

Acquiring Applicable Common Sense Knowledge from the Web

Hansen A. Schwartz and Fernando Gomez

School of Electrical Engineering and Computer Science

University of Central Florida

Orlando, FL 32816, USA

{hschwartz, gomez}@cs.ucf.edu

Abstract

In this paper, a framework for acquiring common sense knowledge from the Web is presented. Common sense knowledge includes information about the world that humans use in their everyday lives. To acquire this knowledge, relationships between nouns are retrieved by using search phrases with automatically filled constituents. Through empirical analysis of the acquired nouns over WordNet, probabilities are produced for relationships between a concept and a word rather than between two words. A specific goal of our acquisition method is to acquire knowledge that can be successfully applied to NLP problems. We test the validity of the acquired knowledge by means of an application to the problem of word sense disambiguation. Results show that the knowledge can be used to improve the accuracy of a state of the art unsupervised disambiguation system.

1 Introduction

Common sense knowledge (CSK) is the knowledge we use in everyday life without necessarily being aware of it. Pantou et al. (2006) of the Cyc project, define common sense as “the knowledge that every person assumes his neighbors also possess”. Although the term common sense may be understood as a process such as reasoning, we are referring only to knowledge. 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 are possible.

To show the need for this information more clearly we provide a couple sentences:

She put the batter in the refrigerator. (1)

He ate the apple in the refrigerator. (2)

In (1), we are dealing with lexical ambiguity. There is little doubt for us 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*. We know it is common for food to be found in a refrigerator and so we easily resolve batter 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 (2). In this case, it is difficult for a computer to determine if “he” is in the refrigerator eating an apple or if the “apple” which he ate was in the refrigerator. Like the previous example, the knowledge that food is commonly found in a refrigerator and people are not, leads us 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”.

Unfortunately, there are not many sources of *CSK* readily available for use in computer algorithms. Those sets of knowledge that are available, such as the CYC project (Lenat, 1995) or ConceptNet (Liu and Singh, 2004) rely on manually provided or crafted data. Our aim is to develop an automatic approach to acquire *CSK*¹ by turning to the vast amount of unannotated text that is available on the Web. In turn, we present a method to automatically retrieve and analyze phrases from the Web.

¹data available at: <http://eecs.ucf.edu/~hschwartz/CSK/>

We employ the use of a syntactic parser to accurately match syntactic patterns of phrases acquired from the Web. The data is analyzed over WordNet (Miller et al., 1993) in order to induce knowledge about word senses or concepts rather than words. Finally, we evaluate whether the knowledge by applying it to the problem of *word sense disambiguation*.

2 Background

The particular type of *CSK* that we experiment with in this paper is described formally as follows:

A relationship, $e1\mathbf{R}e2$, exists between entities $e1$ and $e2$ if one finds “ $e1$ is \mathbf{R} $e2$.”

Some examples include: “a cup is on a table” and “food is in a refrigerator”, which would result in relationships: *cup***on***table* and *food***in***refrigerator*. The next section attempts to make the relationship more clear, as we provide a brief linguistic background of prepositions and relationships.

2.1 Prepositions and Relationships

Prepositions state a relationship between two entities (Quirk et al., 1985). One of the entities is typically a constituent of the sentence while the other is the complement to the preposition. For example, consider the relationship between ‘furniture’ and ‘house’ in the following sentences:

The furniture is...
...at the house.
...on the house.
...in the house.

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

We focus on prepositions which indicate a positive spacial relationship given by 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 1, denoted *on* and *in* throughout the paper. *At*, the 0 dimensional relationship, occurred far less frequently. The

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

Table 1: Spatial dimensions (**dims**) and corresponding prepositions.

sentences below exemplify each of the 1 to 3 dimensional relationships:

on surface The keyboard is on the table.
on line The beach is on US 1.
in area The bank is in New York.
in volume The vegetables are in the bowl.

2.2 Related Work

As a prevalent source of lexical knowledge, dictionary definitions may be regarded as common sense. However, some definitions may be considered expert knowledge rather than *CSK*. The scope of definitions certainly do not provide all necessary information (such as *keys are commonly kept in one’s pocket*). We examine WordNet in particular because the hypernym relation has been developed extensively for nouns. The noun ontology is used in our work to help induce relationships involving concepts (senses of nouns) rather than just among words. This notion of inducing *CSK* among concepts, rather than words, is a key difference between our work and similar research.

The work on VerbOcean is similar to our research in the use of the Web for acquiring relationships (Chklovski and Pantel, 2004). They used patterns of phrases in order to search the Web for semantic relations among verbs. The knowledge they acquire falls into the category of *CSK*, but the specific relationships are different than ours in that they are among verb word forms and senses are not resolved.

ConceptNet was created based on the OpenMind Commonsense project (Liu and Singh, 2004). The project acquired knowledge through an interface on the Web by having users play games and answer questions about words. A contribution of ConceptNet is that it has a wide range of relations. While WordNet provides connections between concepts (senses of words), ConceptNet only provides relationships between word forms.

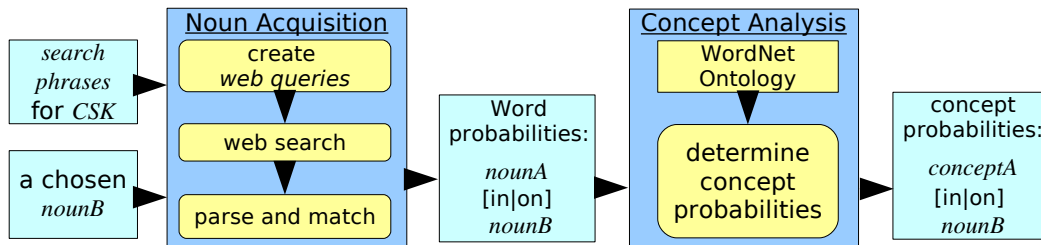


Figure 1: The overall *common sense knowledge* acquisition framework under the assumption that one is acquiring concepts (WordNet synsets) in a relationship with a given *nounB* (word).

A project in progress for over twenty years, CYC has been acquiring *common sense knowledge* about everyday objects and actions stored in 10^6 axioms (Lenat, 1995). The axioms, handcrafted by workers at CYCcorp, represent knowledge rooted in propositions. 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. Only a portion of CYC is available to the public.

Our method for acquiring knowledge is somewhat similar to that of (Hearst, 1992). Patterns are built manually. However, we do not use our manually constructed patterns (referred to as *search phrases*) to query the Web. Instead the *search phrases* are abstract patterns that are used to automatically generate more specific *web queries* by filling constituents based on lists of words.

The SemEval-2007 Task 4 presents a good overview of work in noun-noun relationships (Girju et al., 2007). Our work is related in that the relationships we acquire are between nominals, and in order to build their corpus Girju et al. queried the web with patterns like that of Hearst’s work (Hearst, 1992). The SemEval task was to choose or classify relationships, rather than acquire and apply relationships. Additionally, the relationship classes they use are not necessarily within the scope of *common sense knowledge*.

Similar to our research, in (Agirre et al., 2001) knowledge is acquired about WordNet concepts. They find topics signatures, sets of related words, based on data from the Web and use them for word sense disambiguation. However, the type of relationship between words of a topic signature and the WordNet concept is not made explicit, and the au-

thors find the topic signatures are not very effective for word sense disambiguation.

Finally, we note one approach to using the Web for NLP applications is to acquire knowledge on the fly. Previous work has approached solutions to word sense disambiguation by acquiring words or phrases directly based on the sentences or words being disambiguated (Martinez et al., 2006; Schwartz and Gomez, 2008). These methods dynamically acquire the data at runtime, rather than automatically create a common sense database of relations that is readily available. Additionally, in our current approach, we are able to acquire explicit *CSK* relationships.

3 Common Sense Acquisition

The two major phases of our framework, “Noun Acquisition” and “Concept Analysis”, are outlined in Figure 1 and described within this section.

3.1 Noun Acquisition

The first step of our method is to acquire 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 verify a match with the corresponding *web query*, and the noun(s) filling a missing constituent of the parse are recorded.

The framework itself is very flexible, and it can handle the acquisition of words from other parts of speech. However, to be clear, we focus the explanation on the use of the framework to acquire specific types of relationships between nouns. Below we describe the procedures in more detail.

3.1.1 Creating Web Queries

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

nounA the first noun phrase
nounB the second noun phrase
prep preposition, if any, used in the phrase
verb verb, if any, used in the phrase.

Table 2 lists all of the *search phrases* we use, one of which we use as an example throughout this section:

place *nounA prep nounB*

The verb, “place” in this case, is statically defined as part of the *search phrase*.

Prepositions were chosen to describe the type of relationship we were seeking to acquire as described in the background section. We limited ourselves to the “on” and “in” relationships since these were the most common.

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

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

When noun parameters are provided, determiners or possessive pronouns selected from the list below are included. This provides greater accuracy in our search results.

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

Finally, the undefined parameters are replaced with a “*”. Below is a *web query* created from our *search phrase* where *nounB* is ‘refrigerator’, *prep* is ‘in’, *det* is ‘the’, and *nounA* is undefined:

place * in the refrigerator

3.1.2 Searching the Web

Given a *nounB*, The search algorithm can be summarized through the pseudocode below.

```
for each search_phrase
  for each prep
    for each det
      query = create_query (search_phrase,
                           prep, det, nounB) ;
      samples = websearch (query) ;
```

The searches were carried out through the Google Search API², or the Yahoo! Search Web Services³. Each *search phrase*, listed in Table 2, was run until a maximum of 2000 results were returned. Duplicate *samples* were removed to reduce the effects of websites replicating the text of one another.

²no longer supported by Google

³<http://developer.yahoo.com/search/>

relation	search phrase	voice
on, in	<i>nounA</i> is located <i>prep nounB</i> <i>nounA</i> is found <i>prep nounB</i> <i>nounA</i> is situated <i>prep nounB</i>	passive
on, in	<i>nounA</i> is <i>prep nounB</i> put <i>nounA prep nounB</i> place <i>nounA prep nounB</i> lay <i>nounA prep nounB</i> set <i>nounA prep nounB</i> locate <i>nounA prep nounB</i> position <i>nounA prep nounB</i>	active
on	hang <i>nounA prep nounB</i> mount <i>nounA prep nounB</i> attach <i>nounA prep nounB</i>	active

Table 2: *Search phrases* and relationships used for acquisition of CSK.

3.1.3 Parse and Match

The results we want to achieve in this step should describe a relationship:

nounA is [in | on] *nounB*

We use Charniak’s parser (Charniak, 2000) on both the *web query* and the results returned from the web in order to ensure accuracy. To demonstrate this process, we extend our example, “place * in the refrigerator”.

First, we get a parse with * (*nounA*) represented as ‘something’.

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

We now know the constituent(s) which replace ‘(NN something)’ will be our *nounA*. For example, in the following parse ‘batter’ is resolved as *nounA*.

```
(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))))))
```

The head noun of the matching phrase is determined, which is ‘batter’ in the phrase ‘(DT the) (JJ 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 instead.

The parse also helps to eliminate bad results. For the following sentence, the verb phrase does not

match the parse of the *web query* due to an extra PP, and therefore we do not pull out “for several hours” as *nounA*.

(S1 (S (VP (VP (VB Mix)
 (NP (DT the) (NN sugar))
 (PRT (RP in))
 (PP (TO to) (NP (DT the) (NN dough))))))
 (CC and)
 (VP (VB place)
 (PP (IN for) (NP (JJ several) (NNS hours)))
 (PP (IN in) (NP (DT the) (NN refrigerator))))))

One may note that this malformed sentence is communicating that ‘dough’ is placed in the refrigerator, but the method does not handle this.

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*. This can easily be represented as the probability of a noun, *nA*, being returned to a query for the relationship, **R**, with noun *nB*.

$$p_w(nA, \mathbf{R}, nB)$$

This value along with the other steps we have gone over are stored in a MySQL relational database⁴. One could trace a relationship probability between nouns back to the web results which were matched to a *web query*, and even determine the abstract *search phrase* which produced the web query.

3.2 Concept Analysis

A focus of this work is on going beyond relationships between words. We would like to acquire knowledge about specific concepts in WordNet. In particular, we are trying to induce:

conceptA is [in | on] *nounB*.

where *conceptA* is a concept in WordNet (such as a sense of *nounA*), and *nounB* remains simply a word.

For the analysis, we rely on the vast amount of nouns we are able to acquire in order to create probabilities for relationships of *conceptA***R***nounB*. To get a grasp of the idea in general, consider ‘table’ as a *nounB* of interest. By examining all possible hypernyms of all senses of each *nounA* one will find it is common for abstract entities to be “in a table” (i.e. data in a table), artifacts to be “on a table” (i.e.

⁴<http://www.mysql.com>

cup on a table), and physical things (including living things) to be “at a table” (i.e. the employees at the table). The same idea could be applied in reverse if one acquires knowledge for a set of *nounAs*. However, this paper only focuses on acquiring knowledge for the *nounB* constituent in a *search phrase*.

To begin with, one should note that concepts in WordNet are represented as synsets. A synset is a group of word-senses that have the same meaning. For example, (*batter-1*, *hitter-1*, *slugger-1*, *batsman-1*) is a synset with the meaning “(baseball) a ballplayer who is batting”. We use WordNet version 3.0 in order to take advantage of the latest updates and corrections to the noun ontology. Since a word has multiple senses, we represent the probability that a word-sense, *nAs*, resulted from a query for a relationship, **R** with *nounB* as:

$$p_{ns}(nAs, \mathbf{R}, nB) = \frac{p_w(\text{lemma}(nAs), \mathbf{R}, nB)}{\text{senses}(\text{lemma}(nAs))}$$

where *senses* returns the number of senses of the word (*lemma*) within the word-sense *nAs*. We can then extend the probability to apply to a synset, *syns*, as:

$$p_{syn}(syns, \mathbf{R}, nB) = \sum_{nAs \in syns} p_{ns}(nAs, \mathbf{R}, nB)$$

Finally, we define a recursive function based on the idea that a concept subsumes all concepts below it (hyponyms) in the WordNet ontology:

$$P_c(cA, \mathbf{R}, nB) = p_{syn}(syns(cA), \mathbf{R}, nB) + \sum_{h \in hypos(cA)} P_c(h, \mathbf{R}, nB)$$

where *cA* is a concept/node in WordNet, *syns* returns the synset which represents the concept, and *hypos* returns the set of all direct hyponyms within the WordNet ontology. For example, (*money-3*) is a (*currency-1*), so $P_c(\text{currency-1}, \mathbf{R}, nB)$ receives $p_{syn}(\text{money-3}, \mathbf{R}, nB)$ among others. This type of calculation over WordNet follows much like that of Resnik’s (1999) *information-content* calculation. Note that the function no longer recurs when reaching a concept with no hyponyms and that $P_c(\text{entity-1}, \mathbf{R}, nB)$ is always 1 (*entity-1* is the root node). P_c now represents a probability for the relationship: *conceptA***R***nounB*.

<i>nounB</i>	<i>#nounAs</i>	<i>nounB</i>	<i>#nounAs</i>
basket	3300	boat	2787
bookcase	260	bottle	4742
bowl	5252	cabin	720
cabinet	1474	canoe	163
car	5534	ceiling	1187
city	1432	desk	4770
drawer	1638	dresser	698
floor	2850	house	4627
jar	4462	kitchen	2948
pocket	4771	refrigerator	2897
road	5493	room	5023
shelf	2581	ship	1469
sink	296	sofa	509
table	5312	truck	528
van	301	wall	2285

Table 3: List of nouns which fill the *nounB* constituent in a *search phrase*, and the corresponding occurrences of *nounAs* acquired for each.

4 Evaluation

Our evaluation focuses on the applicability of the acquired *CSK*. We acquired relationships for the 30 nouns listed in Table 3. These nouns represent all possible words to fill the *nounB* constituent of a *search phrase*. The corresponding *#nounAs* indicates the number of *nounAs* that were acquired from the Web for each *nounB*. For example, 4771 *nounAs* were acquired for ‘pocket’. This means 4771 results from the web matched the parse of a *web query* for ‘pocket’ and contained a *nounA* in WordNet (keeping in mind duplicates Web text were removed).

Delving deeper into our example, below are the top 20 *nounAs* found for the relationship *nounAin*pocket.

money, hand, cash, firework, something, dollar, ball, hands, key, coin, pedometer, card, battery, item, phone, penny, music, buck, implant, wallet

As described in the concept analysis section, occurrences of each *nounA* for a given *nounB* lead to p_w values, which in turn are used to produce P_c values for concepts in WordNet. The application of *CSK* utilizes these probabilities rather than simply lists of words or even lists of concepts. However, challenges

were encountered during the noun acquisition step before the probabilities were produced.

Many challenges of the noun acquisition step were overcome through the use of a parser. For example, phrases such as “Palestine is on the road to becoming...” could be eliminated since the parser marks the prepositional phrase “to becoming” as being attached to “the road”. Thus, the parse of the web sample does not match the parse of the *web query* used to acquire it. Other times, noun-noun relationships were common simply because many web pages seem to copy the text of others. This problem was handled through the elimination of duplicate text samples from the Web. In the end, only about one in four results from the Web were actually used. Numbers in Table 3 reflect the result of these eliminations.

Some issues of the acquisition step were not directly addressed in this paper. A domain may tend to be more prevalent on the Internet and skew the *CSK*, such as *fireworkin*pocket. Another example, *babyin*basket was very common due to biblical references. Fictional works and metaphors also provided uncommon relationships dispersed within the results. Additionally, the parser makes mistakes. It was the hope that the *concept analysis* step would help to mitigate some noise from these problems. A final issue was the bottleneck of limited queries per day by the search engines, which restricted us to testing on only the 30 nouns listed.

4.1 Disambiguation System

The *CSK* 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 *notein*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, our knowledge is intended to be used as a reference, consulted by a disambiguation system.

We integrate our knowledge into a state of the art “all-words” word sense disambiguation algorithm. These algorithms are considered unsupervised or

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. Our knowledge 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 our disambiguation system is the publicly available GWSD system (Sinha and Mihalcea, 2007). Sinha and Mihalcea report higher results on the Senseval-2 and Senseval-3 datasets than any of the participating unsupervised system. Additionally, GWSD is compatible with WordNet 3.0 and its output made it easy to integrate our knowledge. Sense predictions from four different graph metrics are produced, and we are able to incorporate our knowledge as another prediction within a voting scheme.

Considering the role of our knowledge as a reference, in some cases we would like the CSK to suggest multiple senses and in others none. For each target noun instance in the corpus, we lookup the $P_c(c, \mathbf{R}, nB)$ value, where c is the WordNet concept that corresponds to a sense of the target noun. We choose nB by matching the phrase “*in|on det nB*” 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 and suggestions 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 CSK suggestions.

4.2 Experimental Corpus

A goal of our work was to acquire data which could be applied to NLP problems. We focus particularly on the difficult problem of *word sense disambiguation*. Due to the lack of sense tagged data, we were unable to find an annotated corpus with instances of all the nouns in Table 3 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).

We created a corpus from sentences in Wikipedia which contained the phrase “*in|on det lemma*”, where *det* is a determiner or possessive pronoun, *lemma* is a noun from Table 3, and *in|on* is a preposition for either relationship described earlier. Below we have provided an example from our corpus where the knowledge from ‘pocket’ can be applied to disambiguate ‘key’.

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 corpus⁵ contained a total of 342 sentences, with one target noun annotated per sentence. The target nouns were selected to potentially fill the *nounA* constituent in the relationship *nounARnounB*, and they were assigned all appropriate WordNet 3.0 senses. Considering the fine-grained nature of WordNet (Ide and Wilks, 2006), 26.3% of the instances were annotated with multiple senses. We also restricted the corpus 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. Because the corpus was built by an author of the work, we asked 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 annotator. Agreement was calculated as:

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

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 4. As a point of comparison, the Senseval 3 all-words task had a 75% agreement on nouns (Snyder and Palmer, 2004). A second evaluation of agreement was also done. The non-author annotations were treated as if they came

⁵available at: <http://eecs.ucf.edu/~hschwartz/CSK/>

	insts	agree	F1_h	F1_{rnd}	F1_{MFS}
on	131	79.9	84.7	28.2	71.0
in	211	80.8	91.9	27.2	67.8
both	342	80.5	89.2	27.6	69.0

Table 4: Experimental corpus data for each relationship (*on*, *in*). **insts**: number of annotated instances; **agree**: inter-annotator agreement %; **F1** values (precision = recall): *h*: human annotation, *rnd*: random baseline, *MFS*: most frequent sense baseline.

	without CSK		with CSK	
	F1_{all}	F1_{indeg}	F1_{all}	F1_{indeg}
on	62.6	63.4	64.9	67.2
in	68.7	69.7	71.6	72.5
both	66.4	67.3	69.0	70.5
ties	37	0	66	72

Table 5: F1 values (precision = recall) on our experimental corpus with and without *CSK*. **F1_{all}**: using all 4 graph metrics; **F1_{indeg}**: using only the indegree metric; **ties**: number of instances where tie votes occurred.

from a disambiguation system in order to get a human upper-bound of performance. Just as the automatic system handled tie votes, when one word had multiple sense annotations, the annotation with the lowest sense number was used. This performance upper-bound is shown as **F1_h** in Table 4.

4.3 Results

Our disambiguation results are presented in Table 5. We found that, in all cases, including *CSK* improved results. It turned out that 54.7% of the noun instances received at least one suggestion from the *CSK*, and 24.5% of the instances received multiple suggestions. It is not clear why the *on* results were slightly below that for *in*. We suspect the *on* portion of the corpus was slightly more difficult because the human annotation (**F1_h**) found a similar phenomenon.

One observation we made when setting up the test was that the indegree metric alone performed slightly better than using the votes of all four metrics. This was not surprising considering Sinha and Mihalcea found the indegree metric by itself to perform only slightly below a combination of metrics on the senseval data (Sinha and Mihalcea, 2007).

Therefore, Table 5 also reports the use of the indegree metric by itself or with *CSK*, **F1_{indeg}**. In these cases we saw the greatest improvements of using *CSK*, producing an error reduction of about 4.5% and outperforming the **F1_{MFS}** value.

Several additional experiments were performed. Note that even during ties, the chosen sense was taken from the predictions and suggestions. When we instead incorporated an MFS backoff strategy for ties, our top results for **F1_{indeg}** with *CSK* dropped to 70.2. We also ran a precision test with no predictions made for tie votes, and found a precision of 71.9% on the 270 instances that did not have a tie for top votes (this also used the indegree metric with *CSK*). All results supported our goal of acquiring *CSK* that was applicable to word sense disambiguation.

5 Conclusion

We found our acquired *CSK* to be useful when incorporated into a word sense disambiguation system, finding an error reduction of around 4.5% for top results. Relationships between nouns were acquired from the Web through a unique search method of filling constituents in a *search phrase*. Samples returned from the Web were restricted by a requirement to match the syntactic parse of a *web query*. The resulting data was analyzed over WordNet to produce probabilities of relationships in the form of *conceptARnounB*, where *conceptA* is a concept in WordNet rather than an ambiguous noun.

In our effort to validate the knowledge through application, many steps along the way were left open for future investigations. First, there is a need to exhaustively search for *CSK* of all nouns and to acquire other forms of *CSK*. With this improvement *CSK* could be tested on a standard corpus, rather than a corpus focused on select nouns and prepositional phrases. Looking into acquisition improvements, a study of the effectiveness of the parse would be beneficial. Finally, the applicability of the knowledge may be increased through a more complex concept analysis or utilizing a more advanced voting scheme.

6 Acknowledgement

This research was supported by the NASA Engineering and Safety Center under Grant/Cooperative Agreement NNX08AJ98A.

References

- Eneko Agirre, Olatz Ansa, and David Martinez. 2001. Enriching wordnet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, USA.
- Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Timothy Chklovski and Patrick Pantel. 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.
- Mona Diab. 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)*, pages 303–310.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of SemEval-2007*, pages 13–18, Prague, Czech Republic, June. Association for Computational Linguistics.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545.
- Nancy Ide and Yorick Wilks, 2006. *Word Sense Disambiguation: Algorithms And Applications*, chapter 3: Making Sense About Sense. Springer.
- Douglas B. Lenat. 1995. CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.
- H. Liu and P Singh. 2004. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22:211–226.
- David Martinez, Eneko Agirre, and Xinglong Wang. 2006. Word relatives in context for word sense disambiguation. In *Proceedings of the 2006 Australasian Language Technology Workshop*, pages 42–50.
- Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 279–286, Barcelona, Spain, July. Association for Computational Linguistics.
- Rada Mihalcea. 2002. Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Languages Resources and Evaluations LREC 2002*, Las Palmas, Spain, May.
- George Miller, R. Beckwith, Christiane Fellbaum, D. Gross, and K. Miller. 1993. Five papers on wordnet. Technical report, Princeton University.
- Kathy Panton, Cynthia Matuszek, Douglas Lenat, Dave Schneider, Michael Witbrock, Nick Siegel, and Blake Shepard. 2006. Common sense reasoning : From cyc to intelligent assistant. In Y. Cai and J. Abascal, editors, *Ambient Intelligence in Everyday Life*, pages 1–31.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.
- Philip Resnik. 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.
- Hansen A. Schwartz and Fernando Gomez. 2008. Acquiring knowledge from the web to be used as selectors for noun sense disambiguation. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 105–112, Manchester, England, August.
- Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. Irvine, CA, September.
- Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *ACL Senseval-3 Workshop*, Barcelona, Spain, July.