Searching for Commonsense

by

Ian Scott Eslick

Submitted to the Media, Arts and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006
August 2006

Author .....
Media, Arts and Sciences
August 11, 2006

Certified by .....
Walter Bender
Senior Research Scientist
Thesis Supervisor

Certified by .....
Hugh Herr
Associate Professor
Thesis Supervisor

Certified by .....
Rada Mihalcea
Assistant Professor, U. of N. Texas
Thesis Supervisor

Accepted by .....
Andy Lippman
Chairman, Department Committee on Graduate Studies

Searching for Commonsense

by

Ian Scott Eslick

Submitted to the Media, Arts and Sciences
on September 11, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Acquiring and representing the large body of "common sense" knowledge underlying ordinary human reasoning and communication is a long standing problem in the field of artificial intelligence. This thesis will address the question whether a significant quantity of this knowledge may be acquired by mining natural language content on the Web. Specifically, this thesis emphasizes the representation of knowledge in the form of binary semantic relationships, such as cause, effect, intent, and time, among natural language phrases.

The central hypothesis is that seed knowledge collected from volunteers enables automated acquisition of this knowledge from a large, unannotated, general corpus like the Web. A text mining system, ConceptMiner, was developed to evaluate this hypothesis. ConceptMiner leverages web search engines, Information Extraction techniques and the ConceptNet toolkit to analyze Web content for textual evidence indicating common sense relationships.

Experiments are reported for three semantic relation classes: desire, effect, and capability. A Pointwise Mutual Infomation measure computed from Web hit counts is demonstrated to filter general common sense from instance knowledge true only in specific circumstances. A semantic distance metric is introduced which significantly reduces negative instances from the extracted hypotheses.

The results confirm that significant relational common sense knowledge exists on the Web and provides evidence that the algorithms employed by ConceptMiner can extract this knowledge with a precision approaching that provided by human subjects.

Thesis Supervisor: Walter Bender
Title: Senior Research Scientist

Thesis Supervisor: Hugh Herr
Title: Associate Professor

Thesis Supervisor: Rada Mihalcea
Title: Assistant Professor, U. of N. Texas

Acknowledgments

This thesis research was originally undertaken in concert with the late Dr. Push Singh. Without the endless hours of discussion and debate over the prior three years I would not have engaged with Commonsense Computing as a research area, nor with this thesis topic specifically. His contributions and collaboration will be greatly missed and I would like to dedicate this thesis to his memory.

I wish to thank my readers, Mr. Walter Bender, Professor Rada Mihalcea. and Professor Hugh Herr and to acknowledge them for their efforts and advice. Deserving particular recognition are Professor Ted Selker and Dr. Henry Lieberman who provided significant input, encouragement and feedback throughout this project.

A heartfelt "thank you" is extended to my Media Lab advisors Mr. Walter Bender and Professor Emeritus Marvin Minsky for their wisdom, advice and vast reserves of patience.

To Professors Andre DeHon, Whitman Richards, Hugh Herr, Patrick H. Winston and Doctors Howie Shrobe, M. Alex O. Vasilescu and Hugo Liu I extend my appreciation for their inspirational example and considerable skills in mentorship. They have helped me move closer to a deep understanding of the processes behind quality research.

I would also like express appreciation to Dr. Barbara Barry for her review of early data, to Kim Reinhold for her initial experiments as a high school intern with our group, and to Peak Xu for his 2005 work in analyzing the OpenMind corpus.

Lastly, I would never have finished this thesis in a timely fashion were it not for the patience, encouragement and editorial skills of my wife, Mary Kelly who always made the time to spread red ink over drafts of this document.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis will review knowledge-acquisition techniques and describe a tool, ConceptMiner, that will significantly enhance the resources available to the designers of user-interfaces. It will do so by outlining and justfiying a methodology that enhances the performance of knowledge-based user interfaces, specifically those employing ambiguous representations based in natural language.

This introduction motivates the utility of knowledge acquired by ConceptMiner, specifically emphasizing a knowledge-based strategy for human-computer interface design that has been advocated by several groups at MIT's Media Laboratory. The strategy is made explicit in the knowledge representations and inference techniques produced by the OpenMind Project and Commonsense Computing Group at the MIT Media Laboratory. Chapter 2 will provide a motivating example that illustrates how the system works and Chapter 3 will emphasize the relevant background for the techniques employed by the system.

1.1 Natural Human-Computer Interaction

Decades of work on the Human-Computer Interface (HCI) have led to many new ways to interact with users. Typically, computer action is directly invoked by the user. This may include a button click, a mouse gesture, or a keyboard command. The paradigmatic limitation today is that computers are only able to do what the

user says; they cannot independently choose actions that satisfy the true goals of the user. However, it is exactly this kind of intent reading that is the basis of the human ability to communicate.

Mainstream attempts to cross this great divide take the form of user modeling. Computer scientists study user behavior with the intent of identifying a policy for taking action that satisfies a model of user preferences. [Horvitz et al., 1998] This modeling effort typically leverages statistical machinery requiring significant training samples. Such systems cannot learn online without making many (typically annoying) mistakes while they are converging on an accurate model. One only has to look at the user interface challenges of Microsoft's "Paperclip" commercial application for evidence [Xiao et al., 2004]. Moreoever, this methodology is domain specific; a specific training set is developed for a particular set of possible policy decisions.

There are other schools of thought that seek to augment machines and programs with information that improves their effective intelligence. For example, projects in Affective Computing aim to augment machine decision making with internal, affective rewards [Ahn and Picard, 2006].

Is it possible to bridge this divide in a new way? The ultimate goal is to properly interpret user communication in order to take action and not to annoy the user when making the inevitable mistake. This means that systems need to be able to guess correctly much of the time, even when encountering novel situations. When our programs are wrong, they should be able to be corrected in a simple, one-shot manner. There are numerous examples of steps in this direction.

Work in user interfaces has illucidated an interesting user interface principle: 'natural mistakes' [Stocky et al., 2004]. This principle is part of a trend of accomodating human social dynamics in the design of user interface [Tzeng, 2004]. The intuition is that if the computer makes a mistake that the user can imagine having made (e.g. a chain of reasoning missing a specific fact), then the user reports a more positive subjective experience.

Advances in text processing applications such as search [Liu et al., 2002] and travel guides [Musa et al., 2003] exploiting heuristic or factual knowledge bases built

from natural language phrases [Liu et al., 2002].

Satisfying intent also requires understanding more about the problems the user is engaged in solving. Today many applications are involved in the processing of text, speech, and other forms of media. Applications ranging from E-mail to text messenging, search and blogging are all highly dependent on the semantics of the constitutent natural language content. Moreoever, the tagging or labeling of non-text content [Vanderwal, 2005] [Liu, 2002] is often in the form of words or phrases that contain Commonsense semantics in the form of natural language; thus bringing non-text content back into the conceptual domain of language.

To achieve the goal of satisfying user intent, an application must have a model of user desires to analyze, retrieve, and display content on the basis of "soft measures" such as similarity, reference, affect, and salience. Each of these soft measures requires knowledge about the relationships among natural language terms. There are three primary ways such knowledge can be acquired:

- Ask the user to input needed knowledge
- Train the system offline using examples
- Have a pre-existing source of required background knowledge

Each of these possible approaches has associated challenges: users do not react well to frequent interruptions in their primary goals, developing labeled examples to train the system is money and manpower intensive[1] for all but the most widespread and common applications (i.e. natural language parsing), and the existing sources of background knowledge fall far short of covering even the commonsense of a five year old child. The scope of of background knowledge is daunting, but attacks on this challenge have been attempted over the years by researchers in Artificial Intelligence interested in understanding human common sense reasoning.

[1]The Cyc project has absorbed almost $70M in funding over its lifetime [Sowa, 1990]

1.2 Commonsense Knowledge and Reasoning

The problem of Commonsense reasoning was first introduced by John McCarthy in 1958 in his paper titled "Programs with Common Sense." [McCarthy, 1986]. Little progress [Mueller, 1999] has been made on the problems he articulates there. The key observation of that paper was to illustrate that interpretation of natural language content requires that we first have a model of all the probable motivations, causes, and effects of the events in the text as well as more basic knowledge such as typical states, properties, and relationships among the referenced actors and objects. For the past forty years, most of the work on natural language has been focused on modeling elements of syntax, reference and anaphora, the structure of the lexicon, such as polysymy and meronymy, and semantics.

The dominant approach to semantics and knowledge has been governed by McCarthy's view that formal expressions in an appropriate logic will suffice to unambiguously define the relationships among all the possible referents in natural language.

But what if this approach to representation is wrong, or at the least incomplete? Sowa's articulation of "Knowledge Soup" [Sowa, 1990] places classic arguments from Kant and Leibniz into the modern context; he emphasizing that no empirically determined concept is likely to yield itself to a formal, logistic definition.

1.2.1 Embracing Ambiguous Representations

If Sowa is right, and being precise, correct and unambiguous is not the correct representational methodology, then what is? Perhaps the purpose of background knowledge is to constrain the universe of possible interpretations to a small set that can be reasoned about using different, multiple, forms of knowledge. Instead of having a formally correct representation, systems should aim for the lesser goal of exhibiting mostly-correct behavior that allows for *natural* mistakes where users are empowered to teach the machine to avoid such mistakes in the future.

One approach to generating correct behavior is to use a preponderance of background knowledge to cut through the murky waters of the knowledge soup. To achieve

a correct answer, the system appeals to higher-level conflict resolution method to discriminate among alternatives [Minsky, 1988] [Minsky, 2006].

Such a system might learn domain-general ways to resolve problems by finding general strategies for learning and reasoning that are effective for particular inferences or actions. The epistemology of such a system lies outside the scope of inquiry of this thesis, but is an essential part of the worldview this research aims to enhance. This thesis addresses the smaller ambition of enabling new user interface paradigms where the computer doesn't have to understand or learn as well as the human - it just has to behave enough as if it does in order to achieve powerful and positive impact on the user experience.

I hypothesize that a great deal of progress can be made on these user interface problems by embracing noisy semantics and looking to inference algorithms to accomodate the noise. This point was argued in [Schubert and Tong, 2003] and has been an implicit assumption behind the OpenMind project [Singh, 2002].

1.2.2 OpenMind and Volunteer Contributors

The OpenMind project was initiatied to discover exactly what kinds of knowledge could be extracted from a community of volunteers on the Web as an attack on the Commonsense knowledge problem. The project took the position that collecting simple language about everyday knowledge that did not exist in dictionaries might yield knowledge that could be automatically manipulated, but maintain a canonical representation understandable by any speaker of the English language. In the short period of two years, OpenMind had available nearly a million expressions of Commonsense statements, in both structured and unstructured English.

The best answer to whether the Open Mind corpus could be used for reasoning has been provided by Liu and Singh's ConceptNet [Liu and Singh, 2004]. Using approximately 50 templates consisting of ordered lexical terms and part-of-speech constrained variables (usually called lexico-syntactic templates) and tuned to the context in which the corpus sentences were acquired, they extracted 20 different relation types including spatial, temporal, causal, affective, linguistic, and taxonomic. These relations

were combined in a Quillian-style semantic network that has been used in a wide variety of inferential reasoning tasks. The system provides interesting performance on a variety of traditionally difficult tasks [Lieberman et al., 2004] [Liu et al., 2003]. Another project [Oertel and Amir, 2005] used ConceptNet as a filter for choosing rules from a more formal knowledge base whenever the world state was unfamiliar to an agent. The names of the rules, which have meaning to users, were mapped to nodes in ConceptNet, allowing words associated with world state to prioritize rules.

The target representation for ConceptMiner is the existing set of ConceptNet relations, with the end goal of dramatically extending the breadth of the knowledge base while only modestly impacting the quality of the resulting inferences.

1.2.3 In-situ Commonsense Acquisition

The aforementioned applications are great examples of 'soft measure' and 'natural mistake' paradigms. Another application, Aria [Liu, 2002], enables users to retrieve pictures from a tagged database of images for inclusion in a piece of e-mail. Moreoever, when a user pulled an image from the database and dropped it into the e-mail, the program automatically back-annotated the retrieved picture with key phrases from the surrounding language context as well as propogating such annotations to related pictures.

This example illustrates a general strategy of augmenting knowledge in Commonsense-based systems by exploiting programmer knowledge about the semantics of particular user interface interactions and how those semantics relate to the natural language content of the application [2].

A second approach to online acquisition is simply to ask users to explain a relationship that the computer does not have. Breaking the flow of attention by interrupting users is challenging, but if it resolves a specifically identified mistake in a way that empowers the user to improve the behavior of the interface rather than frustrating

[2]Natural Language content can be the data manipulated by the application, such as an e-mail, but it can also be natural language annotations placed on program data such as Flickr tags on photographs

them with continued incorrectness, such an approach may be highly effective[3].

1.3 Limitations of Ambiguous Knowledge

The ability of the ConceptNet inference algorithm to accommodate noise arises from a preponderance of evidence relevant to the query. If there is sufficient knowledge in the database relevant to the target context, spurious entries and polysemous meanings of constitutent terms are cancelled out by the superposition property of spreading activation. Intuitively this is similar to the thin-slicing phenomenon popularized by Malcom Gladwell's *Blink* [Gladwell, 2005].

ConceptNet provides a set of semantic associations that are comparable to the associations a human might have if they had to list the phrases that are most related to one-another [Liu and Singh, 2004]. This association property immediately affords the desired natural mistake property. However, any time a query is made without sufficient reinforcing knowledge, or without sufficient knowledge in the knowledge base, the system fails in the worst way: by providing patently incorrect answers. This limits the existing applications to those where failure does not have a negative impact on the user over the same application without Commonsense. What is necessary to move beyond the limitations exhibited by ConceptNet today?

1.3.1 Lack of Breadth

The first challenge for reasoning directly with the knowledge soup is that the system will need knowledge for almost every concept it is likely to encounter. With ConceptNet's emphasis on natural language phrases instead of words there is a potential universe of phrases consisting of common verbs, nouns, and descriptive words on the order of $N_{phrases} = (Adverbs * Verbs * Adjectives * Nouns)$ where each grammatical class will range between ten and fifty thousand in the English language alone. This puts the possible node count of concepts in ConceptNet at sixteen orders of

[3]Such an approach may be especially powerful if the contributions of many different users of an application can be aggregated and redistributed

magnitude, or somewhere between a quadrillion and a quintillion possible nodes.

Not all combinations of words represent Commonsense concepts; take the phrase "flagrantly screaming torpid stones" as an example, this phrase is syntactically valid and to some readers may even seem poetic, it is definitely not an example of Commonsense. While one can take the above numbers as an inflated upper bound, they illustrate the sheer scope of the acquisition challenge. Moreover, we are looking for pairwise relations among these phrases, so the upper bound on edges is $N_{edges} = {N_{phrases}}^2 * N_{reltypes}$ where $N_{reltypes}$ is the number of different relation types[4].

In humans it is expected that the meaning of a phrase of words is not stored independently, but that meaning is established by the combinatorial properties of language and word meaning. "rabid duck" is a very similar concept to "rabid chicken," but more distant from "rabid dog," which has a richer set of social associations in the English speaking culture.

Moreoever, many phrases are effectively paraphrases of a common concept. ConceptNet accommodates some of this combinatorial and synonymous aspect by providing hierarchical links between nodes that share terms, or have known dominance or class/instance relationships. The system does not demand a perfect hierarchy, as that precludes the error-tolerance necessary for volunteer-driven or web-mining models of acquiring knowledge.

Making a concerted attack on this universe of concepts is likely to require a level of scale far above the three hundred thousand concepts embodied in the current ConceptNet database. For this level of acquisition, I believe Commonsense researchers will need to turn to automated techniques that are either in-situ, or offline.

1.3.2 Lack of Depth

While there is a tremendous amount of knowledge in ConceptNet, it is limited to binary relationships of a very general nature. The system may know, for instance, that an **EffectOf** "eat cookie" might be "get fat," but it may not have a causal model that justifies that knowledge. We can use subevent relations to know that a part of

[4]ConceptNet 2.2 has 20 relation types

the act of "eat cookie" is "pick up cookie" and "put cookie in mouth," but it is very difficult to reason precisely with these kinds of concepts in ConceptNet.

The OpenMind group proposed another resource, StoryNet[5], consisting of a sequence of events that are grouped together describing common actions in space or time. Indeed representations like this can produce much more precise and relevant fodder for automated reasoning [Riesbeck and Schank, 1998] [Singh, 2005]. However the notion of reasoning with scripts based on imprecise knowledge is not well developed, so addressing the issue of depth may be premature. Regardless, changing the representation and reasoning mechanism itself lies beyond the intended scope of this thesis.

1.3.3 Accuracy and Robustness

Ultimately the current knowledge base itself is limited in that there is only the notion of relatedness and no way to enable precise reasoning. Liu's Bubble Lexicon [Liu, 2003] was a proposal to deal with some of the immediate representational issues of ConceptNet, namely word polysemy, more accurate link weights, and the development of new links.

Regardless of the enhancements, the current representation and inference techniques are fundamentally limited in their precision. It is clear that significantly enhancing the current abilities of ConceptNet will require external knowledge about when and how ConceptNet is relevant to a particular set of problems [Minsky, 2006].

1.4 Searching for Commonsense

The project described this thesis, ConceptMiner, was initiated to address the first of the aforementioned challenges: dramatically increasing the breadth of relational knowledge in order to bring current applications from prototype examples into the realm of the practical. It is also intended to enable the acquisition of domain-specific Commonsense relevant to specific applications.

[5]Inspired by [Minsky, 1975] and [Schank and Abelson, 1977]

The strategy taken is to leverage the existing corpus of Commonsense knowledge to constraint a search of the web to specific contexts syntactic, semantic or topical, in which the Commonsense we have is expressed. The context can be something simple such as:

```
DesireOf: "... does/VBZ a/DT  ____/NN crave/VBP ____/NN ?/."
DesireOf: "... just/RB a/DT ____/NN demanding/VBG ____/NN ./."
DesireOf: "... believe/VB ____/NN deserve/VBP ____/NN is that ..."
```

which here is expressed as a lexico-syntactic template expressing the DesiresOf relationship between two nouns. In this case "professional" and "respect" would satisfy the template, as would "dog" and "food," although the latter is less likely to occur in normal usage.

This task of identifying dependent contexts is the central subject of research for the Information Extraction community, which traditionally has focused on facts or linguistic knowledge. However, the community has recently published several explorations into acquiring general relational knowledge and Commonsense knowledge as discussed in Chapter 3.

The critera for success in extracting Commonsense relations for the ConceptNet toolkit are:

1. The system can collect knowledge for many or all ConceptNet relation types
2. Collected knowledge is of a similar quality to that extant in ConceptNet
3. The knowledge collected can identify concepts, particularly multi-word concepts, not already known to the ConceptNet knowledge database
4. The inference of ConceptNet improves instead of degrades in the presence of the collected knowledge

Any progress on the first point above is an advancement in the search for Commonsense knowledge, but alone is not enough to make ConceptNet more broadly applicable. Therefore, all four points will be addressed in the evaluation offered in

Chapter 5. The next chapter will offer a model of the system in the thesis by exploring an extended example of how knowledge works in ConceptNet, how we can train a system to find new concepts, what new data can be learned, and how that affects ConceptNet's inference process.

Chapter 2

Extended Example

The primary uses of ConceptNet in applications to date have been centered on a single inference algorithm called **get-context**, which accepts as input a set of natural language phrases and as output provides an ordered list of phrases that are contextually related to the input set. The algorithm performs spreading activation [Liu and Singh, 2004] over a directed graph consisting of concepts (nodes) and relations (links). The result set is determined by the most related set of nodes according to a function of the initial weights, weights within the graph and a link-type weight dictionary. At each step, node weights are computed by a non-linear combination of weights propogated from all upstream nodes. The result list can be modified by adjusting the set of dictionary weights for different relation types.

The purpose of employing spreading activation is to look for multiple sources of evidence, represented as multiple paths through the graph, for determining an association. Concepts that are more richly connected will recieve higher weighing. Low frequency spurious concepts that would be interpreted as incorrect or even as garbage data are ranked lower by this algorithm allowing the system to operate even in the presence of significant noise. In [Singh, 2002] human judges determined that 75% of the items were true, indicating the the existing success of ConceptNet was achieved with as much as 25% incorrect knowledge.

The top results for **get-context** from ConceptNet 2.2 for the input set "dog," "park," and "catch" after filtering the input terms is:

```
run after ball
go fishing
fishing
something
frisbee
chat with friend
move car
drive car
```

More specific uses of **get-context** include inferences such as projection, where only a subset of relations are used. Using a dictionary consisting only of relations like EffectOf, DesirousEffectOf, SubEventOf, etc will make causal or temporal predictions about possible consequences of an event. The projections of this same input set add new terms to the top 10:

```
Spatial:        mailbox, football, helmet, flag, letter

Affective:      wake up in morning, go for run, laugh, use television,
                surprise person

Consequences:   baseball, school bus
```

While adjusting weights can change the results set distribution, the essence of ConceptNet is the knowledge implicit in the connectivity of the graph. By enriching one concept's links to another cluster of concepts, we can influence the results for **get-context** much more directly.

The database underlying the ConceptNet graph is best visualized as a set of relations of the form $(RELATION\ CONCEPT => CONCEPT)$.

For instance,

```
(SubeventOf ''drink milk'' ''swallow liquid'')
```

captures the intuition that one event[1], "swallow liquid," is part of the larger event "drink milk."

The goal laid out in the prior chapter is to help align the results from ConceptNet inferences with the expectations of human application users. To do this it is necessary to identify new relations that are considered generally true by human judges. Section 3 will discuss the nature of this knowledge in more detail. The remainder of this chapter illustrates the information flow in ConceptMiner. A more in-depth, technical review of ConceptMiner is provided in Chapter 4.

2.1 Using Knowledge to Find Knowledge

The task of finding new knowledge is potentially simplified by the existence of prior knowledge. ConceptNet comes with over one million relations, several hundred thousand of which are rich semantic relations such as MotivationOf, DesireOf, EffectOf, CapableOf, etc.

Like many information retrieval system, there are two phases: training and acquisition. The system's training phase begins by sampling relation instances[2]:

```
(DesireOf "dog" "bark")
(desireof "dog" "attention")
(desireof "dog" "food")
(desireof "dog" "be walk")
(desireof "dog" "not starve")
...
```

[1]In the concept representation in ConceptNet most concepts are stored in a canonical form implying a particular type of concept, specifically: NN = Object or Location, VB = action, and VB + NN = event.

[2]I collected examples of instances of between 500 and 1000 relation instances for each of the three relation types explored in this thesis: DesireOf, EffectOf and CapableOf.

2.2 Querying the Net

Now we desire to find a general textual context within which we might find more such relations. A query entered to google of the form[3]: 'allintext: "dog * bark"' yields the following top 5 results (after tagging).

1. "My/PRP$ dog/NN loves/VBZ attention/NN ./."
2. "Horseback/NN riding/VBG dog/NN attracts/VBZ attention/NN ."
3. "With/IN the/DT dog/NN paying/VBG attention/NN to/TO you/PRP ,/, ..."
4. "If/IN it/PRP '/POS s/PRP a/DT long-haired/JJ dog/NN ,/, pay/VB attention/NN to/TO how/WRB the/DT hair/NN falls/VBZ ;/:"
5. "When/WRB out/IN with/IN your/JJ dog/NN ,/, pay/VB attention/NN to/TO your/JJ surroundings/NNS and/CC ..."

Only the first of these examples directly expresses the DesiresOf relation. The second sentence is neutral as it may express the relation of interest because an entity may act to achieve a consequent that "attracts," but they may attract unwanted attention as well. Numbers three through five are bad patterns as they rely on the trope "pay attention," which does not express a clear desire or potential desire of a dog. Thus some patterns will be more precise than others and it is certain that all patterns will generate examples that are not examples of the class.

2.3 Extracting General Patterns

However, if we assume that we have many such examples, it is easy to see how generalizing over the many possible contexts yield some very simple and general patterns of the form:

[3]To get accurate tagging the system needs full sentences and these rarely show up in links or titles, thus the allintext tag.

1. <X>/NN loves/VBZ <Y>/NN
2. <X>/NN attracts/VBZ <Y>/NN
3. <X>/NN paying/VBG <Y>/NN
4. <X>/NN ,/, pay/VB X/NN to/TO
5. <X>/NN ,/, pay/VB X/NN to/TO

To generate the pattern set for DesireOf I collected several tens of thousands of such examples over many different relations. The patterns are then ordered by $Q * log(T)$ where Q is the number of unique patterns and T is the total number of instances the pattern matched.

The top 5 patterns in the pattern set after sorting by this metric are:

```
f (OR */NN */NNP) might/MD (OR (*/VB */TO */NN) (*/RB */NN)
                                (*/VB */NNP) */NN */NNP */VB)
f (OR */NN */NNP) '/POS s/PRP (OR */NNP */JJ */NN */V)
f (OR */NN */NNP) s/PRP (OR (*/NN */IN */NNP) (*/JJ */NNP */NN)
                            (*/JJ */NNP)(*/JJ */NN)
                            */VB */JJ */NNP */NNS */NN)
b (OR */V */NNP */NN) (OR */NN */NNS */V */NNP) and/CC
b (OR */NNP */NN) for/IN (OR (*/JJ */NN) */NN */NNP)
```

The format was adjusted here to paraphrase the actual system's representation of an extraction pattern. The first character is 'f' to indicate that the variables represent a forward directed relationship between extracted terms and 'b' the inverse.

To avoid extracting too many false instances, the system requires that the pattern source and target concepts match one of the part of speech templates of the original queries. The first pattern is of the form "X/* might/MD Y/*" where X or Y can match one or more terms with the appropriate tag or sequence of tags.

To provide an intuition as to why these simple patterns work, we can see that the sentence "the dog might bark" would cause the system to propose that "dog" and "bark" participates in the DesireOf relation. Of course, knowing that someone might do something is not sufficient cause to conclude that they want to do it. It is,

however, evidence towards that conclusion. The second pattern matches "I couldn't grab the dog's bone." indicates that a dog might desire a bone. If an agent posesses an object, it is reasonable to postulate they might want the object.

A listing of all the top 50 discovered patterns for three relations in the current version of the miner can be found in Appendix A.

2.4 Finding New Instances

Given a set of patterns that express, with some probability, that two instances are part of a relation, the next step is to formulate search queries that can find new instances of that pattern. Search queries are generated from a triple containing a pattern, an instance, and a location.

To perform a full query the system searches for 50-200 pattern instances, extracting between 20 and 200 URL's per pattern. Often, multiple instances are retrieved from a single page as the pattern can match more than the sentence referred to by the search engine. Retriving and parsing 1000 to 40,000 pages per concept is the main bottleneck for the system; by comparison matching patterns to text is relatively fast. Searching, fetching, and tagging HTML text can be done at a rate of 1 to 4 pages per second. A short mining pass for a single concept can take a few minutes; a long one can run up to several hours, depending on internet conditions.

A query for the concept "dog," with the first pattern above, results in the search query containing 'allintext: "dog might *." After extracting URLs from the search results, and fetching and tagging the referred-to pages, the result patterns, filtered for duplicates, will look like the following list:

1. "How to tell when a dog might bite." => (DesireOf "dog" "bite")

2. "Although an active, bouncy dog might catch your eye..." => (DesireOf "dog" "catch your eye")

3. "... or other objects a puppy or dog might chew." => (DesireOf "dog" "chew")

4. "The dog might suffer." => (DesireOf "dog" "suffer")

5. "antifreeze, a substance that a roaming dog might encounter ." => (DesireOf "dog" "encounter")

Not surprisingly, some of the contexts illustrate the same classes of problems identified in the pattern extraction examples above. All of the proposed relations emerge from sentences that discuss actions or events that a dog might desire, but none are definitive.

2.5 Filtering and Selecting Instances

Given a set of instances, it is required that we separate instances of the relation from non-instances or terms that are related but perhaps belong in a different semantic category. Dog is a good example because it is a particularly difficult source concept due of the sheer popularity of dogs leading to nicknames "What up dog?," use of dog as a pronoun "he's the big dog," and other such borrowed usages. Moreover popular topics often appear on blogs and dog fan sites with poorly structured English and other such impairments. This results in many types of false positives which we can examine by exploring the following lists of relations from ConceptNet and discovered by ConceptMiner miner.

ConceptNet Relations for "dog":

```
(desireof "dog" "not hear loud noise")
(desireof "dog" "not hear")
(desireof "dog" "not be shoo")
(desireof "dog" "not be hit")
(desireof "dog" "praise from person ' s master")
(desireof "dog" "lot")
(desireof "dog" "eat")
(desireof "dog" "punishment")
(desireof "dog" "not be leave")
```

```
(desireof "dog" "be walk day")
(desireof "dog" "lot of attention")
(desireof "dog" "come inside during winter")
(desireof "dog" "not be shoo off couch")
(desireof "dog" "eat lot")
(desireof "dog" "not hear noise")
(desireof "dog" "attention")
(desireof "dog" "food")
(desireof "dog" "be walk")
(desireof "dog" "not starve")
(desireof "dog" "not be abandon")
(desireof "dog" "not be hit with newspaper")
(desireof "dog" "be love")
(desireof "dog" "be love by person ' s master")
(desireof "dog" "praise")
(desireof "dog" "affection")
(desireof "dog" "flea")
(desireof "dog" "be pet")
(desireof "dog" "bone")
```

Partial list of relations discovered by ConceptMiner:

```
(desireof "dog" "arrive")       (desireof "dog" "shelter")
(desireof "dog" "define")       (desireof "dog" "swim")
(desireof "dog" "establish")    (desireof "dog" "visitor")
(desireof "dog" "lawn")         (desireof "dog" "adopt")
(desireof "dog" "remind")       (desireof "dog" "knock")
(desireof "dog" "thailand")     (desireof "dog" "pit")
(desireof "dog" "succeed")      (desireof "dog" "settle")
(desireof "dog" "bark")         (desireof "dog" "resist")
(desireof "dog" "convince")     (desireof "dog" "snap")
```

```
(desireof "dog" "locate")       (desireof "dog" "yoga")
(desireof "dog" "kennel")       (desireof "dog" "tender")
(desireof "dog" "make sense")   (desireof "dog" "obedience")
(desireof "dog" "counsel")      (desireof "dog" "remedy")
(desireof "dog" "lick")         (desireof "dog" "distinguish")
(desireof "dog" "chew")         (desireof "dog" "accommodate")
(desireof "dog" "behave")       (desireof "dog" "navigate")
(desireof "dog" "attach")       (desireof "dog" "stitch")
(desireof "dog" "interfere")    (desireof "dog" "fetch")
(desireof "dog" "dwarf")        (desireof "dog" "playhouse")
(desireof "dog" "cocoa")        (desireof "dog" "diazepam")
(desireof "dog" "abide")        (desireof "dog" "undertake")
(desireof "dog" "sniff")        (desireof "dog" "outsider")
(desireof "dog" "resign")       (desireof "dog" "posture")
(desireof "dog" "roam")         (desireof "dog" "suppress")
(desireof "dog" "inspect")      (desireof "dog" "archie")
(desireof "dog" "distract")     (desireof "dog" "splinter")
(desireof "dog" "carve")        (desireof "dog" "injure")
(desireof "dog" "deprive")      (desireof "dog" "waterfowl")
(desireof "dog" "mckinley")     (desireof "dog" "groundhog")
(desireof "dog" "new puppy")    (desireof "dog" "socialize")
(desireof "dog" "subdue")       (desireof "dog" "depress")
(desireof "dog" "play jazz")    (desireof "dog" "herd sheep")
(desireof "dog" "avoid dog")    (desireof "dog" "salivate")
(desireof "dog" "heel")         (desireof "dog" "foul")
(desireof "dog" "positive reinforcement")
```

2.5.1 Classes of Data

This list of 69 examples can be broken down into the following categories:

- 17 Positive relations, such as (desireof "dog" "sniff")
- 6 Type mismatches, such as (desireof "dog" "salivate"), which should be (capableof "dog" "salivate")
- 17 Conditional relations, such as (desireof "dog" "playhouse"), which could be true in some circumstance, but are not generally true of dogs as a class.
- 9 Scope mismatch such as (desireof "dog" "abide") which convey little information due to being far too general or unpredicated. Dogs usually "abide" at home or in a doghouse, but one wouldn't associate dog with that concept.
- 1 Inverse relations such as (desireof "dog" "depress"). Dogs can be depressed, but they probably desire to "not be depressed" instead.
- 18 Garbage relations such as (desireof "dog" "diazepam") which are clearly illogical

The goal of the system is to return a high percentage of positive results. Each of the non-positive classes must be filtered by the system.

2.5.2 Filtering Instances

The instances are next run through a set of filters. For the current example the filters described emphasize enriching the connectivity of existing ConceptNet concepts rather than learning new concepts. Filters include:

- Well-formed phrases - This filters out content that has non-ascii characters, numbers or are too long.
- Concept filter - remaining terms are matched against the existing ConceptNet and removed if they are not known terms. This removes significant garbage knowledge and was applied to the example set above.
- Inferential distance - When filtering only concepts, we can further rank concepts by their distance within the existing network.

- PMI filter - Pointwise Mutual Information is a common information extraction measurement used to capture the co-occurance relationship between two terms in a corpus. This proves to be very effective in removing concepts characterized as Conditional, Scope and Inversion data types.

- Pattern Frequency - Further filtering can be performed by taking the remaining concepts and running a set of search queries to get instance counts for some number of top patterns. This is an expensive operation, but is used to separate positive from negative instances when no other filtering criteria dominates.

2.5.3 After Filtering

The system developed for this thesis does not solve all the problems for difficult nodes like "dog," but the resulting knowledge is much cleaner than raw information from the web. The cost of filtering is losing many positive instances and there is a great deal of opportunity for exploring ways of segmenting the positive from negative without so much attrition of positive examples.

Applying the filtering process to the dog examples above results in the following list.

```
(desireof "dog" "lawn")          ;; Conditional
(desireof "dog" "swim")          ;; Positive
(desireof "dog" "bark")          ;; Positive
(desireof "dog" "locate")        ;; Positive
(desireof "dog" "succeed")       ;; Positive/Scope
(desireof "dog" "remind")        ;; Conditional/Scope
(desireof "dog" "resist")        ;; Conditional/Scope
(desireof "dog" "kennel")        ;; Conditional
(desireof "dog" "shelter")       ;; Scope
(desireof "dog" "arrive")        ;; Scope
(desireof "dog" "adopt")         ;; Type
(desireof "dog" "establish")     ;; Scope
```

The threshold for inference length was chosen for this example to illustrate that a separator exists that eliminates negative examples while retaining 40% of the original positive examples.

Only four of these concept pairs existed in the original ConceptNet corpus and none were part of the DesireOf relation. The results for other concepts and relation types is summarized in Chapter 5.

2.6 Augmenting ConceptNet

Once a final set of relations have been chosen, they can be fed back into ConceptNet directly to augment the existing graph structure.

Chapter 3

Theory and Rationale

The goal of the thesis is to leverage existing language content to identify general, Commonsense relationships. This thesis makes several assumptions about where such knowledge can be found, how to separate good knowledge from bad, and how to evaluate the effectiveness of the process. This chapter reviews the relevant work and introduces the specific hypotheses I wished to test and the rationale behind the expectation of the outcome. Chapter 4 presents the specific methodology and system used to test these hypotheses and Chapter 5 summarizes the results.

3.1 What is Commonsense?

The relational representation of ConceptNet allows any English phrase to be entered. Thus, we have to define what terms are appropriate to put into relation to augment ConceptNet's domain-general Commonsense. The following examples were discussed in [Vanderwende, 2005]:

- (CapableOf "Birds" "fly")
- (NotCapableOf "Penguin" "fly")
- (CapableOf "Flipper" "swim")

Number one above is a commonly valid assumption. If one sees a bird, one can assume without other evidence that it is capable of flying. However, as the second

entry implies, the Penguin is a bird that can't fly. The second sentence is *definitional* commonsense that is universally true, whereas the first clause is *defeasible*, meaning that it can be overridden by valid exceptions. The third example we can describe as *instance* knowledge. It may be defeasible or definitional, but typically refers to instances of classes or proper nouns, in this case "Flipper" the TV dolphin.

This illustrates one of the differences between the targets of Commonsense acquisition and those of Fact Extraction or Question Answering - the aim is to identify general definitional or defeasible information about classes of entities and actions rather than facts about specific instances of those class. For the purpose of this thesis, I am interested in extracting both defeasible and definitional knowledge, and will refer to both as Commonsense knowledge. Instance knowledge is something that can be very useful, but in the search for Commonsense, is to be avoided as much as possible.

3.1.1 Does Commonsense exist in ordinary text?

The key assumption behind the OpenMind is that "such knowledge is typically omitted from social communications, such as text" [Singh, 2002] (p. 211). However, there are quite a number of surface forms that can express either defeasible or definitional knowledge. A set of such forms is cataloged by [Vanderwende, 2005]:

- Complements of factive verbs (i.e. "knew that")
- Temporal adverbials (i.e. "when X/NN was Y/NN")
- Non-restrictive relative clauses (i.e. "X, which are Y")

That study indicated that Commonsense is often explicitely stated in order to set up a contrast or in text that is intended to teach. However the syntactic forms introduced above tend not to contain the most general common sense knowledge.

3.2 Commonsense Relation Discovery

A great deal of literature has been dedicated to the automated acquisition of syntactic and semantic information from natural language text. I will emphasize work from a subfield called Information Extraction (IE) concerned primarily with extracting facts, questions, and other information from natural language text.

There are a number of sub areas of particular interest; specifically those interested in constructing "IsA" ontologies ([Etzioni et al., 2004], [Hearst, 1992], [Geleijnse and Korst, 2006], [Pantel and Pennacchiotti, 2006]), identifying linguistic relationships ([Hearst, 1998], [Montemagni and Vanderwende, 1992], [Riloff and Jones, 1999], [Chklovski and Pantel, 2004]), and extracting facts from the Web([Etzioni et al., 2004], [Pantel et al., 2004], [Craven et al., 2000]).

Several researchers have attempted to use information extraction techniques to specifically identify common sense knowledge within text, most notably [Vanderwende, 2005] and [Schubert and Tong, 2003].

Vanderwende describes an experiment to extract causal relations from a dictionary using the hand-built linguistic templates described above (factives, temporal adverbials, and non-restrictive relative clauses). The results were judged in comparison to the knowledge in the OpenMind knowledge base and found to have more garbage information (26% collected vs. 19% OMCS), but similar quality for the remaining data except for generality, where the contributions were judged to be more specific than in OMCS. The author speculated that a less specific corpus might yield less garbage and information with more generality. This study also motivated the Pointwise Mutual Information (PMI) technique used here to separate common sense truths from truths that were more dependant on context. The argument is that the PMI score for a concept A and a common sense concept B will form a higher total number of hits than a conditional concept C due to C being less frequent in a Web-sized corpus than the more general concept B. (For more discussion, see sections 4.5.5 and 5.3.4).

A similar template-based approach was taken by [Schubert and Tong, 2003] but using more general lexical-semantic templates with the intention of acquiring all the

upper ontology information available. A significant fraction of knowledge collected by this system is of too general a nature, such as "A female person can want something," but nevertheless underscores both the existence of quality information on the Web and the difficulty of extracting that knowledge. The system also depends on hand-built templates as well as a hand-built ontology to convert proper nouns and other terms into category labels to more directly capture semantic information.

The Know-It-All system by [Etzioni et al., 2004] and the Espresso system by [Pantel and Pennacchiotti, 2006] bear the most similarity to the system described in this thesis and are both intended to extract general classes of semantic relations from open text using variations on pattern-based extraction techniques. The relationship between ConceptMiner and these systems is discussed below.

3.3 Acquisition Methodology

To the four systems and experiments described above, this work adds a very important constraint: the availability of a large class of instances for broadly defined commonsense relations. The ConceptNet corpus consists of thousands of instance pairs for relations such as DesireOf (an agent generally is known to be motivated to perform an action, posess an object or be in a state) and EffectOf (the consequence of actions or events) and tens of thousands of pairs of more general classes of relationships including operative, temporal, spatial, and affective.

3.3.1 Pattern Based Approaches

Almost all of the approaches described use hand-built or automatically discovered lexical-syntactic patterns. This approach was first reported by Hearst [Hearst, 1992] in 1992 with limited success. These patterns were expressed as sequences of word tokens coupled with a part-of-speech identifier (paraphrased from [Riloff and Jones, 1999] and [Pantel et al., 2004] respectively):

```
Lexical: "offices in <x>" => IsA(<x>, location)
```

```
Lexico-syntactic: "<x>/NNP is/* a/* */* metal/*" => IsA(<x>, metal)
```

Wildcard entries allow a word to be accepted if they have a particular part of speech, or any word is allowable regardless of part of speech. Usually captured as a simple form of regular expression, patterns can match sequences in part-of-speech tagged text to identify specific word phrases. Because the human understands the semantics of the templates, they can guarantee that the instantiation of any two concepts in the pattern is strongly correlated to the presence of those instances in a specific relation.

In general, no single pattern is sufficient to identify a Commonsense relation with high confidence. There are two primary approaches to filtering prospects. The first relies on multiple patterns expressing a given relation (e.g. [Hearst, 1992]) to raise confidence over any any single pattern. The second employs a separate validation step after acquisition of prospective instances.

KnowItAll [Etzioni et al., 2004] uses a web hit-count based Pointwise Mutual Information (PMI) score similar to that discussed by Vanderwende. Web PMI is determined from the ratio of an assessment phrase co-occuring with the instance knowledge over the instance phrase occuring independently. This technique works more effectively with class information ("the city of Boston" vs. "Boston") than with instance information, but may be an additional step worth exploring to separate good relation instances from bad.

Espresso [Pantel and Pennacchiotti, 2006] uses a much more complex scoring algorithm that recursively estimates quality of patterns and instances by concurrently exploiting PMI among pattern and instances, retrieved pattern-instance frequency and the clustering effects of mutual bootstrapping (discussed below). The Espresso PMI calculation is prohibitively expensive for ConceptMiner due to the numerous per-instance hit count queries. Espresso avoids this problem by exploiting and mining local corpora (TREC-9 and CHEM) for relations and using the Web only for query expansion and not for PMI scores.

The relations used in Espresso are also more specific, lending themselves to more specific extraction patterns than ConceptMiner. Pantel uses generic patterns for Web

query expansion whereas ConceptMiner uses a large set of generic patterns coupled with a more aggressive instance filter.

The general goal for all these systems, is to take a set of instances of a given relation, or patterns expressing that relation, and train a general algorithm to find new instances of the relation.

3.3.2 Mutual Bootstrapping

In 1999, Riloff, Jones, et al. [Riloff and Jones, 1999] introduced a technique called "Mutual Bootstrapping." Mutual Bootstrapping identifies a way to construct a semantic lexicon without having to develop extraction patterns by hand. Over the past five years, a number of researchers have extended bootstrapping techniques to address issues of scale when using the Web as a corpus for information extraction. There are two recent systems in this tradition that directly motivate the work in this thesis.

Etzioni et al.'s "KnowItAll" [Etzioni et al., 2004] is one of the better known extraction systems to fully accommodate the varying quality of text on the Web. The system splits the extraction process into two key components: generic extraction patterns and assessment phrases. Generic patterns are hand-generated lexical-syntactic templates that express the exact desired relationship between two words or phrases. Assessment phrases are used to confirm the correctness of a suspected relation. The assessment phrases are used in a point-wise mutual information calculation to assess the probability that a relation is valid given Web search engine statistics (e.g., Google or Yahoo) of the bare word as compared to the word instantiated in the assessment phrase. The first study of this system emphasized facts about states, countries, actors and films – again, a more restrictive set of relationships than targeted here.

Chklovski and Pantel's VerbOcean [Chklovski and Pantel, 2004] utilize a similar strategy, but instead focus on relationships between verb pairs, specifically: similarity, strength, antonymy, entailment and happens-before. This work provides another example of the use of generic extraction patterns and assessment phrases and illustrates the prospect that this general approach can accommodate a wide variety of relational knowledge as opposed to simple class-instance groups.

Espresso's scoring methodology is similar to bootstrapping, but instead of iteratively choosing top instances on each iteration, it rescores both patterns and instances, dropping those that score too low and adding a single, high-scoring pattern each time.

3.4 Evaluating Extracted Data

The quality of the described research systems is typically measured by computing **precision** and **recall**.

3.4.1 Recall Measurements

Extracting content from the Web raises challenges for recall (number of instances retrieved / number of instances present), as there is no sound method for determining the total number of available instances. Every researcher deals with this differently by computing an estimate and performing an empirical study to validate the procedure for computing the estimate. In this thesis recall is estimated only as a relative recall of proposed instances before and after filtering. This assesses the cost of the filtering stages, but does not characterize adequately the absolute recall of the system. A human provided corpus of web pages known to contain the desired information could be used as a golden reference to assess absolute recall in future studies.

3.4.2 Precision Measurements

Precision is handled by using subject human assessment to rate a sampled subset of the data for accuracy. Given the low number of samples presented in Chapter 5, the accuracy of precision estimates is less important than demonstrating an improvement in the ratio of good data to bad. Even if the system has to accept low total recall to maintain a reasonable noise ceiling, it would prove capable of successfully augmenting ConceptNet,

3.4.3 Reference Systems

Previous systems provides a reference for what could ultimately be expected from ConceptMiner in terms of recall and precision. In KnowItAll, the extraction phase alone yielded IsA class precision between 0.35 and 0.96, but after the assessment phase the results were between 0.79 and 0.98 with only a modest reduction in recall (1.0 to 0.85 on average). VerbOcean's precision was less favorable, at 67.5% averaged across the various relations.

Espresso provides the most interesting example of large-scale attempts to build a general system for extracting relations. Reported data incloudes is-a (precision of 73%/TREC and 85%/CHEM), part-of (60%/TREC and 80%/CHEM), domain-specific ("production" and "reaction" in CHEM at 72% and 91% respectively) and succession[1] (50%/TREC).

These techniques are promising for the ConceptNet precision goal of 75+%. However, the Commonsense relations being targeted are closer to Espresso's succession relation indicating that new techniques will be needed in order to yield the desired precision.

3.4.4 Application Evaluation

When a sufficient critical mass of instances is available over several relation types, an important assessment step will be to determine whether the acquired data has a positive subjective effect on ConceptNet inferences.

3.5 Challenges and Opportunities in ConceptNet

Targeting the ConceptNet representation requires extending prior work in the following ways:

1. Explore the applicability of IE techniques to broad semantic relation types such as EffectOf, SubeventOf, DesireOf, and MotivationOf.

[1] A title, like CEO that can move from one entity to another: "George Bush succeeds Bill Clinton".

2. Accommodate full preposition phrases, compound nouns, and event structures instead of singular nouns, verbs or noun phrases (Espresso is another system to accommodate rich phrases)

3. Exploit the additional degrees of freedom afforded by the ConceptNet spreading activation algorithm. Some wrong data is more damaging to the algorithm than others.

The relations of ConceptNet are general semantic relations that have not necessarily been heavily lexicalized within the English language. As such there may be less locality in the syntactic patterns that express them. This implies that the there will either be many specific patterns or a set of very general patterns, each of which provides some evidence but none of which are determinative.

Bootstrapping often makes up for a lack of clear ability to compute either the recall or precision of a pattern and is an obvious extension to the current system. However, general relations are likely to require significantly more data to converge on the desired recall/precision metrics. Therefore the thesis emphasizes extraction of relation instances between pairs of concepts already existing in ConceptNet.

3.5.1 Phrase Extraction

The issue of extracting compound nouns has also been a prior subject of study. In [Rosario and Hearst, 2001] compound phrases were extracted with high precision. This was reproduced in [Pantel et al., 2004] which used POS patterns in extraction templates 'to identify sequences of POS patterns that are valid phrases in that location'.

Applied to the problem of this thesis, the patterns will need to have boundary information on both sides of the two phrases - or will need to have a regex part-of-speech pattern for each phrase type. "X/* such as Y/*" is an insufficient pattern, as Y might be 1, 2, or 3 words.

Solving this problem requires that the system identify regular expression patterns that allow for the variations among phrases. Phrases in ConceptNet can be

compound nouns, simple noun phrases, a prepositional phrase, or a verb-argument (event) phrase. The procedure for generalizing from a set of similar patterns will need to accommodate these cases.

An immediate concern is one of yield; more complex phrase types require more accurate part-of-speech tagging, which means a higher probability of passing over valid sentences. If directly expressed Commonsense phrases occur only rarely, ostensibly because they are those concepts which are shared and thus go unsaid, then there is a smaller sample space to target as compared to the factual and linguistic oriented systems described in the literature. It was not immediately apparent how far the existing techniques will scale to accommodate these representational changes or what degree of modification will be needed to acquire knowledge with high precision and reasonable recall.

3.5.2 Soft Inference Properties

ConceptNet also provides certain benefits. The ConceptNet algorithm is insensitive to certain types of noise in the data. Therefore precision can be treated in a more nuanced fashion. We can separate the types of errors into six categories, as illustrated in Chapter 2.

1. Positive - instances that are clearly and intuitively part of the relation

2. Negative - instances that are clearly not part of the relation

3. Inversion - Often Negative examples show up where one concept in an instance pair is the antonym of a concept that is in the relation[2]

4. Conditional - instances that are in the relation, but only under specific circumstances. Often these are talked about in text to contrast against the general, default assumptions of the Commonsense knowledge

[2]One possible filter for these errors is if an antonym pair are both proposed, the term with the highest PMI when correlated with a subset of the extraction patterns is likely to be the one chosen.

5. Scope - All entities desire "be" or "to be" in the general case. Having this as a target concept of every agent in the database is redundant and unnecessarily general, but it isn't necessarily harmful as it serves to enrich the context of the query without requiring propogation up the class heirarchy during spreading activation operations

6. Type - instances that illustrate Type errors are those returned for one relation that really should be part of another. Dogs are both capable of barking and like, or desire, to bark. However the effect of a dog is not barking. An EffectOf instance between dog and bark would represent a Type error. Separation of concepts into the appropriate type is important although not explicitely treated by the current system. Again, Type mismatches may skew the algorithm, but remain part of the object's general context and thus do not pollute the contextual locality of the ConceptNet graph structure

We can further characterize precision in relation to categories that help or hurt ConceptNet inferences:

- Good: Positive
- Neutral: Type, Scope and Conditional
- Bad: Negative and Inversions

Thus a successful extraction emphasizes the Good, tries to completely avoid the Bad and tolerates some Neutral, although the preference is to minimize Neutral results. Thus two precision measures are appropriate for ConceptNet, $Good/Total$ and $(Good + Neutral)/Total$. This characteristic helps in the selection of thresholds for various filters where we minimize the Negative and Neutral but we don't minimize Neutral at the expense of the Good.

Chapter 4

Design and Implementation

ConceptMiner was designed to serve as the base for a long-term, high-throughput Web mining project, the construction of which entailed significant complexity. Detailing the architecture of the system falls outside the scope of this thesis. Accordingly, I only include the important features that had a significant effect on the algorithm, performance or quality of result. The purpose of this chapter is primarily to detail the essential information necessary to reproduce the results reported in Chapter 5.

The first section below introduces the basic dataflow within ConceptMiner. The next section addresses linguistic aspects of the problem which are applicable to any text base corpus. The final section details the specific problems that occur when doing this kind of retrieval from general Web content.

4.1 ConceptMiner Benefits and Design

The primary differentiation offered by ConceptMiner is found in:

1. Larger templates with POS constraints on term extraction allows for phrases as opposed to words

2. The system attempts to learn broad, informal relations directly from volunteer contributions capturing the human intuition behind those relations. The system should generalize to any relation so defined.

3. Use of OMCS/ConceptNet corpus and the path length measure as an instance filter

4. Use of a "band-pass" PMI measure to separate commonsense from Conditional knowledge

The goal of ConceptMiner was to make the process of mining relations from the Web simple and robust without significantly sacrificing throughput. A series of processing stages commences with a set of seed relations to identify lexical contexts to train on. Learning patterns or training a classifier results in a pattern object that can match any occurance in a POS-tagged document and extract appropriate instances, including different POS patterns for different phrase lengths.

After patterns are created, they are used to acquire a set of prospective instances. Web queries are generated that find contexts likely to contain a relation, constituents are extracted and the resulting prospective relation is stored. After a large set of instances are extracted, they are ranked, scored and filtered. The results are intended to be re-integrated with ConceptNet.

The overall dataflow for a mining cycle is predominantly feed-forward.

- Select relation instances to train from
- The pattern miner identifies the top 50-200 patterns for that relation
- The instance miner identifies potential instances (typically 100's per concept)
- A post-processing stage filters bad instances
- Instances are stored for later use

Pattern mining and instance mining can be separated, so that a pattern set can be learned for a set of relations allowing all relation types for a given concept to be mined after pattern identification.

4.2 Facing up to the Web

Central to using the Web as a source of information is accommodating artifacts not found in traditional corpus-based natural language work. The Web is fraught with badly structured pages, missing servers, stingy search engines and spoof sites that serve to obfuscate data. Extracting plaintext from HTML can also introduce artifacts that need to be accommodated by the system. The specific pre-processing I employed is described in the following subsections.

File Restrictions

The first set of constraints to satisfy is getting pages with well-formed text. To accomplish this, you specify that only plain text or plain HTML documents are to be returned (some search engines have a type filter, otherwise the servers for specific pages should avoid non-text, non-HTML content). Moreover, some resources indexed on search engines are much larger than you might like to download and search. I limited my documents to a maximum of 100k bytes after the HTML was stripped out.

Blacklisting

For any given type of query, there are often specific sites that contain information that will interfere with what you are trying to collect. I discovered at least two versions of the ConceptNet predicate files indexed by both Google and Altavista that I had to blacklist to avoid parsing and storing ConceptNet in the mining repository. The specific sites (via URL prefix) I had to blacklist to avoid unnecessary noise were:

```
http://www.cs.caltech.edu/~sidd/
http://pedia.media.mit.edu/wiki/
http://www.eturner.net/omcsnetcpp/
http://www.conceptnet.org/
```

Fetching URLs

Accessing servers referred to by search results is rife with problems. The URL server may be down, return an error, return bad data, return the wrong content type, or just lock up the TCP session until it times out. To keep the process moving at a reasonable clip multiple page fetches need to be done in parallel. I found that 15 processes were sufficent to balance out the dataflow in the pipeline as the tagging and extraction processes couldn't handle many more than a few pages per second.

Text from HTML

All Web mining systems face the problems of extracting reasonable text from HTML. It took a simple two step approach to maintain the throughput at a few pages per second. The first step is to simply remove any tags from the document. Tags with argument text are deleted. Tags with text in the body (tables, address tags, font type tags, div tags, etc) are inlined in sequence. This results in a significant amount of garbage text generated from menus, tables and other such tags.

The second step is taken (as described in 4.3) after applying the POS tagger and locating contexts in text. This stage serves to catch many of the artifacts that result from the simple approach to HTML stripping above.

4.2.1 Retrieving Evidence from the Web

The central operation in ConceptMiner is extracting and analyzing Web content. Many operations such as search and HTTP fetching of pages can fail or timeout and involve signifcant latency. ConceptMiner employs a pipeline of processes which run in parallel using asynchronous queues and a data-driven scheduler. The important steps in this process are similar for pattern discovery and relation extraction.

- **Query**: this component takes a relation, performs surface expansions on each of the concepts, and generates all valid surface combinations of the two terms. Queries are passed one at a time to the next stage.

- **Search**: Each query is then submitted to a search engine. In the pattern mining example above, a search engine is queried for 20-200 URLs related to the submitted query. Searching is the typical bottleneck for mining, as search engines limit the number of queries a client can submit in any given day from a given IP or via a research API.

- **URL Filter**: This applies blacklisting for sites exporting the OpenMind corpus or ConceptNet relation lists and also filters potentially problematic URLs by applying a length limit.

- **Fetch**: URLs passing the filter are sent to the fetching component, which runs a dozen parallel worker processes that fetch pages using the provided URL and strip the returned HTML. A page object is created to map the URL to the downloaded content. To save disk storage space, only the stripped text is maintained, as the HTML can be regenerated by re-fetching the page

- **Tag**: The langutils tagger is used to generate a vector-document, an efficient representation of the text content and part-of-speech tags

- **Extract**: At this stage, each run will have a different component depending on what is being extracted. Typically, the original query is used to search the page content for all occurrences of the search terms. In the case of the pattern miner, terms found within a given sized window are extracted as pairs and indexed under the search query. In this way, multiple patterns can be identified for each page.

4.3 Mining with Concepts

Concepts in ConceptNet are stored in a simple canonical form.

Concepts consist of six main phrase types: simple noun phrase, simple verb phrase, event phrase (verb + object), prepositional phrase, complex verb phrase (verb + pp) and other. Other captures garbage phrases that slipped through the extraction

process. Each phrase type has a canonical representation. Verbs are all in the present tense and are the first word in the phrase. The noun is typically left in singular or plural form, as ducks can do things that a duck can't, such as "be flock."

To search for instances of concepts on the Web, the system has to generate surface forms of these sentences and reinsert markers like determiners when appropriate. Some of this canonicalization is already performed by most search engines. All common words such as 'a,' 'an,' and 'the' are ignored and replaced with '*' wildcards indicating that any word can go there.

Finally, personal pronouns are given special treatment. All the objective personal pronouns are mapped to the term "person." All possessive pronouns are mapped to the two token sequence "person 's," which is the form that possessives are put into by the tokenization strategy used to generate ConceptNet and also employed in Langutils.

Surface Form Generation

To generate queries for concepts, the system must first generate surface forms of the concepts. "eat cookie" expands[1] to "eat cookie," "ate cookie," and "eated cookie."[2] "person" becomes "me," "you," "us," "him," "her," "she," "he," and "them."

For each pair of concepts, all surface forms are generated and all combinations of them produced. In the NEAR query case order doesn't matter and so only the "C1 NEAR C2" queries need to be generated. In the case of pattern queries like "C1 * C2" are used. "C2 * C1" is also required to identify patterns that express the relation using the opposite lexical order.

Natural Language Pre-processing

Once a text string, free of HTML, is provided to the tagger process, the text is tokenized (punctuation is separated and each string token is mapped to a unique id). The resulting sequence of token ids is stored in an array and tags are then

[1]The ConceptMiner function **concept-surface-forms** generates these surface forms

[2]The "eated cookie" is caused by noise in the lexicon generated by the default Porter stemming algorithm that backs up the lexicon-driven lookup in the Langutils library.

iteratively assigned to a parallel array. Both of these functions are implemented using the functionality in the Langutils [Eslick and Liu, 2005] library[3]. Substantial pattern matching to identify phrases, including regular expressions, can be performed over this token representation with almost no performance impact.

Extracting Patterns and Instances

Once a document has been tagged, the query concepts are used to identify instance locations within the document. A list of all occurances are generated for each concept and all concepts that are within some finite distance of each other are extracted into a context object. Tagging works best on period delimited sentences. The sentence containing the two query terms is extracted by using period tags as boundary conditions. Thus, a query for "dog * attention" results in pages with textual context such as "My/PRP$ dog/NN loves/VBZ attention/NN ./." Instance pairs for which sentence boundaries cannot be determined within twenty words are discarded.

Bad Contexts

Once the appropriate textual context has been identified, it is then filtered. The simple heuristics I found effective, based on visual inspection of the output, removes enough noise that truly noisy patterns and instances were rare:

- **One Verb Rule:** A given region of context needed to include at least one verb to be accepted. Sentences that were mis-tagged or consisted of noise (such as javascript) were generally discarded by this rule.

- **Noun Repeat Rule:** Regions with more than five noun tags in a row were discarded. The accomodates a wide variety of errors: sites that mimic search results, menu titles extracted from frame-based HTML documents and other such HTML extraction artifacts.

[3]The library itself employs an implementation of the Brill rule-based tagger [Brill, 1990] specialized to the langutils document representation. The rules used are the default Brill rules trained on the Brown Corpus and the Wall Street Corpus

- **No PDF rule:** Tests are performed on returned text becuase due to lack of standards compliance or proper tagging on some sites "%PDF-xxx" can be returned in response to a request for text content.

These simple heuristics capture most of the problems with tagging and act as a catch-all for the kinds of unusable Web content described below.

4.4 Identifying Patterns

Patterns are represented in a simple s-expression based pattern language:

```
Pattern ::= (<Term> ...)
Term    ::= (OR <Term> ...) | (:TERM <Token> <POS>)
Token   ::= * | TID
POS     ::= * | PennTag
TID     ::= lisp-eq-type(fixnum)
PennTag ::= lisp-eq(:NN) | lisp-eq(:VBZ) | ...
*       ::= lisp-eq:'*
```

Where lisp-eq and lisp-eq-type operators mean that a token is a lisp object under #'eq or an instance of a lisp type. The "PennTag" non-terminal consists of symbols representing one of the Penn TreeBank part-of-speech tags.

Human consumable version of this s-expression language is written as follows: "This/DT is/VBZ */* pattern/NN with/* (or some/DT many/*) wildcards/NN."

Patterns are extracted by finding a match to the search query within the lexical terms of a stripped and tagged web document. A simple scanner identifies the query terms within a fixed distance, returns an interval indicating the start and end and another function generates the appropriate pattern object given the interval.

After a set of patterns instance are identified, the system seeks to generalize the pattern and to identify the set of POS tag sequences that are known to be valid for that relation. The limited word count between terms that result from wildcard queries to google allows for a simple unification algorithm:

- Group items with the same middle-terms (e.g., "the cow might moo at" and "A dog might bark at you" are put into a set together because "might" lies between both cow/moo and dog/bark, the query terms)

- Count how many unique query pairs generated the set and calculate a score (see below)

- Seek to make the context more specific:

  - Recalculate unique and total matches by combining patterns sharing terms to the left or right of the query pair

  - Accept new patterns if the derived pattern's new score is within some factor of the original score[4]. The pattern above would unify to "*/X might/MD */Y at/IN".

- Combine all query term POS tag patterns into a giant OR statement.

- Sort according to the score

This is similar, but simpler than the edit-distance used in [Pantel et al., 2004]. One flaw in this technique is that the POS patterns for the two query terms are not linked. Fixing this could potentially reduce some of the noise in the extraction process.

4.4.1 Ranking Patterns

Given a set of patterns, it is desirable to find patterns with reasonable precision (small false positive rate) and reasonable reasonable recall (generate a good number of instances). To achieve this, a method is needed to identify those patterns with the proper balance of each.

[4]The factor chosen for this set of runs was within 1/2 of the original. An absolute threshold is also set at 0.1, chosen empirically to avoid polluting the space with patterns that won't be selected. The search proceeds for all combinations of 1, 2, and 3 words to each side.

Each pattern is assigned a score derived from a metric introduced in AutoSlog-TS [Riloff, 1996]:

$$score(pattern_i) = R_i * log_2(F_i)$$

where F_i is the number of known instance pairs extracted using the $pattern_i$ and $Ri = F_i/N_i$. N_i is the total number of unique instance pairs returned by $pattern_i$. This does not account for precision, but only for patterns that generate more unique instances. Feedback through bootstrapping may improve the quality of the top patterns.

After sorting, the top 50 to 200 patterns are extracted. For the data in this thesis 50 patterns was used to reduce the time taken during instance mining.

4.4.2 Retrieval of Pattern Instances

After patterns are generalized, the instance miner pipeline is invoked and a list of source concepts are combined with generalized patterns to generate surface queries. The process is similar to the pipeline discussed above for mining pattern, except the extraction stage identifies instances using the pattern and records them as instance prospects prior to filtering.

4.5 Filtering Instances

Instances are not accurately filtered by any single measure. In the course of this thesis many different tactics were attempted and a combination of measures appears to be the most effective. This section reviews the most successful measurements that were combined to produce the results discussed in Section 5.

4.5.1 Theory

There are three primary types of filtering performed by the current system, each of which addresses a different dimension of error that crops up in response to overly general patterns:

- Nonsense - The work in this thesis focused primarily on enriching and making more specific the connections within the existing database. A great deal of nonsensical and instance terms can be removed by requiring instance concepts to already exist, or synonyms of those concepts to exist, within the network. Garbage terms are also filtered by a set of lexical features

- Contextual Specificity - Instances that suffer from conditionality or scope in theory fall into a continuum of highly general, or true for most entities in most contexts, to more highly specific, or true only for some classes in special contexts. A measurement which can order topics by their global frequency relative to a baseline can be used in a band-pass filter that ignores both very common and very rare instances. A PMI technique is used to filter along the specificity axis as described below

- Conceptual Specificity - Some words and phrases are very broad entities "things can be," to very specific instances like "johnny has tape on his nose." Filtering by existing concepts is helpful for these variations as well[5]

- Correct vs. Incorrect members - The contextual locality of terms in ConceptNet can have a powerful effect in separating positive members from negative members of a relation all the above measures being equal. If one concept is in the conceptual neighborhood of another concept, then it is far more likely to be related than a concept that is far away or weakly connected

4.5.2 Lexical Analysis

Garbage terms are removed when they meet one of the following lexical conditions:

- Contain a non-ascii character (i.e., standard alphabetic characters). This filters all binary data that slips through the URL filters described above

[5]A dictionary lookup to remove instances or terms that are high in a formal class ontology could be helpful for single term concepts, but may have less effectiveness for multi-word phrases. Some phrases, however, can be filtered by whether their head verb or root noun passes the lookup test.

- Contains a numeric character. Phrases with numbers are unlikely to be of much use in common sense, although they are powerful as instance knowledge and so terms with any numbers are forbidden for now
- Terms must be more than two characters and less than 40
- Are on a stopword list. Stopwords are used extensively in NLP to avoid words that have a primarily syntactic role (determiners, conjunctions, etc). Personal pronouns are exempted as they indicate, typically, that an entity is a person or is acting like a person

4.5.3 Concept Filtering

The knowledge emphasized by this thesis is that which augments the existing nodes in ConceptNet. In this way, without significant scale of data in the web mining, existing knowledge can be employed to filter mined knowledge.

4.5.4 Pattern Retrieval Frequency

The number of patterns that retrieve a given instance and the number of times a pattern retrieves an instance can both be used to rank the likely salience of one concept to another under the pattern's relation. This has not proven to be highly useful in practice, due to the low probability of overlap resulting from the relatively limited number of instances extracted for each pattern. While it is expected that this measure will become more useful, initial experiments to determine for each prospect instance whether they match each of the patterns failed to show significant benefit.

4.5.5 Pointwise Mutual Information

Additional information useful for sorting signal from noise comes from a Pointwise Mutual Information (PMI) score commonly used in the literature to determine the relative salience of two terms to each other. For this application the PMI score is computed by querying a web engine for:

$$PMI(C_1, C_2) = (Hits(C_1 + C_2)/Hits(C_1))$$

Two thresholds were chosen empirically based on a set of supervised examples. Future work should give this measure a more formal treatment to understand whether this is a universal property of concepts and the web, or is sensitive to relation type, concept type or some other property. Scores that are high are typically highly general and scores that are low can be considered too rare to be Commonsense. This metric does not normalize by the target hit count because the goal is to rank concepts according to their degree of affinity with the source concept. Three search queries per relation are required to generate the PMI score.

For example filtering DesireOf "researcher" in this way removes general knowledge such as: "researcher," "page," "news," and "site" which are highly general terms. The low end of the filter removes instances such as "beta," "analyst," "designer," "desk," and "reporter." However the low end of the filter does remove a high amount of positive knowledge so future versions of ConceptMiner will attempt to compensate by combining the PMI score with other sources of evidence prior to filtering instances.

4.5.6 Inferential Distance

The final filter applied to retrieved instances uses inferential distance between the two concepts in the existing ConceptNet. Concepts are ranked according to the number of paths at the shortest path length. Pairs that are directly connected (rare) are ranked according to the number of link types already shared. This increases the prospect of type errors so an additional type selection step may be needed when this technique is applied at scale. Paths with two edges are ranked lower than directly connected and are internally ordered by the number of paths shared between the two concepts. This does appear to provide a better than random separator between positive and negative instances, although substantially reduces the total recall of positive instances.

This metric is highly sensitive to the knowledge that already exists in ConceptNet.

When ConceptMiner is scaled the goal should be to mine neighborhoods of concepts so that the mined knowledge can be searched for paths that compensate for the missing knowledge in ConceptNet.

4.5.7 Combining Measures

Future research may uncover more sophisticated ways of bookeeping evidence from these different scoring methods, but for the initial evaluation of mining Commonsense the aim is to focus on precision and then to sufficently characterize recall in order to understand what future opportunity remains.

In this example, each measurement was used to rank the instances and a threshold value empirically determined across a set of examples.

Filters were applied in the following order to move computationally inexpensive operations earlier in the sequence:

1. Basic Lexical Filters
2. Concept Filters
3. Inferential Distance Filter
4. Pointwise Mutual Information Filter

4.5.8 Other Experiments

Other explorations were performed to try to separate positive and negative instances of a class. One attempt used the ratio of hit counts for an instance instantiated in a set of patterns that use positive and negative surface expression patterns for instances of a relation (e.g., "X wants to Y" vs. "X does not want to Y"). There was not a sufficiently strong effect with this approach to make the details worth reporting.

Similar lack of benefit was seen using a small set of extraction patterns to attempt to validate extracted instances. It may be that there were too few patterns (and Web search hits were limiting) or that additional normalization was needed to account for absolute hit count differences.

want to Y"). There was not a sufficiently strong effect with this approach to make the details worth reporting.

Similar lack of benefit was seen using a small set of extraction patterns to attempt to validate extracted instances. It may be that there were too few patterns (and Web search hits were limiting) or that additional normalization was needed to account for absolute hit count differences.

4.6 LISP Implementation

ConceptMiner is written entirely in Common Lisp and was run primarily on under Franz Allegro versions 7.0 and 8.0. The system depends on several external free libraries for functions such as regular expression, networking and search engine access. Two libraries were created explicitly for this application to address particular challenges of the domain and a third was significantly modified to support my work.

- Langutils - A natural language processing library described elsewhere [Eslick and Liu, 2005]
- Process Components (PCOMP) - A Lisp Library that encapsulates Lisp functions in a dataflow model, enabling easy logging, error recovery, and asynchronous scheduling of components as well as ACID compliance when integrated with Elephant.
- Elephant Persistent Object Store - A pre-existing Lisp interface to the Berkeley DB library that enables an ACID compliant persistent object store in Common LISP. The code was enhanced for stability and particularily to include indexing features to make data management and querying trivial.

The features of the language environment and ConceptMiner implementation I found the most helpful in making progress with this thesis were:

- Process Components package (PCOMP) leverages the Common Lisp restart system to catch different classes of errors that occurred during search, fetching, or processing and to fix them, and restart the component, to ignore them or to

from each query. Given a set of relations the system can query the persistent store and regenerate all the extracted data.

- Elephant's transactionalism integrated with PCOMP's restart ability meant that errors in generating data would be aborted, making sure the system rarely fell into inconsistent states.

- The Langutil's representation of textual content was also very inexpensive to serialize to and from the persistent store as well as being an efficient in-memory representation.

An example of one of the extraction pipelines is captured below. This process mines for patterns given a list of input relations in *relation-list*. This structure is easy to modify by adding or removing components, inspecting ongoing dataflow and system state, etc. The results of the run are implicitely maintained in the slots of the persistent objects passed among the objects.

Each module in the container can be found under the :children heading and describes a specific stage in the processing of Web data. The :netlist section describes the dataflow connectivity of the modules. The important steps that these modules implemented was described in the opening section of this chapter.

```
(defcontainer pattern-miner
  (:children
   (rgen list-generator :list *relation-list* :rate 100 :manual nil)
   (qgen send-fn-results :fn 'generate-pattern-queries :threshold 10)
   (search url-fn-search :searcher-class 'google-searcher :urls 20
   :fn 'relation-pattern-search
   :threshold 10 :threaded t)
   (url-filter filter :filter-fn 'page-filter)
   (fetch fetch-url :numprocs 10)
   (tagger text-tagger :quanta 20 :print t)
   (pattern-extractor extract-patterns)
```

```
 (search url-fn-search :searcher-class 'google-searcher :urls 20
 :fn 'relation-pattern-search
 :threshold 10 :threaded t)
 (url-filter filter :filter-fn 'page-filter)
 (fetch fetch-url :numprocs 10)
 (tagger text-tagger :quanta 20 :print t)
 (pattern-extractor extract-patterns)
 (relay handle-markers :fn nil :type-filter :end-of-results))
(:netlist
 (rgen -> qgen)
 (qgen -> search)
 (search -> url-filter)
 (url-filter -> fetch)
 (fetch -> tagger)
 (tagger -> pattern-extractor)
 (pattern-extractor -> relay)))
```

Chapter 5

Results and Evaluation

At the end of the introductory chapter I introduced a set of criteria necessary to declare success in this effort:

1. The system can collect knowledge for many or all ConceptNet relation types

2. Collected knowledge is of a similar quality to that extant in ConceptNet

3. The knowledge collected can identify concepts, particularly multi-word concepts not already known to the ConceptNet knowledge database

4. The inference of ConceptNet improves instead of degrades in the presence of the collected knowledge

The current state of the system does not enable evaluation of items three and four. It emphasizes enrichment of the current concept set, rather than expanding it, and data for a sufficient number of relations is not yet available to perform quality experiments with ConceptNet inference. Item one is partially addressed by the use of three relation types. This chapter will provide the reader with the results of applying the system extract new relations for a small set of concepts.

5.1 Evaluation Standards

To characterize the quality of extracted instances it is critical to understand how much of the data is "good" and how much is "bad". Traditionally this would be a measure of precision relative to a clear baseline of all the members of a relation. However, the lack of a complete baseline, the ambiguous nature of the predicates employed, and the different classes of knowledge that can be identified within a relation means that a more nuanced evaluation is called for. The limited coverage of the current data reduces the validity of independant human judgements and the emphasis here is on understanding the relationship of the learned patterns and filtering measures in the distribution of resulting instances.

A set of mined instances, after basic lexical filtering, are hand labeled[1] with one of the six categories introduced in chapter 3. Precision is calculated according to these labels, accounting for both absolute positive instances as well as the positive plus neutral instances.

For a set of retrieved relations, R:

$$Precision(R) = Positive/Total$$

$$SoftPrecision(R) = (Positive + Conditional + Scope)/Total$$

Additionally, a relative recall measure is provided to characterize the amount of positive examples collected that were subject to false positives in the filtering process.

$$RelativeRecall = FilteredPositive/RetrievedPositive$$

This is not a true measure of recall against all possible instances of the relation. However, it does serve to characterize the opportunity cost of the filtering process and the potential room for improvement.

[1]The samples presented here were labelled by the author to ensure a consistent labeling. Future studies should include multiple annotators to measure the degree of agreement.

5.1.1 Labeled Data

I generated labeled data to illustrate filtering for the following concepts for each of the relations: EffectOf, CapableOf and DesireOf. The concepts are slightly different for each relation to correlate to the kinds of entities the relations are intended to modify DesireOf source concepts are typically single word noun entities whereas the EffectOf relation typically modifies actions: verbs or verb+noun pairs.

Each dataset is approximately 250 total samples. Some of the datasets are a random subset of the total instances retrieved from the Web and others returned less than 250 instances. For example in the case of DesireOf mining for the "dictator" concept, there were 1400 responses, 245 of which were labeled.

5.2 Generalized Pattern Distribution

Despite the highly general list of top 50 patterns, provided in concise form in Appendix A, there is a relatively small amount of overlap between the surface forms as shown in Table 5.1. There are likely to be additional differences between the part-of-speech templates in each of the X and Y slots that will serve to further separate extracted instances for similar concepts.

	DesireOf	CapableOf	EffectOf
DesireOf	100%	16%	12%
CapableOf	16%	100%	18%
EffectOf	12%	18%	100%

Table 5.1: Pattern Overlap Matrix

5.3 Instance and Filter Assessment

After the above patterns are used to extract instances, it is possible to evaluate the filter functions by applying them independently and collectively to the hand labeled results of the mining process. The impact of each filtering technique on precision for

each labeled category shows the kind of information that each filter can discriminate. When the filter fails to discriminate input instances it returns the original set. This case is shown in the tables as "N/A".

An example listing of the data sets used, after lexical filtering, is provided in Appendix B.

5.3.1 Raw Results from Patterns

The initial pattern mining run statistics listed in table 5.2 typically generate more negative data than positive and often under 50% on the soft-measure metric. Given the generality of patterns and the lack of a pattern frequency filter, this result is quite encouraging.

	Total	Positive	Negative	*Prec.*	*SoftPrec.*
DesireOf("dog")	276	51	133	18%	41%
DesireOf("researcher")	283	89	109	31%	55%
DesireOf("dictator")	245	56	111	22%	42%
CapableOf("dog")	227	37	153	16%	22%
CapableOf("researcher")	224	54	96	24%	50%
CapableOf("dictator")	19	6	8	32%	47%
EffectOf("bark dog")	287	20	158	7%	18%
EffectOf("dictatorship")	40	3	32	8%	15%

Table 5.2: Initial Distribution of Mined Data

The raw results will establish a reference dataset against which the filters will be evaluated. The collected data is not useful to a ConceptNet style inference in its current form; to be of use the *Precision* and *SoftPrecision* scores will both need to demonstrate significant improvement, preferably without significant impact to the relative recall measure.

5.3.2 Concept Filter

The concept filter merely drops any relations that do not consist of two words that exist in ConceptNet. Table 5.3 demonstrates that most positive instances in the datasets are already part of ConceptNet.

	Precision	Change	*SPrecision*	Change	Recall
DesireOf("dog")	22%	1.18x	49%	1.19x	96%
DesireOf("researcher")	36%	1.15x	61%	1.10x	92%
DesireOf("dictator")	30%	1.36x	58%	1.58x	83%
CapableOf("dog")	23%	1.39x	31%	1.41x	95%
CapableOf("researcher")	29%	1.19x	58%	1.16x	94%
CapableOf("dictator")	38%	1.19x	56%	1.19X	100%
EffectOf("bark dog")	11%	1.57x	26%	1.47x	90%
EffectOf("researcher")	8%	1.00x	15%	1.00x	100%

Table 5.3: Concept Filter Results

The only other additional trend deserving mention in this dataset is the high variance in the change in the *Precision* and *SoftPrecision*. This may be due to interactions between topics covered in ConceptNet vs. the query term polysemy creating a higher percentage of spurious concepts. In future work it would be worthwhile to examine correlations between word senses, precision of initial results, and topic distribution within the existing ConceptNet.

5.3.3 Inferential Distance Metric

The inferential distrance metric implicitly includes the concept filter above as paths can only exist between members of the database. Moreover, it also filters any concepts for which there are no paths in the database.

Because of the wide distribution of path lengths within the database, no single fixed threshold is a sufficient filter. Instead we take the median path ordered inversely by length and then by the number of connections at that minimum distance.

The median computation is performed over path scores which are pairs defined as minimum-path-length and number-paths-at-minimum-length:

```
(defun get-path-median (instances)
  (let ((scores (filter-if (lambda (x) (equal '(0 0) x))
   (sort (mapcar #'get-path-score instances)
 #'path-score-<))))
    (nth (ceiling (/ (length scores) 2)) scores)))
```

```
(defun path-score-< (ascore bscore)
  (cond ((and (= (first ascore) 0) (not (= (first bscore) 0)))
 nil)
((and (not (= (first ascore) 0)) (= (first bscore) 0))
 t)
((= (first ascore) (first bscore))
 (> (second ascore) (second bscore)))
(t
 (< (first ascore) (first bscore)))))
```

Table 5.4 provides the median generated for each query and the associated impact on our evaluation measures.

	Precision	Change	*SoftPrec*	Change	Rel Recall
DesireOf("dog")	32%	1.72X	63%	1.51X	53%
DesireOf("researcher")	38%	1.20x	64%	1.16x	74%
DesireOf("dictator")	36%	1.60x	72%	1.71x	33%
CapableOf("dog")	35%	2.12x	44%	1.98x	51%
CapableOf("researcher")	33%	1.38x	76%	1.52x	20%
CapableOf("dictator")	60%	1.90x	80%	1.69x	50%
EffectOf("bark dog")	11%	1.59x	24%	1.34x	35%
EffectOf("dictatorship")	N/A	1.00x	N/A	1.00x	N/A

Table 5.4: Inferential Distance Filter Results

The inferential distance filter will automatically accept any term that has a direct link as it is seeking to exploit the vast about of correlative knowledge in the ContextuallyRelatedTo predicate of ConceptNet. ConceptNet is constructed in such a way that when there is not enough evidence to determine the exact relation, but there is clear lexical or semantic correlation between the source and target concepts, a ContextuallyRelatedTo predicate is added to record this observation. Typically this enhances the contextual locality in get-context operations, but upgrading it to a more precise relation will aid more focused queries such as ConceptNet get-context projections.

The failure case marked with "N/A" failed because the source concept was not actually represented in ConceptNet.

5.3.4 PMI Filter

The Pointwise Mutual Information filter produced these scores by querying the Google search engine on September 5-8th, 2006. The results of this filter operating over the reference set are summarized in Table 5.5. The empirically determined band-pass thresholds used for all these runs were:

- Upper Bound: 0.6
- Lower Bound: 0.2

	Prec	Change	*SoftPrec*	Change	Rel. Recall
DesireOf("dog")	21%	1.13X	42%	1.01x	27%
DesireOf("researcher")	48%	1.50x	74%	1.35x	42%
DesireOf("dictator")	29%	1.30x	68%	1.62x	16%
CapableOf("dog")	22%	1.32x	31%	1.42x	30%
CapableOf("researcher")	29%	1.20x	57%	1.14x	50%
CapableOf("dictator")	33%	1.06x	58%	1.23x	67%
EffectOf("bark dog")	N/A	1.00x	N/A	1.00x	N/A
EffectOf("dictatorship")	11%	1.48x	17%	1.11x	67%

Table 5.5: Pointwise Mutual Information Filter Results

As with the inferential distance metric each relation type and concept had a slightly different optimal filter to balance precision and recall. The fixed thresholds appear well balanced, unlike in the inferential distance case. However further investigation is warranted to see if there is a property of the concept or relation that enables these thresholds to be determined emprically, or correct via some other information measure.

The case that fails here does so becauses the concept itself has no hits from the Web search engine. The PMI score needs to detect this case and augment by computing the hit count for surface forms. This variation was not performed for the results in this thesis.

5.3.5 Composed Filters

As described in section 4, we compose the first three filters above in the order of Concept, Inference and PMI to produce the results in Table 5.6.

	Prec.	Change	*SoftPrec.*	Change	Rel Recall
DesireOf("dog")	26%	1.38x	56%	1.35x	22%
DesireOf("researcher")	51%	1.65x	76%	1.38x	46%
DesireOf("dictator")	36%	1.64x	82%	1.95x	14%
CapableOf("dog")	18%	1.12x	30%	1.34x	22%
CapableOf("researcher")	24%	1.00x	72%	1.45x	13%
CapableOf("dictator")	50%	1.58x	75%	1.58x	33%
EffectOf("bark dog")	11%	1.62x	24%	1.36x	35%
EffectOf("researcher")	11%	1.48x	17%	1.11x	67%
AVERAGES	28%	1.48x	54%	1.44x	32%

Table 5.6: Label Distribution for Composed Filters

On average across the examples the combined filter demonstrates a modest overall precision that falls short of the required quality level for augmenting ConceptNet. However, a 48% improvement over the base precision and soft precision measures is demonstrated at a relatively low average recall (32%) implying that there is additional opportunity for filtering and improving the end result. Removing the EffectOf examples the average *SoftPrecision* result is 65% while the average *Precision* result is 34%. I believe these two results are more representative of how the system as a whole will behave. Future work will need to address the question of whether the instances incorporated under the *SoftPrecision* label actually help the inference as compared to the *Precision* items alone. This will help to characterize the actual distance between the current system and the target quality of mined results.

These aggregate results also demonstrate differences between the three relation types and how the effects of the filters combine. The EffectOf relation had a much lower signal-to-noise ratio. This may be due, in part, to insuffient training of the extraction pattern set for EffectOf and was certainly due to problems with the PMI filter and Inferential Distance filter not filtering for each of the two EffectOf examples.

In the case of (CapableOf "dog" X) the combined filter result has a lower precision

than the Inferential filter or PMI filter alone. The explanation for this case is that the PMI filter removed positive instances but failed to remove sufficient negative instances to compensate. This demonstrates that impairments in a given filter can combine negatively with other filters and it will be important to better characterize these interactions and determine if an additional filtering step can be introduced which compensations. In general, however, this case was the exception as in most runs the aggregate outperfomed the individual filters, but at a 2x cost in the relative recall.

A final observation is that the inferential distance filter, when it worked appeared to have a much stronger impact than the PMI filter on overall quality. This implies the most fruitful avenue for future work - understanding how to compute inferential distance for terms that are not contained within ConceptNet.

5.3.6 Pattern Recall Frequency

While there is a measure in the system that can order instances by the number of unique patterns that extracted them, very few concepts are retrieved up with sufficient frequency in the current system for this to be a useful measure.

If it were employed, this filter would lend itself to a single threshold, similar to that employed in ordering extraction patterns, computed by $Qlog(T)$; Q is the number of unique extractions and T is the total number of times the instance was extracted.

5.4 Supplementary Experiments

In the process of this thesis I performed a number of exploratory experiments which deserve mention.

5.4.1 OpenMind Sentence Web Retrieval

One assertion of the OpenMind project was that the OpenMind sentences were explicit expressions of concepts that are typically only expressed implicitly. The question

raised is how often do these sentences actually occur on the Web. Without reporting formal data, the initial experiment indicates that OpenMind sentences rarely occur on the Web, seeming to validate the OpenMind assertion. However it is not necessarily the case that the knowledge expressed in those sentences is unavailable on the Web, merely that it is more difficult to extract.

5.4.2 OpenMind Proof-of-Concept Study

Before embarking on the experiments described above, I performed a quick study [2] exploring whether a classifier-based approach was feasible by testing the baseline hypothesis, that the ConceptNet relations could be used to train a bag-of-words classifier to recognize the original source sentences; i.e., it was able to reverse engineer the hand-built regular expressions used to generate ConceptNet.

1. Retrieve sentences from 1/2 of the subset of the OpenMind corpus used to generate ConceptNet relation instance pairs[3]
2. Train a binary or N-way naive-Bayes classifier
    - Retrieve sentences containing a pair of terms from a relation
    - Extract the non-query lexical terms and add a positive training example
    - Randomly pick a negative example from among the other relations
3. Test the classifiers on the other 1/2 of the dataset

In the original experiment, an N-way Naive Bayes classifier achieved an average accuracy of 44%. However, a binary classifier for single relation on its own was able to achive 90% accuracy (a large number of false positives). However when a boosting classifier[4] was applied, the reliability jumped to 96% range. The templated sentences

[2]Peak Xu, an MIT undergraduate student in the UROP program, was responsible for the bulk of the analysis described here

[3]We emphasized 12 relations that had sufficient source sentences for training and were richer relation types; e.g., ContextuallyRelatedTo and K-lines were ignored as they are computable from lexical features and co-location information.

[4]Peak suggested and tested a simple Euclidean distance metric in a 12-dimensional

used in generating the OpenMind corpus led us to anticipate this result, but we were unclear if it was possible without adding features that captured bi-gram, tri-gram, phrase sequences or POS tags from the source sentences.

The intuition behind the success of the boosting algorithm is that the Naive Bayes algorithm misses conditional dependencies between specific words in the template phrases. These dependencies result in regular patterns of false positives across a set of binary classifiers that the boosting classifier could identify.

5.4.3 Bag-of-Features Approach

I hypothesized that some of the relationships between terms are difficult to find in the small context windows leveraged by the pattern based approach. Therefore I ran several experiments to see see whether it was possible to access relation-identifying features in textual information across multiple sentences or only implicitly expressed within a single sentence.

In a process very similar to the pattern mining described in chapter 4, the AltaVista search engine's NEAR[5] operator. This results in a large corpus of textual contexts where query terms were separated by between 10 and 30 words. This means that contexts often passed over sentence boundaries where traditional syntactic parsing methods would fail[6].

Three Naive Bayes classifiers were trained on these contexts and demonstrated an ability to classify novel contexts that expressed known relations with 70-80% accuracy on a small dataset. However the classifer alone was unable to discriminate between inversions.

However there were additional problems that may have been caused by the training set. A review of the top features of the classifier led to the subjective observation was that the key terms determining the separation of relation classes were topical nouns,

[5]I have found this operator to not be reliable over different runs. In part I believe that the company enables and disables. On some runs NEAR behaves like AND and in others it retrieves terms that lie within small windows of 20-30 words of each other.

[6]It is possible to formally link parse trees between sentences by looking for topic affinity, performing anaphor resolution and other methods, but the techniques are not sufficiently reliable and are typically too expensive to be deployed in a large-scale text mining context.

rather than the syntactic words seen in the top patterns in Appendix A.

This line of inquiry was abandoned in favor of the pattern based approach, but the quality of the separation was surprising and deserves further investigation.

Chapter 6

Conclusions

This thesis reports on a system which learns relation-specific extraction patterns from textual contexts retrieved from the Web using pairs of concepts from the ConceptNet database. Extraction patterns are used to extract instances of a relation from the Web and filters the resulting instances. Specifically an Inferential distance metric based on ConceptNet and a Web-based Pointwise Mutual Information measure were introduced to boost the quality of the extracted data.

The use of a large set of volunteer-contributed seed instances and, in particular, the ConceptNet graph structure to constrain large-scale mining of relations, is a novel contribution of this thesis. While the data reported here is by no means definitive proof of practical utility, the data shows significant promise for the goal of enabling unsupervised augmentation of ConceptNet at a level of quality approaching that of volunteer contributors.

As a methodology to increase the specificity and connectivity of existing knowledge, the system and algorithms presented shows even greater promise. Moreover, the powerful role of a body of existing knowledge to seed the Inferential distance metric, indicates a new research direction in the quest to develop large Commonsense databases as well as in the more general domain of information extraction. Validating acquired knowledge, not by statistical measures like PMI, but by semantic evidence that can serve to rule out or reinforce particular relationships may be the key to enabling practical, large-scale mining of relational knowledge.

To condone the approach taken by this thesis for large-scale acquisition, the analyses of the prior chapter need to be significantly scaled. Data is needed on a much wider variety of instances and relation types to better characterize where automated techniques can be applied and what degree of quality can be expected over a broad distribution of topics.

I conclude that to the extent that Commonsense knowledge can help the design of Human-Computer Interfaces, an automated knowledge acquisition methodology is an important boosting technique to augment background knowledge available to developers and interface designers.

6.1 Experimental Variations

The work necessary to improve quality and breadth of the acquired data falls into three categories. The first is better filtering of sources of noise on the Web and working around the query limits of existing search databases. The second is to improve the use of syntactic and lexical information in pattern learning and the third is to understand how far the knowledge-based filtering techniques can be taken in the context of large-scale mined, rather the pre-existing, knowledge.

6.1.1 Pre-processing Retrieved Contexts

Preprocessing of data can have a large impact on the quality of the returned results. It can also have an impact on recall that should be analyzed before committing to a particular set of heuristics. The measures I would recommend trying for improvements on this work include:

- A better blacklist for sites that contain garbage information or copies of various existing Commonsense data (such as the original OMCS corpus!)

- Filter text contexts in both the pattern discovery and classifier training procedures by matching the dominant part-of-speech of the ConceptNet nodes and

that in the retrieved text. This would serve to remove much of the noise introduced by polysemy. This requires regenerating ConceptNet with this information

- Compare precision and recall on a set of known concepts against a set of preferred sources, such as wikipedia and children's education sites

- Provide support for parsing plaintext from Acrobat files. A great deal of high quality text on the Web is stored in PDF files. The coverage of an extraction system could be increased by adding support for PDF files with text content

6.1.2 Richer Feature and Pattern Representations

In order to extend the types of textual contexts from which we can identify relations, there are extensions to the existing system that need to be made to the feature spaces employed in the classifier approach as well as the pattern representations for the extraction-pattern approach. Specifically:

- Rerun the classifier experiments using a set of relations from a shared set of topics to remove the noise caused by topic-based descrimination of relation types.

- Explore more complex phrases and phrase templates in the Naive Bayes context. Combining some of the short templates discovered in template matching and finding ways to filter contexts by whether the concept in one sentence connects to the other is a more aggressive tactic for pattern recognition.

- Semantic templates. Of particular interest is to identify whether or not the contexts with implicit Commonsense expressed over multiple sentences could be captured by more complex templates that link local syntactic patterns to global semantic relationships.

- Improve overlap assessment by combining concepts that are paraphrases or synonyms of each other

- Improve and systematize the relationship between the canonical form of ConceptNet relations and the surface forms used to query the web. Poor support for this relationship limits the number of compound phrases extracted by the current system.

For example, consider the two sentences: "I played with my dog yesterday. I was annoyed by the barking and running around." The knowledge implicit in them is:

```
Syntactic:
(CapableOf "dog" "play with person") <= personal pronoun
(CapableOf "dog" "owned by person") <= target of personal possessive
(CapableOf "person" "play with person's dog") <= my and I agree
(CapableOf "person" "annoyed") <= "person was X"
(CapableOf "person" "annoyed by barking") <= "person was X1 and X2"
(CapableOf "person" "annoyed by running around") <= ""

Given semantics:
(CapableOf "dog" "bark")
(CapableOf "dog" "run around")
conclude...
(CapableOf "dog" "annoy person")
```

The challenge above requires a first level of anaphor resolution (the subject pronouns in each sentence are the same entity), but further demands that the origin of the behaviors in the object sentence be identified implicitly. The order and attachment of phrases in the two sentences along with the semantics of words used might be sufficient features to enable classification of the above sentence with sentences like the following: "I played with my dog yesterday. I was annoyed by the heat and humidity, so didn't have much fun." In this one, the clause at the end serves to explain the introduction of heat and humidity. Without this additional clause, the default assumption would be that the object of the prior sentence's prepositional phrase, the dog, had hot and humid properties which served to annoy the speaker.

6.1.3 Filtering and Knowledge

The final, and most promising area of inquiry is to expand on the tantilizing results from the inferential distance filter. Rather than looking at total path depth and count, it would be fruitful to descriminate by the actual relation types in the paths when comparing potential relatedness of a mined instance target concept to the source concept. For example the quality of a transitive closure (of depth 2) varies greatly by relation type.

It is also possible to link new concepts into ConceptNet rather than restricting the analysis purely to existing concepts. Low-hanging fruit here are using WordNet synsets to link new synonyms into the network to enable new paths to be discovered. Analysis of existing relations and antonym sets would enable additional inversions to be filtered from the retrieved datasets. Validation PMI statistics could be run, for example, to determine the affinity of a source relation and relation pattern to one or the other of the antonym pairs.

Moreoever, as mentioned in section 4.5.6 this metric could be used to constrain a mutual-bootstrapping or clustering methodology to identify highly affiliated concept groups in the mined data itself. In this way the reference, ConceptNet, would provide a source of constraint in constructing a new graph which over multiple mining runs would converge to a similarly structured topology.

Appendix A

Patterns

A.1 Top 50 Patterns for DesireOf

"X/* might/MD Y/*"

"X/* s/PRP Y/*"

"Y/* for/IN X/*"

"X/* or/CC Y/*"

"X/* is/V Y/*"

"one/CD X/* Y/*"

"X/* Y/* with/IN"

"X/* shall/MD Y/*"

"X/* with/IN Y/*"

"Y/* by/IN X/*"

"Y/* to/TO X/*"

"Y/* if/IN X/*"

"of/IN X/* '/POS s/PRP Y/*"

"Y/* of/IN X/*"

"of/IN X/* Y/*"

"one/CD X/* s/PRP Y/*"

"X/* had/V Y/*"

"with/IN X/* Y/*"

"X/* '/POS s/PRP Y/*"

"Y/* X/* and/CC"

"X/* who/WP does/V Y/*"

"X/* would/MD Y/*"

"X/* must/MD Y/*"

"X/* Y/* '/POS"

"in/IN X/* Y/*"

"X/* Y/*"

"X/* Y/* of/IN"

"Y/* X/* can/MD"

"X/* Y/* for/IN"

"for/IN Y/* X/*"

"X/* should/MD Y/*"

"X/* with/IN Y/*"

"X/* will/MD Y/*"

"X/* Y/* '/POS s/PRP"

"X/* whose/WP$ Y/* was/V"

"X/* should/MD Y/*"

"X/* their/PRP$ Y/*"
"X/* and/CC Y/*"
"X/* Y/* in/IN"
"if/IN X/* has/V Y/*"
"X/* will/MD Y/* and/CC"
"X/* '/POS s/PRP Y/* is/V"
"X/* does/V not/RB Y/*"
"Y/* but/CC X/*"
"X/* whose/WP$ Y/*"
"X/* Y/* ((("
"X/* have/V Y/*"
"by/IN X/* Y/*"
"X/* shall/MD Y/* or/CC"
"X/* to/TO Y/*"

A.2 Top 50 Patterns for EffectOf

"X/* he/PRP Y/*"
"X/* we/PRP Y/*"
"X/* she/PRP Y/*"
"Y/* when/WRB i/NN X/*"
"X/* without/IN Y/*"
"to/TO X/* Y/*"
"Y/* you/PRP X/*"
"X/* they/PRP Y/*"
"Y/* how/WRB to/TO X/*"
"to/TO X/* without/IN Y/*"
"X/* i/NN Y/*"
"X/* and/CC Y/*"
"X/* Y/* is/V"
"Y/* or/CC X/*"
"you/PRP Y/* you/PRP X/*"
"X/* during/IN Y/*"
"X/* at/IN Y/*"
"X/* so/RB Y/*"
"Y/* out/IN X/*"
"X/* Y/*"
"X/* in/IN Y/*"
"Y/* or/CC X/*"
"to/TO X/* and/CC Y/*"
"Y/* X/*"
"X/* Y/* of/IN"
"X/* her/PRP$ Y/*"
"Y/* X/* and/CC"
"to/TO Y/* and/CC X/*"
"to/TO Y/* or/CC X/*"
"X/* then/RB Y/*"
"Y/* but/CC X/*"
"Y/* if/IN they/PRP X/*"
"Y/* he/PRP X/* for/IN"
"X/* from/IN Y/*"
"X/* i/NN Y/* to/TO"
"Y/* just/RB X/*"
"Y/* /NN X/*"
"X/* without/IN Y/*"
"Y/* is/V X/*"
"they/PRP X/* they/PRP Y/*"

"and/CC X/* Y/*"
"Y/* while/IN X/*"
"you/PRP X/* Y/*"
"X/* Y/* in/IN"
"Y/* and/CC X/* with/IN"
"X/* of/IN Y/*"
"they/PRP X/* Y/*"
"X/* his/PRP$ Y/*"
"Y/* when/WRB they/PRP X/*"
"X/* your/JJ Y/*"

A.3 Top 50 Patterns for CapableOf

"X/* we/PRP Y/*"
"Y/* to/TO X/* and/CC"
"Y/* X/* would/MD"
"s/PRP X/* Y/*"
"how/WRB X/* Y/*"
"X/* Y/* is/V"
"X/* has/V Y/*"
"X/* does/V not/RB Y/*"
"do/VB n't/RB Y/* X/*"
"X/* will/MD Y/*"
"X/* never/RB Y/*"
"Y/* to/TO X/* in/IN"
"X/* then/RB Y/*"
"X/* did/V Y/*"
"for/IN X/* Y/*"
"Y/* X/* ca/MD"
"Y/* X/* was/V"
"Y/* X/* ca/MD n't/RB"
"X/* and/CC Y/*"
"X/* Y/* as/NNP"
"Y/* through/IN X/*"
"X/* also/RB Y/*"
"X/* which/WDT Y/*"
"X/* he/PRP Y/*"
"X/* does/V Y/*"
"n't/RB Y/* X/*"
"X/* will/MD Y/*"
"for/IN X/* to/TO Y/*"
"Y/* by/IN X/*"
"Y/* in/IN X/*"
"X/* would/MD Y/*"
"Y/* /NN X/*"
"Y/* my/PRP\$ X/*"
"X/* also/RB Y/*"
"X/* Y/* of/IN"
"of/IN X/* Y/*"
"all/PDT Y/* X/*"
"way/NN X/* Y/*"
"or/CC X/* Y/*"
"X/* they/PRP Y/*"
"X/* who/WP Y/*"
"X/* do/VB n't/RB Y/*"
"Y/* more/RBR X/*"
"X/* did/V Y/*"

```
"X/* only/RB Y/*"
"X/* may/MD Y/*"
"Y/* X/* '/POS"
"Y/* for/IN X/*"
"Y/* X/* are/V"
"X/* she/PRP Y/*"
```

Appendix B

Mined Instances

This appendix contains a subset of the labelled datasets used in the evaluation of Section 5. It is provided for purposes of reference and for developing intuition through reviewing the data. Category label annotations are attached to each instance target according to the following legend:

- Positive (P)
- Conditional (C)
- Scope (S)
- Type (T)
- Negative (N)

B.1 DesireOf Researcher Example Dataset

Before filtering: (DesireOf "researcher" X) for all $X \epsilon$

`versa. (N)`	`pot (N)`	`yup (N)`
`activist (C)`	`life (S)`	`act (S)`
`lover (S)`	`people (S)`	`inquire (P)`
`spend one-fourth (N)`	`benefit (N)`	`contribute (P)`

advance (P) requested. (N) identification (N)
congress (N) university (P) reporter (P)
legislator (P) analyst (C) family (S)
publication (P) product (P) agent (C)
development (P) engineer (C) practice (P)
entity (N) laboratory (P) process (P)
confidential (C) invention (P) originator (P)
offline (N) microsoft (N) java (N)
compaq (N) cash (S) continuity (N)
cycle (N) beta (N) tester (N)
description (N) phone (S) cbs (N)
project (P) director (C) station (N)
depot (N) retailer (N) supplies (P)
sticker (N) brand (N) image (N)
pant (N) patch (N) fabric (N)
wenner (N) operation (N) material (S)
practitioner (T) nyt (N) ngos (C)
refusal (N) customer (N) mary (N)
beginner (N) username (N) ethic (P)
industry (C) snyder (N) rock (N)
spot (N) market (C) control (P)
business (C) mold (N) graduation (C)
labour (P) hawaiireporter.com (N) status (P)
logo (N) feature (P) mask (C)
review (P) cpap (N) area (P)
institute (P) health (S) neuroscience (C)
conference (P) microinjection (C) grant (P)
essay (N) transnet (N) network (S)
technology-enabled (C) networld+interop (N) cenzic (N)
internal (N) software (P) mcgonigal (N)

morning (S)
criticism (P)
equipment (P)
reference (P)
department (P)
window (N)
political expediency (C)
medicine (C)
therapy (C)
read (P)
electronic (N)
scan (N)
statistician (N)
weakness (I)
performance (P)
clinician. (N)
your inter (G)
number (N)
architecture (N)
optimizations (P)
student (P)
language (C)
art (I)
class (C)
interview (C)
helper (P)
citation (P)
output (N)
h-index (C)
profit (C)
designer (N)
bogost (N)
academician. (T)
fein (N)
clinician (N)
doxpara (N)
foundation (P)
scientist (P)
president (N)
select (N)
agreement (C)
scholar (T)
test (P)
sponsor (P)
evaluator (P)
partici.. (G)
implementation (C)
broker (N)
technology (N)
speculation (P)
tool (P)
framework (P)
institution (P)
program (P)
server (C)
best-kept (N)
contribution (P)
visibility (P)
impact (P)
internet (P)
rhetoric (P)
e-mail (C)
engineering (C)
professorship (P)
information (P)
suri (N)
california (N)
pediatrician (N)
desk (S)
advice (P)
deposit (N)
understand (P)
economy (C)
observation (P)
explanation (P)
survey (P)
tradeoffs (P)
patent (P)
environment (C)
compilation (C)
pattern (P)
surgery (N)
protein (N)
system (P)
agricultural (N)
jul (G)
numb (N)
coverage (N)
krokstad (N)
israel (N)

fraud (I)
money (P)
roitman (N)
agenda (P)
lappan (N)
email (S)
trauma (I)
town (N)
maynor (N)
center for international (N)
inability (I)
pivx (N)
milton (N)
news (P)
agency (C)
clock (S)
appointment (C)
competence (P)
policy (C)
consciousness (C)
plan (P)
creator (I)
library (P)
menu (N)
nutrition (S)
resource (N)
design (P)
research (P)
group (N)
encounter (P)

application (C)
mathematics (P)
board (N)
council (N)
toobar (G)
academic (T)
staines (N)
issue (P)
biography (N)
blog (S)
agentless (N)
manufacture (C)
extremelabs (N)
page (N)
company (C)
committee (P)
effectiveness (P)
professor (C)
personnel (P)
painting (N)
deck (N)
video (N)
manager (I)
site (N)
rate (N)
history (C)
developer (C)
presentation (P)
skill (P)
expect similar (P)

education (P)
level (N)
topology (C)
state (S)
bhis (N)
zone (P)
coder (C)
subscription (C)
unique drivers. (N)
lunch (S)
heighten security (N)
exchange (N)
security (S)
leadership (C)
recruitment (C)
member (N)
quality (P)
activity (N)
work (P)
gibson (N)
tag (N)
jazz (N)
requirement (I)
ecstacy (N)
space (N)
researcher (P)
consultant (C)
article (P)
uncover (P)
hypothesize (P)

conclude (P)

After composite filtering: (DesireOf "researcher" X) for all $X \epsilon$

university (P)
act (S)
microsoft (N)
phone (S)
control (P)
software (P)
president (N)
agreement (C)
technology (N)
tool (P)
class (C)
education (P)
town (N)
company (C)
library (P)
design (P)
language (C)
family (S)
cash (S)
material (S)
status (P)
equipment (P)
select (N)
understand (P)
environment (C)
surgery (N)
program (P)
money (P)
exchange (N)
quality (P)
space (N)
manager (I)
practice (P)
cycle (N)
rock (N)
conference (P)
window (N)
advice (P)
test (P)
student (P)
art (I)
internet (P)
level (N)
security (S)
plan (P)
history (C)

Bibliography

[Ahn and Picard, 2006] Ahn, H. and Picard, R. W. (2006). Affective cognitive learning and decision making: The role of emotions. In *The 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)*, Vienna, Austria.

[Brill, 1990] Brill, E. (1990). A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy. Association for Computational Linguistics.

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

[Craven et al., 2000] Craven, M., Dipasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. (2000). Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1/2):69–113.

[Eslick and Liu, 2005] Eslick, I. and Liu, H. (2005). Langutils: A natural language toolkit for common lisp. In *Proceedings of the International Lisp Conference (ILC'2005)*, Stanford, CA.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. (2004). Web-scale information extraction in KnowItAll (preliminary results). In *Proceedings of the 13th World Wide Web Conference*, pages 100–109.

[Geleijnse and Korst, 2006] Geleijnse, G. and Korst, J. (2006). Learning effective surface text patterns for information extraction. In *Proceedings of EACL 2006 Workshop on Adaptive Text Extraction and Mining*, Trento, Italy.

[Gladwell, 2005] Gladwell, M. (2005). *Blink: The Power of Thinking without Thinking.* Little, Brown.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the fourteenth international conference on computational linguistics*, pages 539–545, Nantes, France.

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

[Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI.

[Lieberman et al., 2004] Lieberman, H., Liu, H., Singh, P., and Berry, B. (2004). Beating common sense into interactive applications. *Artificial Intelligence Magazine*, 25(4):63–76.

[Liu, 2002] Liu, H. (2002). Semantic understanding and commonsense reasoning in an adaptive photo agent. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

[Liu, 2003] Liu, H. (2003). Unpacking meaning from words: A context-centered approach to computational lexicon design. In et al. (Eds), B., editor, *Modeling and Using Context, 4th International and Interdisciplinary Conference, CONTEXT 2003*, pages 218–232, Stanford, CA.

[Liu et al., 2002] Liu, H., Lieberman, H., and Selker, T. (2002). Goose: A goal-oriented search engine with commonsense. In Bra, D., Brusilovsky, and Conejo,

editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference*, pages 253–263, Malaga, Spain.

[Liu et al., 2003] Liu, H., Lieberman, H., and Selker, T. (2003). A model of textual affect sensing using real-world knowledge. In *Proceedings of the ACM International Conference on Intelligent User Interfaces IUI 2003*, pages 125–132, Miami, FL, USA. ACM.

[Liu and Singh, 2004] Liu, H. and Singh, P. (2004). Conceptnet: a practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4):211–226.

[McCarthy, 1986] McCarthy, J. (1986). Programs with common sense. In *Readings in Knowledge Representation.* Morgan Kaufmann, Los Altos, CA.

[Minsky, 1975] Minsky, M. (1975). A framework for representing knowledge. In Winston, P., editor, *The Psychology of Computer Vision*, pages 211–277. Mcgraw-Hill, New York.

[Minsky, 1988] Minsky, M. (1988). *SOCIETY OF MIND.* Simon & Schuster.

[Minsky, 2006] Minsky, M. (2006). *THE EMOTION MACHINE: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind.* Simon & Schuster.

[Montemagni and Vanderwende, 1992] Montemagni, S. and Vanderwende, L. (1992). Structural patterns vs. string patterns for extracting semantic information from dictionaries. In *Proceedings of COLING92*, pages 546–552.

[Mueller, 1999] Mueller, E. (1999). Prospects for in-depth story understanding by computer. http://www.signiform.com/erik/pubs/storyund.htm.

[Musa et al., 2003] Musa, R., Scheidegger, M., Kulas, A., and Anguilet, Y. (2003). Globuddy, a dynamic broad context phrase book. In *Lecture Notes in Computer Science*, volume 2680/2003, pages 467–474. Springer Berlin / Heidelberg.

[Oertel and Amir, 2005] Oertel, P. and Amir, E. (2005). Commonsense knowledge retrieval. In *AAAI Spring symposium workshop on Knowledge Collection from Volunteer Contributors.*

[Pantel and Pennacchiotti, 2006] Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL-06)*, Sydney, Australia.

[Pantel et al., 2004] Pantel, P., Ravichandran, D., and Hovy, E. (2004). Towards terascale knowledge acquisition. In *Proceedings of Conference on Computational Linguistics (COLING-04)*, pages 771–777, Geneva, Switzerland.

[Riesbeck and Schank, 1998] Riesbeck, C. K. and Schank, R. C. (1998). *Inside Case-Based Reasoning.* Lawrence Erlbaum Associates.

[Riloff, 1996] Riloff, E. (1996). An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, 85.

[Riloff and Jones, 1999] Riloff, E. and Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-level Boot-strapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 1044–1049. The AAAI Press/MIT Press.

[Rosario and Hearst, 2001] Rosario, B. and Hearst, M. (2001). Classifying the semantic relations in noun compounds via a domain-specific lexical heirarchy. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*, Pittsburgh, PA.

[Schank and Abelson, 1977] Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Press, Hillsdale, N.J.

[Schubert and Tong, 2003] Schubert, L. and Tong, M. (2003). Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT/NAACL Workshop on Text Meaning*, Edmonton, Alberta, Canada.

[Singh, 2002] Singh, P. (2002). The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, CA. AAAI.

[Singh, 2005] Singh, P. (2005). *EM-ONE: An Architecture for Reflective Commonsense Thinking.* PhD thesis, Massachusetts Institute of Technology.

[Sowa, 1990] Sowa, J. (1990). Crystallizing theories out of knowledge soup. *Intelligent Systems: State of the Art and Future Directions*, pages 456–487.

[Stocky et al., 2004] Stocky, T., Faaborg, A., and Lieberman, H. (2004). A commonsense approach to predictive text entry. In *Proceedings of Conference on Human Factors in Computing Systems*, Vienna, Austria.

[Tzeng, 2004] Tzeng, J.-Y. (2004). Polite computers. *International Journal of Human-Computer Studies*, 61(3):319–345.

[Vanderwal, 2005] Vanderwal, T. (2005). Off the top: Folksonomy entries. http://www.vanderwal.net/random/index.php.

[Vanderwende, 2005] Vanderwende, L. (2005). Volunteers created the web. In *Proceedings of the AAAI 2005 Spring Symposium Series, working notes of the symposium on Knowledgte Collection from Volunteer Contributors*, Stanford, CA.

[Xiao et al., 2004] Xiao, J., Stasko, J., and Catrambone, R. (2004). An empirical study of the effect of agent compentence on user performance and perception. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, New York, New York.