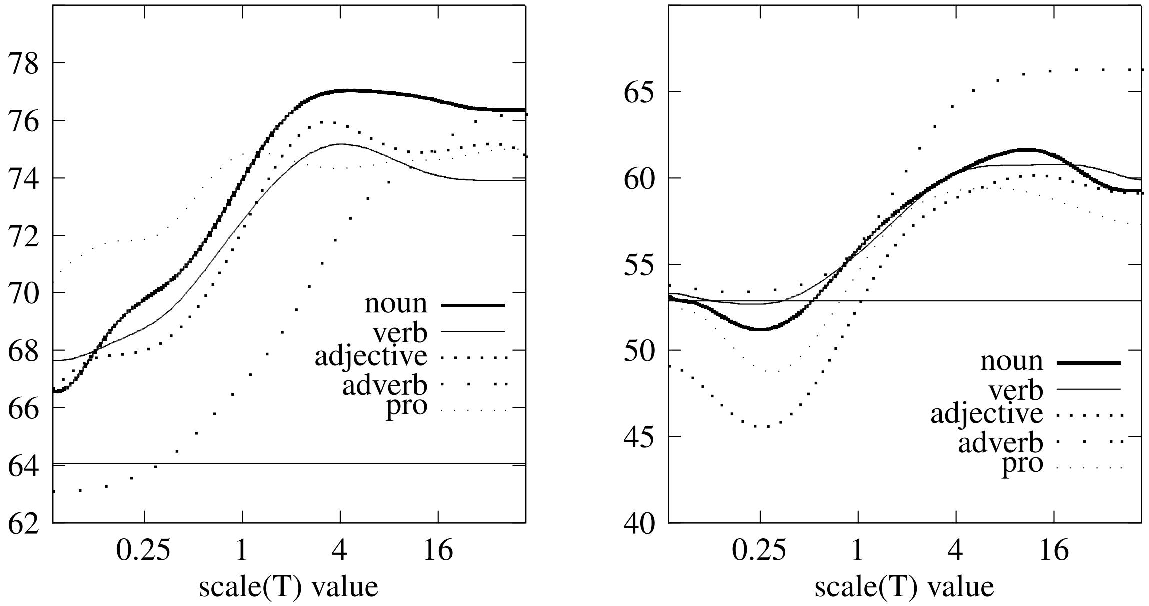
to WordNet are not included. Table 4.12 lists the instance occurrences for each of the four parts of speech that were disambiguated, based on these restrictions.

Figures 4.6 and 4.7 show graphs of the precision score while increasing the influence of each context selector type. Each graph corresponds to the disambiguation of a different part of speech, and each line in a graph represents one of the five types of context selectors:

1. *noun context*
2. *verb context*
3. *adjective context*
4. *adverb context*
5. *pro context*

The lines are formed with a Bezier curve algorithm[2] on the precision data. The horizontal line represents the precision of only using the target selectors to disambiguate instances with target selectors (see Table 4.11). Precision either decreases or remains the same if any graph line was extended past the right-most boundary.

[2] http://www.gnuplot.info/docs/node124.html



(a) **noun** sense disambiguation

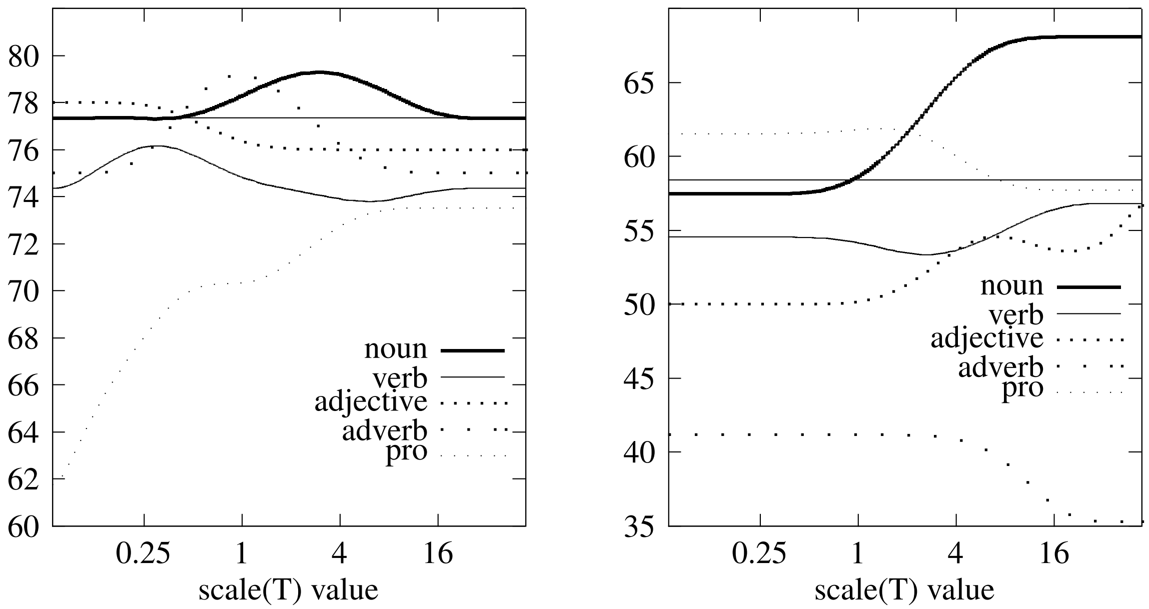
(b) **verb** sense disambiguation

Figure 4.6: The **noun** (left) and **verb** (right) *WSD* precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is always 1.

When examining the figures, one should note when the precision increases as the $scale$ value increases. This indicates that increases in influence of the particular type of context selector improved the results. If the precision decreases as the $scale$ becomes greater, this indicates that the *context selector*'s influence dominated the *target selector*'s. The x-axis increases exponentially, since the graphs present a ratio of $scale(T)$ to $scale(target)$. At $scale(T) = 1$ the *context selector* has the same influence as the *target selector*.

4.5 Discussion

Experiment 1 results showed strength in the use of selectors from the Web for noun *WSD*. The system was out-performed only by systems using training data or substantially more annotated



(a) **adjective** sense disambiguation

(b) **adverb** sense disambiguation

Figure 4.7: The **adjective** (left) and **adverb** (right) *WSD* precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is always 1.

data. Additionally, better precision was achieved when requiring a minimum number of selectors, giving promise to improved results with more work in acquiring selectors. Finally, different types of similarity and relatedness measures are appropriate for different roles in the disambiguation algorithm. A path-based measure was best with *target selectors* while a slower gloss-based method was appropriate for *context selectors* in order to handle multiple POS. For many tasks, information content based measures perform better than path-based measures (Budanitsky & Hirst, 2006; Patwardhan et al., 2003). However, this experiment found a path-based measure to be just as strong if not stronger for the selectors approach.

While the first experiment helped to validate the Web selectors approach as a top performing *WSD* algorithm, the experiment was limited to noun *WSD* and adverb *context selectors* were not used. Experiment 2 found the use of Web selectors to be a worthwhile approach to the disambiguation of other parts of speech in addition to nouns. However, results for verb, adjective, and adverb disambiguation were slightly below the most frequent sense baseline, a point which noun sense disambiguation overcomes. One reason suspected for this is that work in similarity and relatedness has a longer history over nouns than over other parts of speech Budanitsky & Hirst (2006). Additionally, the hypernym (is-a) relationship of the noun ontology in WordNet captures the notion of *similarity* more clearly than the primary relationships of other parts of speech in WordNet. Accordingly, future work should look into specific measures of *similarity* for each part of speech. In all the tests of this experiment, all $scale(T)$ values were the same and consistent, weighting all selector types equally.

Results from experiment 3 indicated that the system performs relatively the same with domain predominant sense information as without, scoring well above a random baseline, but still 5 percentage points below results of using the first sense. A primary goal was to apply the pre-existing system with minimal changes. To do this an automatic part of speech tagger was incorporated, which was found to only have a small impact on the error (incorrectly tagged 0.9% of all target instances). Overall, the results showed the system to perform below the *1sense* baseline for both nouns and verbs. This is a lower relative performance than past studies which found the disambiguation performance above the *1sense* for nouns. One reason for the lower noun performance is that, for 10.2 % of our errors, the system did not consider the correct sense choice as a possibility. Future versions of the system will need to expand the sense inventory to include other forms of a word (example: ‘low_tide’ when disambiguating ‘tide’). Toward domain adaptation, an experiment was run in which one aspect of our system was tuned to the domain by using domain prevalence scores (or ‘predominant senses’). No improvement was found from using this adaptation technique, but it was also discovered that results entirely based on predictions of the domain predominant senses were only minimally superior to *1sense* (F-score of 0.514 versus 0.505 for *1sense*). There is certainly room for future studies to examine better implementations of the predominant sense algorithm, as well as explore other complimentary techniques for domain adaptation: customizing similarity measures for the domain, or restricting areas of WordNet as sense choices based on the domain.

Experiment 4 used various $scale(T)$ values and found that all types of *context selectors* improve the results for noun and verb sense disambiguation. Thus, inclusion of all context selectors was worthwhile for nouns and verbs. It is difficult to draw a similar conclusion from the adverb and adjective disambiguation graphs (Figure 4.7), although the *noun context selectors* are helpful for both and the *pro context selectors* are helpful for the adjective task. Most context selector types achieve highest precision above a $scale(T)$ value of 1, indicating that the *context selector* should have more influence than the *target selectors*. This is probably due to the existence of more selectors from context than those from the target word. The results of adverb disambiguation should be taken lightly, because there were not many disambiguation instances that fit the restrictions (see Table 4.12).

Overall, the evaluation found that Web selectors are a valid approach to *WSD*. The selectors were used in a method utilizing similarity measures over WordNet, and achieved results comparable with top minimally-supervised approaches to the problem. Finally, these experiments also help to draw insights about the future direction of research.

5 CONCEPT SIMILARITY IN THE CONTEXT OF KNOWLEDGE ACQUISITION

As exemplified by the Web Selector algorithm, similarity measures are useful for analyzing and applying acquired knowledge. In particular, measures that function on concepts rather than words are beneficial under the assumption that *CSK* is about concepts rather than simply words. This Chapter describes three related but distinct contributions of this dissertation. First, a novel semantic similarity measure is presented that was created during development of the Web selectors algorithm. Next, two sections describe evaluations of WordNet-based semantic similarity and relatedness measures in tasks focused on concept similarity. Assuming similarity as distinct from relatedness, the goal is to fill a gap within the current body of work in the evaluation of similarity and relatedness measures. Past studies have either focused entirely on relatedness or only evaluated judgments over words rather than concepts. In the first evaluation, concept similarity measures are evaluated over human judgments by using existing sets of word similarity pairs that were annotated with word senses. Lastly, an application-oriented study is presented by integrating similarity and relatedness measures into an algorithm which relies on concept similarity.

Two distinctions are important within this chapter: that between *words* and *concepts*, and that between between *relatedness* and *similarity*. Although many measures are designed for comparison of *concepts* (word senses), past comparisons of similarity and relatedness measures with human judgments have looked into similarity between *words* themselves, leaving some ambiguity. For example, while one would likely agree that ‘bat’ as in “a club used for hitting a ball” is similar to ‘stick’, one would be hard-pressed to agree that ‘bat’ as in “nocturnal mouselike mammal with forelimbs modified to form membranous wings” is also similar to ‘stick’ (definitions from WordNet Miller et al. (1993)). On the other hand, while application-oriented studies have applied measures to concepts the field has yet to see an evaluation utilizing an application calling for *similarity* judgments. This paper views *similarity* as a specific type of relatedness characterized by the relationships: synonymy, antonymy, and hyponymy. As an example, one would say a ‘wooden stick’ is *similar* and *related* to a ‘baseball bat’, while a ‘baseball player’ is only *related* to a ‘baseball bat’. Although this similarity distinction has been noted previously Resnik (1999); Patwardhan et al. (2003); Agirre & Soroa (2009), this section presents the first evaluation of measures for tasks of *concept similarity*.

5.1 Similarity based on Normalized Depth

As noted from Section 2.3, path based similarity measures have the benefits of not requiring any data beyond WordNet relationships, as well as a fairly quick run-time. As part of the work developing the Web Selectors algorithm (Schwartz & Gomez, 2008), a novel path-based similarity measure was created.

One may recall a problem with early path-based measures is the assumption that the edges between concepts are all uniform (Resnik, 1999). Although other path-based measures, such as

Leacock et al. (1998) took taxonomic depth into account, previous measures did not consider the various sub-graphs of the ontology suffering from the *uniformity problem*. For example, both *fractal* and *hydrangea* would be scaled by the same depth. The *normalized depth* measure works by scaling the depth of a concept by the depth of the specific portion of the ontology it belonged. It requires computing *average leaf depth* (ald), a value indicating the average depth of all descendants (hyponyms) that do not have hyponyms themselves:

$$ald(c) = \frac{\sum_{L \in lnodes(c)} depth(l)}{|lnodes(c)|} \tag{5.1}$$

where $lnodes$ returns a list of leaf nodes attached to c, those nodes without hyponyms that are themselves a type of (a hyponym of) c. An alternative approach may use depths of all descendants, but a true normalization is based on a maximum so ald is an average of maximum depths rather than an average of all depths. From ald a *normalized depth* (nd) of c follows:

$$nd(c) = \frac{depth(c)}{ald(c)} \tag{5.2}$$

Finally, the metric can consider the lcs, the deepest (or lowest) concept which is a hypernym (directly or by transitive closure) of both concepts. Following Wu & Palmer (1994), function (5.3) adds consideration for the depth of the lcs compared with depth of the specific concepts. The final metric is below.

$$S_{SchwartzGomez}(c_1, c_2) = \frac{2 * nd(lcs(c_1, c_2))}{nd(c_1) + nd(c_2)} \tag{5.3}$$

Table 5.1: Categorization of similarity and relatedness measures.

Similarity - Path Based	
$S_{WuPalmer}$	Wu & Palmer (1994)
$S_{LeacockChodorow}$	Leacock et al. (1998)
$S_{SchwartzGomez}$	Schwartz & Gomez (2008)
Similarity - Information Content	
S_{Resnik}	Resnik (1999)
$S_{JiangConrath}$	Jiang & Conrath (1997)
S_{Lin}	Lin (1998b)
Relatedness - Path Based	
$R_{HirstStOnge}$	Hirst & St Onge (1998)
$R_{YangPowers}$	Yang & Powers (2006)
Relatedness - Gloss Based	
$R_{BanerjeePedersen}$	Banerjee & Pedersen (2002)
$R_{PartwardhanPedersen}$	Patwardhan & Pedersen (2006)

A concept compared to itself will have a score of 1, while the most dissimilar concepts will have a score of 0.

5.2 Evaluation based on Human Judgments

Semantic similarity and relatedness has a substantial history in computational linguistics signifying its importance to the field. However, an extensive evaluation of similarity and relatedness measures for the task of concept similarity has yet to be carried out. Such an evaluation could benefit applications of measures such as word sense disambiguation or query expansion for information retrieval. This section and the next seek to address this gap in the current body of work by providing results on the performance of various WordNet-based measures for tasks utilizing similarity judgments among concepts (word senses) (Schwartz & Gomez, 2011).

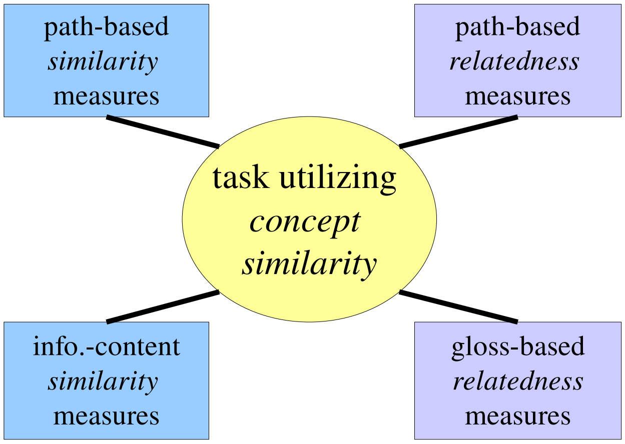

path-based *similarity* measures
path-based *relatedness* measures
task utilizing *concept similarity*
info.-content *similarity* measures
gloss-based *relatedness* measures

Figure 5.1: Depiction of the experimental setup, showing the similarity and relatedness measures as distinct from the task, which is solely focused on similarity.

Table 5.1 lists all of the metrics used in the evaluation. All implementations were either downloaded from the respective authors or provided by the WordNet::Similarity package (Pedersen et al., 2004). Note that the notion of similarity in this work is applied over two *single* concepts; Other works have applied *similarity* over different terms, such as comparing two *pairs* of words when measuring analogy (Turney, 2006). Two types of experiments were implemented over semantic similarity measures. The first is based on adding sense annotations to existing gold-standard judgments of similarity. The second evaluation is based on an application of the measures to WSD. Note that although the task is focused on *similarity*, measures are also included that are more correctly categorized as measuring relatedness. Because relatedness subsumes similarity, these measures should not be excluded from the study. Figure 5.1 shows this distinction between the task focus and the type of measure.

Three datasets of human judgments of similarity were used; namely *RG* (Rubenstein & Goodenough, 1965), *MC* (Miller & Charles, 1991), and *WS-Sim* (Agirre et al., 2009). *RG* and *MC* were created specifically for *similarity* (*MC*'s 28 pairs, listed in Resnik (1999), are a subset of *RG* with

Table 5.2: The inter-annotator agreement and complete pair agreement.

	ITA	**CPA**	**pairs**	**drops**
MC	0.89	0.79	28	0
RG	0.93	0.86	65	0
WS-Sim	0.86	0.73	97	3

ITA: interannotator agreement, **CPA**: complete pair agreement, **pairs**: number of pairs, **drops**: number of instances not annotated due to lack of WordNet sense.

independent judgments). *WS-Sim* is a subset of the WordSim dataset (Finkelstein et al., 2001), which had subjects rate pairs on *relatedness* in general. Agirre et al., (2009) created the similarity subset by including pairs of words with relationships: identical, synonymy, antonymy, hyponymy, and unrelated.

As part of this dissertation, two annotators marked the *RG*, *MC*, and *WS-Sim* datasets with the most similar pair of senses among each pair of words. The original scores of similarity between words were kept for the sense/concept annotated pairs. This approach is motivated by past works which have found the greatest correlation with human judgments by using the maximum similarity over all pairs of senses (Resnik, 1999; Yang & Powers, 2006). WordNet 3.0 served as the sense inventory (Miller et al., 1993). Annotators were able to indicate if a most similar sense was not present in WordNet, in which case the instance was dropped. For example 'jaguar' and 'car' were dropped because the automobile sense of 'jaguar' is not present in WordNet. This *WS-Sim* dataset does not include the pairs which Agirre et al. marked as unrelated, because there was no basis for annotating senses of words considered unrelated.

Statistics of the datasets can be seen in Table 5.2. Inter-annotator agreement (ITA) was calculated as the mean percentage of senses agreed upon within a pair (1, 0.5, or 0 for *completely agreed*, *agreed on one word*, or *completely disagreed* respectively). The complete agreement figure (CPA) is the percentage of pairs with which both words were annotated identically. To finalize each dataset the two annotators were asked to come to an agreement on all instances which were not in complete agreements. There are two types of tests run over the final datasets [1]:

wrd correlation of similarity values based on the word pairs (measures choose the max similarity over all pairs of senses).

cpt correlation of similarity values based on sense annotated (concept) pairs.

Table 5.3 presents the results based on human judgments over all three datasets. Correlations are reported as Spearman rank correlations, avoiding issues arising from non-linear measure outputs as Agirre et al (2009) noted. Normal approximations of confidence intervals at 95% are also presented.

There was no single measure that performed best across all the datasets. When examining the results of the *MC* and *RG* datasets, we see that $R_{PartwardhanPedersen}$ had consistently high correlations. Keep in mind that the *MC* dataset contains a subset of the pairs in the *RG* dataset, with a different set of human judgments. For the *WS-Sim* dataset, which was a distinct set of words and concepts, it was $R_{YangPowers}$ with the highest correlations. In each case, a best performing metric was categorized under *relatedness*, but there is never a significant difference over the top performing metric categorized under *similarity*.

[1] Datasets available for others to use in research.

Table 5.3: Correlation between similarity measure judgments and human judgments for each dataset.

	MC		***RG***		***WS-Sim***	
	wrd	**cpt**	**wrd**	**cpt**	**wrd**	**cpt**
$S_{WuPalmer}$	0.76 [.54, .88]	0.76 [.54, .88]	0.76 [.66, .86]	0.79 [.67, .87]	0.62 [.48, .73]	0.57 [.42, .69]
$S_{LeacockChodorow}$	0.75 [.52, .88]	0.75 [.52, .88]	0.78 [.67, .86]	0.80 [.69, .87]	0.62 [.48, .73]	0.58 [.44, .70]
$S_{SchwartzGomez}$	0.77 [.60, .90]	0.81 [.62, .91]	**0.82** [.71, .88]	0.77 [.65, .85]	0.61 [.47, .72]	0.54 [.38, .66]
S_{Resnik}	0.76 [.55, .88]	0.76 [.53, .88]	0.74 [.61, .83]	0.76 [.63, .84]	0.62 [.47, .73]	0.59 [.45, .71]
$S_{JiangConrath}$	0.82 [.65, .92]	0.85 [.70, .93]	0.78 [.66, .86]	0.80 [.69, .87]	0.60 [.45, .71]	0.51 [.34, .64]
S_{Lin}	0.77 [.56, .89]	0.80 [.61, .91]	0.77 [.64, .85]	0.78 [.66, .86]	**0.64** [.50, .74]	0.58 [.43, .70]
$R_{HirstStOnge}$	0.77 [.56, .89]	0.72 [.47, .86]	0.78 [.66, .86]	0.76 [.63, .85]	0.49 [.32, .63]	0.53 [.37, .66]
$R_{YangPowers}$	0.88 [.75, .94]	0.76 [.55, .88]	**0.82** [.72, .89]	0.78 [.66, .86]	**0.64** [.51, .75]	**0.63** [.49, .74]
$R_{BanerjeePedersen}$	0.81 [.62, .91]	0.76 [.54, .88]	0.72 [.58, .82]	0.69 [.54, .80]	0.49 [.32, .63]	0.46 [.29, .60]
$R_{PartwardhanPedersen}$	**0.92** [.83, .96]	**0.88** [.75, .94]	0.81 [.71, .88]	**0.81** [.71, .88]	0.57 [.41, .69]	0.55 [.39, .67]

Confidence intervals are normal approximations at 95%.

When examining the differences between the 'wrd' and 'cpt' tests, on average, *similarity* measures had higher correlations on the 'cpt' tests within the *MC* and *RG* datasets, while the *relatedness* measures had higher correlations on the 'wrd' tests. This suggests the similarity measures benefit from dealing specifically with concepts rather than ambiguous words, though the differences are small enough that a concrete conclusion can not be drawn. On the other hand, for the *WS-Sim* dataset, the *similarity* measures performed better at the 'wrd' test relative to the 'cpt' test. This difference between the *WS-Sim* dataset and the *MC/RG* dataset may have been due to *WS-Sim* containing more pairs of dissimilar words.

5.3 Evaluation based on Application in Web Selectors Algorithm

The evaluations mentioned for previous application oriented studies used metrics for comparing a target word (or senses of a target word) to other words in context. The assumption is that concepts in context are *related*, but as was previously mentioned *relatedness* does not imply *similarity*. Thus, the measures which are more appropriately categorized as measuring *similarity* (those which do not consider relationships beyond hyponymy, antonymy, and synonymy) may be at a disadvantage. The $S_{SchwartzGomez}$ measure was used in a noun *WSD* algorithm, where noun senses were compared with senses of words that are found to replace that noun in its context (a task calling for *similarity* comparisons) (Schwartz & Gomez, 2008). They experimented over a few similarity and relatedness measures and found *path-based* measures to perform in line with *information content based* and *gloss-based* measures. However, unlike the previously mentioned *WSD* evaluations, this algorithm was focused on achieving top results for a *WSD* task rather than evaluating metrics, and the results were influenced by more than similarity comparisons. This evaluation uses the Web selectors algorithm with restrictions to limit influences beyond similarity comparisons. and also tests on a wider variety of measures.

In order to focus on the impact that a similarity measure has on the accuracy, restrictions are placed on the algorithm. First, senses are chosen by only considering target selectors, words which replace the target word that is being disambiguated. Target selectors are intended to be *similar* to the target sense, while other types of selectors within the algorithm are only intended to be *related*. The system is also setup to only attempt annotations of instances in which it acquires five or more selectors from queries of seven words or more in length. This restriction insures that there is both enough selectors and that the selectors are reliable. Finally, the use of a first sense heuristic as a backoff strategy is turned off to eliminate unnecessary bias.

The testing corpus consisted of the training set from the SemEval-2007 Task 17: Lexical Sample (Pradhan et al., 2007). The lexical sample contained many instances of nouns and verbs, leaving the sample size quite large after the restrictions are placed on the algorithm. Note that the all-words portion of Task 17 contained fewer instances of nouns. The corpus, annotated with WordNet 2.1 senses, was also restricted to eliminate instances of monosemous words according to WordNet. This restriction in addition to those placed on the algorithm are likely to decrease disambiguation accuracy of the algorithm, in order to get a stronger comparison focused on each similarity measure.

Table 5.4 presents the results of the word sense disambiguation experiment. After the restrictions were placed on the corpus, we ended up with 795 instances (431 nouns and 364 verbs). The F1 values shown are calculated based on precision(P) and recall(R) as $F1 = 2 * \frac{P*R}{P+R}$.

Unlike the human judged experiment, this evaluation found one measure performs significantly better than any other measure in this experiment. The information-content similarity measure of Jiang and Conrath ($S_{JiangConrath}$) gives us the top results for both the noun and verb portions of the corpus. All of the relatedness measures ($R_{BanerjeePedersen}$, $R_{PartwardhanPedersen}$, $R_{YangPowers}$) along with the S_{Lin} measure performed approximately equally with over 10.4% more error than the $S_{JiangConrath}$ measure. The path-based similarity measures were all among the least effective for the task.

Table 5.4: Results of the application-oriented evaluation on the SemEval-2007 Task17.

	noun	**verb**	**both**
$S_{WuPalmer}$	41.5	56.3	48.3
$S_{LeacockChodorow}$	44.1	59.3	51.1
$S_{SchwartzGomez}$	48.0	-	-
S_{Resnik}	46.3	51.1	48.5
$S_{JiangConrath}$	**59.6**	**65.1**	**62.1**
S_{Lin}	52.6	57.8	54.9
$R_{HirstStOnge}$	50.9	55.1	52.8
$R_{YangPowers}$	53.2	54.6	53.9
$R_{BanerjeePedersen}$	49.9	57.7	53.5
$R_{PartwardhanPedersen}$	50.6	61.5	55.6

Results are F1 values which are broken down by part of speech.

One can see improvement from all measures between noun and verb instances. Among the relatedness measures, the differences in values indicate that the $R_{YangPowers}$ measure may be better suited for nouns, while the $R_{PartwardhanPedersen}$ method may be stronger with verbs. The suspicion is that the verb results were higher overall because the verb selectors were more often acquired with surrounding context, and were thus more reliable than noun selectors which were more often acquired at the beginning or end of a sentence. Had the algorithm not been restricted to focus on similarity, the noun results would have been higher as was reported originally by Schwartz and Gomez (2008).

5.4 Discussion

This chapter presented a novel semantic similarity measure as well as evaluations of WordNet-based semantic similarity and relatedness measures (included the one presented here) focused on *concept similarity*. One type of experiment was based on human judgments and the other was an application-oriented task. Interestingly, the results found metrics categorized as measuring relatedness to be strongest in correlation with human judgments of concept similarity, though the difference in correlation is small. On the other hand, an information content metric, categorized as measuring similarity, is notably strongest according to the application-oriented evaluation. In particular, the measures of Patwardhan and Pederson (2006), and Yang and Powers (2006) had consistently high correlations with human judgments. Both of these measures were categorized as more broad *relatedness* measures, though the best performing *similarity* measures were not significantly lower for any of the datasets. For the application-oriented experiment, the *similarity* measure of Jiang and Conrath (1997) clearly gave us the best results with an error reduction of 10.4% over the next best measure.

There are several possible extensions to this work to provide additional insights about similarity measures. The existing gold-standard judgments of similarity that were annotated with senses only included nominal concepts. To address this drawback, a human annotated dataset of verb pairs could be created. Additionally, one could replicate experiments over different versions of WordNet as an evaluation of the WordNet improvements. Never the less, the results of this study alone are intended to impact work in computational linguistics when a task calls for similarity judgments over concepts.

6 CONCLUSIONS

The primary goal of this work was to effectively use knowledge acquired from the Web in problems of *semantic interpretation*. The knowledge utilized was described as *common sense knowledge* (*CSK*), knowledge which helps one with understanding and perception in their everyday life. The acquisition approaches utilized the idea of searching the Web with context rather than searching with individual words. Towards semantic interpretation, the focus was on the central problem of *word sense disambiguation* (*WSD*), which tries to determine the meaning of words in a sentence based on the context of each word. It is the hope that improvements to *semantic interpretation* will be able to benefit real world technologies such as machine translation, question answering, web search, sentiment analysis, and text mining. Furthermore, outside of technological advances, semantic interpretation has benefits for fields such as cognitive psychology, neurology, psycholinguistics, and other aspects of cognitive science. A model of understanding language which is useful to a computer could be a good direction of exploration into the mechanics of the human mind.

Through experiments in *WSD*, this work validated the idea of using automatically acquired *CSK* from the Web for aspects of semantic interpretation. In the case of evaluating a database of common sense knowledge (the *CSKB*), integrating the knowledge into an existing minimally-supervised disambiguation system significantly improved results with a 20.5% error reduction. Similarly, the Web selectors disambiguation system, which acquires knowledge directly as part of the algorithm, achieved results comparable with top minimally-supervised systems, an F-score of 80.2% on a standard noun disambiguation task. An impact analysis of the various types of selectors and the amount of data acquired, suggest stronger results could be achieved with more selectors. The results for both major approaches were comparable to most frequent sense baselines and other top minimally-supervised systems. Lastly, the evaluation of semantic similarity and relatedness measures found metrics categorized as measuring relatedness to be strongest in correlation with human judgments. On the other hand, an information content metric, categorized as measuring similarity, is notably strongest according to the application-oriented experiment.

Achievements of this dissertation can be broken into three parts: two major works on knowledge acquisition and knowledge application, and one additional work on semantic similarity. The first work, concerning the automatic acquisition of a database of *common sense knowledge*, gathered relationships by searching the Web with automatically constructed queries. Results were run through a statistical parser to verify that the results from the Web matched an intended structure. This approach then took the knowledge of word relationships and induced knowledge about concepts by using an information-theoretic analysis over WordNet. The *Web selectors* research, the second major work presented, acquired knowledge directly within a word sense disambiguation algorithm. Selectors are words which can take the place of a target word within its local context, and they serve as a source of concepts which should be similar to the sense of the target word. The Web selectors and the database of *CSK* both employed the idea of searching the Web with context in order to gain more targeted results. Lastly, this dissertation presented work in semantic similarity and relatedness measures which are useful tools for semantic interpretation using acquired knowledge. This area of work included the development of a novel path-based similarity measure, as well as evaluations of measures under tasks of concept similarity.

The key contributions from this dissertation to the field of NLP spanned all aspects of the research. During acquisition of the common sense knowledge database (*CSKB*), a novel idea was used by incorporating a statistical parser to validate the syntactic structure of results from the Web. In order to create more applicable knowledge about concepts rather than ambiguous words the *CSKB* system utilized information theoretic approaches to generalize information about concepts in WordNet. This novel analysis could be applied to data containing word relationships of other types. The Web selectors algorithm for *WSD* reworked a previous approach to disambiguation which required dependency parses into a method able to leverage the large amounts of data on the Web. Additionally, the Web selectors algorithm had an original component of incorporating the selectors from words in context in addition to those of the target word. With respect to the research in semantic similarity, this dissertation contributed a path-based method that normalized depths according to the subgraphs. An evaluation was presented with a specific focus on concept similarity, filling a gap in past semantic similarity studies which focused entirely on relatedness or only evaluated judgments over words rather than concepts. Lastly, the *CSKB* is available for other researchers to use as a resource; This is the first work to acquire a more general knowledge describing what is often found in or on something.

The research achievements fit this dissertation's primary objective to investigate the effective acquisition of lexical knowledge from the Web to perform aspects of semantic interpretation. The Web provided an unprecedented amount of unannotated sentences from which to gain knowledge useful for semantic interpretation. Both the work on the *Web selectors* and *CSKB* utilized the idea of searching the Web with context and found the acquired knowledge to be helpful for *WSD*, an aspect of semantic interpretation. Accuracy was increased on an existing state-of-the-art disambiguation system using the *CSKB*, and the *Web selectors* disambiguation approach was found to rival top *minimally-supervised* systems. Overall, this work contributes a piece to broad goals of computational linguistics: solving the puzzle of how humans understand natural language, and enabling technologies that make a positive impact on society.

6.1 Future Directions

This section highlights a few possible future directions of research enabled by the work in this dissertation. These directions include applying the techniques introduced in this work to other aspects of semantic interpretation, both supervised and minimally-supervised, analyzing relationship data provided by others, and improvements to the particular techniques of this work by developing new methods of searching the Web with context. Below, one will find details of how such research may proceed.

One of the largest areas for future work is to apply acquired *CSK* to aspects of semantic interpretation in addition to *word sense disambiguation*. Other aspects include *prepositional phrase attachment* (*PP attachment*), *anaphora resolution* (*AR*), and *named entity recognition* as defined in Section 2.2. These algorithms will follow in a minimally-supervised fashion much like the *WSD* approach completed in this dissertation.

To get an idea of how the approaches in this dissertation may help these problems, consider the problems of *PP attachment* and *anaphora resolution*. *PP attachment*, though dealing with the syntactic structure, requires semantic knowledge in many cases. The *CSKB* could be used in order to determine which word or phrase is being modified by the prepositional phrase. The algorithm can proceed by identifying the prepositional phrase and the relationship it describes. In sentences (28) and (29), the *CSKB* would ideally indicate that it is common for grapes to be in a refrigerator, but not a person. Thus, in (28) the phrase would be attached to 'the grapes' while in (29) the attachment happens at 'she ate'.

(28) *She ate the grapes in the refrigerator.*

(29) *She ate the grapes at home.*

In the case of *AR* one is trying to determine the antecedent for a reference word (Jurafsky & Martin, 2000). In sentence (30), the pronoun 'it' is referencing an antecedent. 'Hank', 'ball', and other entities listed in previous sentences in context would be candidate antecedents. One method to determine the antecedent could use the *CSKB* to find relationships between candidate antecedents and the context of the pronoun. For example, the *CSKB* may indicate a 'ball' travels, and a 'person' can run. The antecedent with the strongest relationships would thus be chosen. In the second approach to *AR*, *Web selectors* would be acquired for the pronouns or reference word. The predicted antecedent would be chosen by comparing similarity of the selectors with the candidates. For example, in (30) selectors for 'it' may include 'baseball', 'object', and 'ball' itself.

(30) *After Hank hit the ball with the bat, it traveled far.*

Although *CSK* acquired from the Web can be applied to semantic interpretation algorithms that do not require training data, the use of *CSK* within supervised algorithms is worth investigation. There is a chance knowledge from the Web could provide information not otherwise available. Typical supervised systems, utilizing standard machine learning algorithms such as maximum entropy or support vector machines, rely on a set of features indicating syntactic and lexical information of the context of a given target word (Gildea & Jurafsky, 2002; Gildea & Palmer, 2002; Dang & Palmer, 2005; Pradhan et al., 2005; Dligach & Palmer, 2008; Schwartz et al., 2008). An additional feature could be a list of *selectors* obtained from the Web for arguments.

The motivation behind a *Web selectors* feature is the potential that a concept can be represented more robustly by its set of selectors than by a word itself. Supervised systems learn entirely from the features of examples they are trained on. In the case of lexical features (features based on words), generalization from one sentence to another can be limited. Consider a system that was trained on sentence (31), seen previously in Figure 2.3. When using a trained system to annotate sentence (32), the lexical features of the nouns (‘surface tension’ verse ‘pressure’, ‘liquid’ verse ‘fluid’, and ‘capillary’ verse ‘tube’) will be almost entirely different. However, the verb sense and semantic roles for the noun phrases should be annotated nearly identically. Web selectors could provide a strong clue to the similarity in that each constituent will likely have the same selectors.

(31) Surface tension will draw liquid into a capillary.

(32) Pressure will draw fluid through a tube.

Additionally, when lexical features (features based on words) are included in current systems they are ambiguous (their disambiguation being the task of *WSD*), leading to matching lexical features of different senses of words. Selectors help with this situation under the assumption that a different sense of a word will have a different set of selectors.

There are many lexical knowledge sources of noun-noun relationships. Girju et al. (2007) plus Hendrickx et al. (2010) provide a good overview. The information-theoretic concept analysis which was used to induce concept information from noun-noun relationships could be applied to other existing sources of noun-noun relationships. As was found in this dissertation, in many situations knowledge about concepts is more beneficial than knowledge about ambiguous words. Additionally, the method could be expanded to work in both directions, concept-concept relations form the noun-noun relations.

In order to improve the Web acquisition techniques introduced in this dissertation, one could look into reconstructing phrases, such as Web queries, in a fashion where the meaning is still maintained. The precision results from requiring a minimum numbers of selectors in *Web selector* experiment 1 (Table 4.3) gives promise to the idea that more selectors can improve *WSD* accuracy. Furthermore, Figure 4.3 indicated that there is a loss of similarity between a selector and its target word as queries get shorter. Finally, considering that the Web selectors algorithm was only able to acquire selectors for 54.7% of word instances (computed from Table 4.1), the ability to create alternative *web queries* would be very helpful. Essentially, this idea of alternative queries is concerned with the trade-off between the quality and quantity of selectors. As one shortens a query to receive more quantity, the quality goes down due to a less accurate local context. One may be able to side-step this trade-off by searching with alternative queries that capture just as much local context. For example, the query (33) can be mapped into the passive transformation (34).

(33) He * the strikers at the rally

(34) The strikers were * at the rally by him

Despite being motivated to improve results for selector acquisition, alternative queries can also assist with creating the *CSKB*. The idea is to create more *web queries* using alternative query construction based on a *search phrase*. Since search phrases are given by hand, the supervision of the algorithm would be minimized as more data could be gathered through a single search phrase.

The future directions enabled by this dissertation are broken down into many tasks, but they are inter-related under the idea of searching with context, and subsequent studies may combine tasks. Many of these improvements will be concerned with both the *CSKB* and *Web selectors*, addressing commonalities between the two such as query construction or applications to similar problems. A single study may incorporate the query reconstruction and supervised semantic interpretation using Web selectors. In the end, the directions of future research may go down many paths and the implications of such work are rich with benefits for computational linguistics, cognitive models of language understanding, and real-world technologies.

LIST OF REFERENCES

Abney, S. (2004). Understanding the yarowsky algorithm. *Computational Linguistics*, *30*(3), 365–395.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., & Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceeding of the North American Chapter of the Association for Computational Linguistics*, (pp. 19–27). Boulder, Colorado.

Agirre, E., Ansa, O., & Martínez, D. (2001). Enriching wordnet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. Pittsburg, Pennsylvania.

Agirre, E., López de Lacalle, O., Fellbaum, C., Hsieh, S.-K., Tesconi, M., Monachini, M., Vossen, P., & Segers, R. (2010). Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, (pp. 75–80). Uppsala, Sweden.

Agirre, E., & Martínez, D. (2001). Learning class-to-class selectional preferences. In *Proceedings of the 2001 Conference on Computational Natural Language Learning*, (pp. 1–8). Morristown, NJ, USA.

Agirre, E., & Martínez, D. (2004). Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of EMNLP 2004*, (pp. 25–32). Barcelona, Spain.

Agirre, E., & Soroa, A. (2009). Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, (pp. 33–41). Athens, Greece: Association for Computational Linguistics.

Allen, J. (1994). *Natural Language Understanding*, (2nd ed.). The Benjamin/Cummings Publishing Company, Inc.

Banerjee, S., & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City, Mexico.

Budanitsky, A., & Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*. Morristown, NJ, USA.

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

Buscaldi, D., & Rosso, P. (2007). UPV-WSD : Combining different WSD methods by means of fuzzy borda voting. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, (pp. 434–437). Prague, Czech Republic.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, (pp. 1306–1313). Atlanta, Georgia, USA.

Chan, Y. S., Ng, H. T., & Zhong, Z. (2007). NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, (pp. 253–256). Prague, Czech Republic.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics (NAACL-2000)*, (pp. 132–139). San Francisco, CA, USA.

Chklovski, T., & Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*. Barcelona, Spain.

Church, K. W., & Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics (ACL'89)*, (pp. 76–83).

Clark, P., & Harrison, P. (2009). Large-scale extraction and use of knowledge from text. In *Proceedings of the fifth International Conference on Knowledge Capture*, (K-CAP '09), (pp. 153–160).

Clear, J. H. (1993). The british national corpus. (pp. 163–187).

Curtis, J., Cabral, J., & Baxter, D. (2006). On the application of the cyc ontology to word sense disambiguation. In *FLAIRS-19: Proceedings of the nineteenth Florida Artificial Intelligence Research Society*. Melbourne Beach, Florida: AAAI Press.

Dang, H. T., & Palmer, M. (2005). The role of semantic roles in disambiguating verb senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, (pp. 26–28). University of Michigan.

Diab, M. (2004). Relieving the data acquisition bottleneck in word sense disambiguation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, (pp. 303–310). Barcelona, Spain.

Dligach, D., & Palmer, M. (2008). Novel semantic features for verb sense disambiguation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL'08): Short Papers*, (pp. 29–32). Columbus, Ohio.

Edmonds, P., & Cotton, S. (2001). Senseval-2: Overview. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, (pp. 1–5). New Brunswick, NJ.

Fillmore, C. (1968). The case for the case. In E. Bach, & R. Harms (Eds.) *Universals in Linguistic Theory*. New York: Rinehart and Winston.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2001). Placing search in context: The concept revisited. In *ACM Transactions on Information*

Systems.

Gabrilovich, E., & Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, *34*, 443–498.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, (pp. 233–237). Morristown, NJ, USA.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*(3), 245–288.

Gildea, D., & Palmer, M. (2002). The necessity of syntactic parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, (pp. 239–246). Philadelphia, Pennsylvania.

Girju, R., Badulescu, A., & Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2003)*, (pp. 1–8). Morristown, New Jersey.

Girju, R., Badulescu, A., & Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics*, *32*(1), 83–135.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., & Yuret, D. (2007). Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th In-*

ternational Workshop on Semantic Evaluations (SemEval-2007), (pp. 13–18). Prague, Czech Republic.

Gomez, F. (2001). An algorithm for aspects of semantic interpretation using an enhanced wordnet. In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, (pp. 1–8). Morristown, NJ, USA.

Gomez, F. (2004a). Building verb predicates: A computational view. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, (pp. 359–366). Barcelona, Spain.

Gomez, F. (2004b). Grounding the ontology on the semantic interpretation algorithm. In *Proceedings of the Second International WordNet Conference*, (pp. 124–129). Masaryk, Czech Republic.

Gonzalo, J., & Verdejo, F. (2006). Chapter 8: Automatic acquisition of lexical information and examples. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*, (pp. 253–274). Springer.

Grimshaw, J. (1990). *Argument Structure*. Cambridge, MA: MIT Press.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, (pp. 539–545). Nantes, France.

Hearst, M. A. (1998). Automated discovery of wordnet relations. In C. Fellbaum (Ed.) *WordNet: An Electronic Lexical Database*. MIT Press.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, (pp. 33–38). Uppsala, Sweden.

Hirst, G., & St Onge, D. (1998). *Lexical Chains as representation of context for the detection and correction malapropisms*. The MIT Press.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, (pp. 57–60). Sydney, Australia.

Hu, X., Sun, N., Zhang, C., & Chua, T.-S. (2009). Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, (pp. 919–928). Hong Kong, China.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, *24*(1), 2–40.

Ide, N., & Wilks, Y. (2006). Chapter 3: Making sense about sense. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*. Springer.

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference of Research on Computational Linguistics (ROCLING X)*. Taiwan.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, (first ed.). Prentice Hall.

Klein, D., & Manning, C. D. (2003). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, (pp. 3–10).

Koeling, R., McCarthy, D., & Carroll, J. (2005). Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the conference on Human Language Technology and Experimental Methods in NLP*, (pp. 419–426). Morristown, NJ, USA.

Komachi, M., Kudo, T., Shimbo, M., & Matsumoto, Y. (2008). Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, (pp. 1011–1020). Honolulu, Hawaii.

Krovetz, R. (1998). More than one sense per discourse. In *Proceedings of the workshop on Evaluating Word Sense Disambiguation Systems*. Sussex, England.

Lapata, M., & Lascarides, A. (2006). Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, *27*, 85–117.

Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense indentificaiton. *WordNet: An elctronic database*, (pp. 265–283).

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

Lenat, D. B. (1995). CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33–38.

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

Li, Y., Musilek, P., Reformat, M., & Wyard-Scott, L. (2009). Identification of pleonastic it using the web. *Journal of Artificial Intelligence Research*, *34*, 339–389.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, (pp. 64–71). Madrid, Spain.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of 36th annual meeting on Association for Computational Linguistics(COLING-ACL 98)*, (pp. 768–774). Montreal, Canada.

Lin, D. (1998b). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, (pp. 296–304). Madison, WI, USA.

Liu, H., & Singh, P. (2004). Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, *22*, 211–226.

Màrquez, L., Escudero, G., Martínez, D., & Rigau, G. (2006). Chapter 6: Unsupervised corpus-based methods for wsd. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*, (pp. 132–166). Springer.

Martínez, D., Agirre, E., & Wang, X. (2006). Word relatives in context for word sense disambiguation. In *Proceedings of the 2006 Australasian Language Technology Workshop*, (pp. 42–50).

Martinez, D., de Lacalle, O. L., & Agirre, E. (2008). On the use of automatically acquired examples for all-nouns word sense disambiguation. *Journal of Artificial Intelligence Research*, *33*, 79–107.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, (pp. 280–287). Barcelona, Spain: Association for Computational Linguistics.

Mihalcea, R. (2002). Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Languages Resources and Evaluations LREC 2002*. Las Palmas, Spain.

Mihalcea, R. (2006). Chapter 5: Knowledge-based methods for wsd. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*, (pp. 107–131). Springer.

Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. In *North American Chapter of the Association for Computational Linguistics (NAACL 2007)*. Rochester, NY, USA.

Mihalcea, R., & Moldovan, D. I. (1999). An automatic method for generating sense tagged corpora. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI-99)*, (pp. 461–466).

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). Five papers on wordnet. Tech. rep., Princeton University.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*(1), 1–28.

Miller, G. A., Chodorow, M., L, S., Leacock, C., & Thomas, R. G. (1994). Using a semantic concordance for sense identification. In *In Proc. of ARPA Human Language Technology Workshop*.

Nakov, P., & Hearst, M. A. (2005). Using the web as an implicit training set: application to structural ambiguity resolution. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (pp. 835–842). Morristown, NJ, USA.

Nakov, P., & Hearst, M. A. (2008). Solving relational similarity problems using the web as a corpus. In *Proceedings of Annual Conference of the Association for Computational Linguistics (ACL-08: HLT)*, (pp. 452–460). Columbus, Ohio.

Navigli, R., Litkowski, K. C., & Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of SemEval-2007*, (pp. 30–35). Prague, Czech Republic.

Navigli, R., & Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, (pp. 216–225). Uppsala, Sweden.

Navigli, R., & Velardi, P. (2005). Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, *27*(7), 1075–1086.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*, (pp. 613–619). New York, NY, USA.

Pantel, P., & Pennacchiotti, M. (2006). Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 44th annual meeting of the Association for Computational Linguistics*, (pp. 113–120). Morristown, NJ, USA.

Panton, K., Matuszek, C., Lenat, D., Schneider, D., Witbrock, M., Siegel, N., & Shepard, B. (2006). Common sense reasoning : From cyc to intelligent assistant. In *Ambient Intelligence in*

Everyday Life, (pp. 1–31).

Pasca, M., Lin, D., Bigham, J., Lifchits, A., & Jain, A. (2006). Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 44th annual meeting of the Association for Computational Linguistics (ACL'06)*, (pp. 809–816). Morristown, NJ, USA.

Patwardhan, S., Banerjee, S., & Pedersen, T. (2003). Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, (pp. 241–257). Mexico City, Mexico.

Patwardhan, S., & Pedersen, T. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *EACL 2006 Workshop Making Sense of Sense—Bringing Computational Linguistics and Psycholinguistics Together*, (pp. 1–8). Trento, Italy.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity - Measuring the Relatedness of Concepts. In *Human Language Technology Conference of the NAACL Demonstrations*, (pp. 38–41). Boston, MA.

Pederson, T. (2006). Chapter 6: Unsupervised corpus-based methods for wsd. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*, (pp. 132–166). Springer.

Ponzetto, S. P., & Strube, M. (2007). Knowledge derived from wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research*, *30*, 181–212.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., & Jurafsky, D. (2005). Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. University of Michigan.

Pradhan, S. S., Loper, E., Dligach, D., & Palmer, M. (2007). Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007)*, (pp. 87–92).

Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). *A Comprehensive Grammaer of the English Language*. Longman.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *IEEE International Conference on Computer Vision*. Rio de Janeiro.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, (pp. 17–30).

Ravichandran, D., & Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL conference. Philadelphia, PA.*.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, (pp. 448–453). Montreal, Quebec, Canada.

Resnik, P. (1996). Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, *61*(1-2), 127–59.

Resnik, P. (1997). Selectional preference and sense disambiguation. In *Proceedings of the ANLP Workshop: Tagging Text with Lexical Semantics*. Washington, DC, USA.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, *11*, 95–130.

Resnik, P. (2006). Chapter 11: Wsd in nlp applications. In E. Agirre, & P. Edmonds (Eds.) *Word Sense Disambiguation: Algorithms And Applications*, (pp. 299–337). Springer.

Richardson, R., & Smeaton, A. F. (1995). Using wordnet in a knowledge-based approach to information retrieval. Tech. rep., School of Computer Applications, Dublin City University.

Riloff, E., & Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, (pp. 117–124).

Rubenstein, H., & Goodenough, J. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*, 627–633.

Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2007). Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Data Knowledge Engineering*, *61*(3), 484–499.

Schubert, L. (2002). Can we derive general world knowledge from texts? In *Proceedings of the second international conference on Human Language Technology Research*, (pp. 94–97). San Diego, California.

Schubert, L., & Tong, M. (2003). Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning - Volume 9*, (pp. 7–13). Morristown, NJ, USA.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, *24*(1), 97–123.

Schwartz, H. A., & Gomez, F. (2008). Acquiring knowledge from the web to be used as selectors for noun sense disambiguation. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, (pp. 105–112). Manchester, England.

Schwartz, H. A., & Gomez, F. (2009a). Acquiring applicable common sense knowledge from the web. In *Proceedings of the NAACL-2009 Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, (pp. 1–9). Boulder, Colorado.

Schwartz, H. A., & Gomez, F. (2009b). Using web selectors for the disambiguation of all words. In *Proceedings of the NAACL-2009 Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, (pp. 28–36). Boulder, Colorado.

Schwartz, H. A., & Gomez, F. (2010). Ucf-ws: Domain word sense disambiguation using web selectors. In *Proceedings of SemEval-2010: the 5th International Workshop on Semantic Evaluation*, (pp. 392–395). Uppsala, Sweden.

Schwartz, H. A., & Gomez, F. (2011). Evaluating semantic metrics on tasks of concept similarity. In *FLAIRS-24: Proceedings of the twenty-fourth Florida Artificial Intelligence Research Society*. Palm Beach, Florida: AAAI Press.

Schwartz, H. A., Gomez, F., & Millward, C. (2008). A semantic feature for verbal predicate and semantic role labeling using svms. In *FLAIRS-21: Proceedings of the twenty-first Florida Artificial Intelligence Research Society*, (pp. 213–218). Coconut Grove, Florida.

Sinha, R., & Mihalcea, R. (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007)*. Irvine, CA.

Snyder, B., & Palmer, M. (2004). The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain.

Stetina, J., Kurohashi, S., & Nagao, M. (1998). General word sense disambiguation method based on a full sentential context. In *the workshop on Usage of WordNet in Natural Language Processing*, (pp. 1–8). Montreal, Canada.

Strat, T. M., & Fischler, M. A. (1991). Context-based vision: Recognizing objects using information from both 2d and 3d imagery. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, *13*, 1050–1065.

Summer, D., & Gadsby, A. (Eds.) (2002). *Longman Dictionary of American English*, (3rd 'new' ed.). New York, USA: Addison Wesley Longman.

Swier, R., & Stevenson, S. (2004). Unsupervised semantic role labeling. In *Proccedings of the Conference on Emperical Methods in Natural Language Processing*, (pp. 95–102). Barcelona, Spain.

Szumlanski, S. R., & Gomez, F. (2010). Automatically acquiring a semantic network of related concepts. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM-10)*, (pp. 19–28). Toronto, Ontario, Canada.

Thelen, M., & Riloff, E. (2002). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in NLP (EMNLP-02)*. Philadelphia, Pennsylvania, USA.

Tjong Kim Sang, E., & De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, (pp. 142–147). Edmonton, Canada.

Torralba, A., Murphy, K. P., & Freeman, W. T. (2010). Using the forest to see the trees: exploiting context for visual object detection and localization. *Communications of the ACM*, *53*(3), 107–114.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics*, *32*, 379–416.

Turney, P. D. (2008). The latent relation mapping engine: algorithm and experiments. *Journal of Artificial Intelligence Research*, *33*, 615–655.

Vasilescu, F., Langlais, P., & Lapalme, G. (2004). Evaluating variants of the lesk approach for disambiguating words. In *Proceedings of LREC 2004*, (pp. 633–636). Lisbonne.

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd. Annual Meeting of the Association for Computational Linguistics*, (pp. 133–138). New Mexico State University, Las Cruces, New Mexico.

Yang, D., & Powers, D. M. W. (2005). Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38*, (pp. 315–322). Darlinghurst, Australia: Australian Computer Society, Inc.

Yang, D., & Powers, D. M. W. (2006). Verb similarity on the taxonomy of wordnet. In *Proceedings of the Third international WordNet Conference (GWC-06)*. Jeju Island, Korea.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, (pp. 189–196). Cambridge, Massachusetts, USA.

Yates, A., & Etzioni, O. (2009). Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, *34*, 255–296.

Yu, C.-H., & Chen, H.-H. (2010). Commonsense knowledge mining from the web. In *Proceedings of the Twenty-Fourth meeting of the Association for the Advancement of Artificial Intelligence (AAAI-10)*. Atlanta, GA.

Yuret, D. (2007). KU: Word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, (pp. 207–214). Prague, Czech Republic.