Commonsense Reasoning in and over Natural Language

Hugo Liu and Push Singh

Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{hugo,push}@media.mit.edu

Abstract. ConceptNet is a very large semantic network of commonsense knowledge suitable for making various kinds of practical inferences over text. ConceptNet captures a wide range of commonsense concepts and relations like those in Cyc, while its simple semantic network structure lends it an ease-of-use comparable to WordNet. To meet the dual challenge of having to encode complex higher-order concepts, and maintaining ease-of-use, we introduce a novel use of semi-structured natural language fragments as the knowledge representation of commonsense concepts. In this paper, we present a methodology for reasoning flexibly about these semi-structured natural language fragments. We also examine the tradeoffs associated with representing commonsense knowledge in formal logic versus in natural language. We conclude that the flexibility of natural language makes it a highly suitable representation for achieving practical inferences over text, such as context finding, inference chaining, and conceptual analogy.

1 What is ConceptNet?

ConceptNet (www.conceptnet.org) is the largest freely available, machine-useable commonsense resource. Structured as a network of semi-structured natural language fragments, ConceptNet presently consists of over 250,000 elements of commonsense knowledge. We were inspired dually by the range of commonsense concepts and relations in Cyc (Lenat, 1995), and by the ease-of-use of WordNet (Fellbaum, 1998), and hoped to combine the best of both worlds. As a result, we adopted the semantic network representation of WordNet, but extended the representation in several key ways.

First, we extended WordNet's lexical notion of nodes to a conceptual notion of nodes, but we kept the nodes in natural language, because one of the primary strengths of WordNet in the textual domain is that its knowledge representation is itself textual. ConceptNet's nodes are thus natural language fragments which are semi-structured according to an ontology of allowable syntactic patterns, and accommodate both first-order concepts given as noun phrases (*e.g.* "potato chips"), and second-order concepts given as verb phrases (*e.g.* "buy potato chips").

Second, we extended WordNet's small ontology of semantic relations, which are primarily taxonomic in nature, to include a richer set of relations appropriate to concept-level nodes. At present there are 19 semantic relations used in ConceptNet, representing categories of, *inter alia,* temporal, spatial, causal, and functional

knowledge. By combining higher order nodes with this relational ontology, it is possible to represent richer kinds of knowledge in ConceptNet beyond what can be represented in WordNet (Fig 1.). For example, we can represent a layman's common sense observation that "you may be hurt if you get into an accident" in ConceptNet as EffectOf("get into accident", "be hurt"). Note that because the knowledge representation is semi-structured natural language, there are often various ways to represent the same knowledge. This is a source of ambiguity, but as we will argue in this paper, maintaining some ambiguity lends us greater flexibility for reasoning.

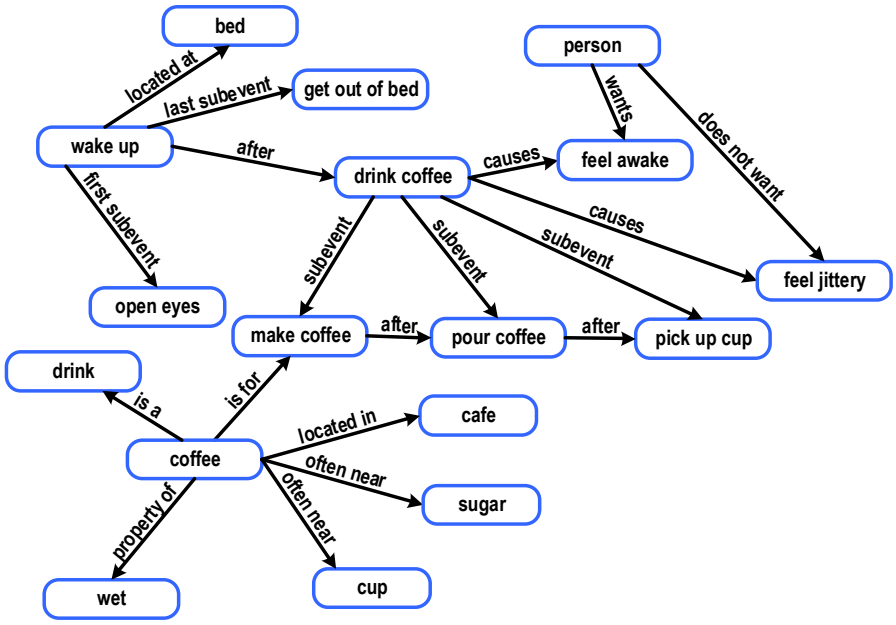

Fig. 1. An excerpt from ConceptNet's semantic network of commonsense knowledge. Relation names are expanded here for clarity.

Third, we supplement the ConceptNet semantic network with some methodology for reasoning over semi-structured natural language fragments. This methodology prescribes techniques for managing the ambiguity of natural language fragments, and for determining the context-specific similarity of nodes. For example, sometimes we want the nodes "buy food" and "purchase groceries" to be synonymous in an inference chain, and other times, not.

Fourth, we supplement the ConceptNet semantic network with a toolkit and API which supports making practical commonsense inferences about text, such as context finding, inference chaining, and conceptual analogy.

In a related paper (Liu & Singh, 2004a), we describe how ConceptNet was created, how it is structured, how it compares with other commonsense knowledge bases, and how it has been used and evaluated. This paper begins with a few perti-

nent details of the system but largely focuses on the knowledge representation aspect of ConceptNet, that is, how to reason over concepts expressed as semi-structured natural language fragments. We also speak more generally about the suitability of natural language as a knowledge representation for commonsense reasoning.

This paper is structured as follows. First, we give a brief history of the origin and structure of ConceptNet. Second, we discuss the affordances and limitations of representing commonsense knowledge in natural language, particularly in regards to its suitability for making practical commonsense inferences about text. Third, we present some methodology for reasoning about semi-structured natural language concepts in ConceptNet. We conclude with a summary of contribution.

2 Origin and Structure of ConceptNet

We explain briefly the history of ConceptNet with respect to the Open Mind Commonsense (OMCS) Project, how ConceptNet was mined from the OMCS repository of sentences, and how ConceptNet's nodes and edges are structured.

2.1 Origin

ConceptNet is a machine-usable resource mined out of the Open Mind Commonsense (OMCS) corpus (Singh *et al.* 2002), a collection of nearly 700,000 English sentences of commonsense facts collected from over 14,000 contributors from around the world, over the past three years.

CRIS (Commonsense Robust Inference System), ConceptNet's earliest predecessor, mined predicate argument structures from OMCS, where arguments were semi-structured natural language fragments, and used this for conceptual expansion (Liu & Lieberman, 2002). Since then, we finalized the ontology of semantic relations and added an inference toolkit that is distributed with the semantic network to support some practically useful textual inferences tasks. Recently, we also added methods for automatically comparing and reconciling natural language nodes to make inference more flexible and robust.

ConceptNet is produced by an automatic process which applies a set of 'commonsense extraction rules' to the semi-structured English sentences of the OMCS corpus. The key to being able to do this is that the OMCS website already elicits knowledge in a *semi-structured* way by prompting users with fill-in-the-blank templates (*e.g.* "*The effect of* [falling off a bike] *is* [you get hurt]"). A pattern matching parser uses roughly 40 mapping rules to easily parse semi-structured sentences into an ontology of predicate relations, and arguments which are short fragments of English. These arguments are then normalized to conform to preferred syntactic patterns. Certain stop-words and stop-parts-of-speech are filtered out, and the verb and nouns are reduced to their canonical base forms. A small part-of-speech-driven grammar filters out non-compliant text fragments (thus only a subset of the OMCS

knowledge is used in ConceptNet) to ensure all arguments conform to these syntactic constraints.

2.2 Structure

ConceptNet nodes are natural language fragments semi-structured to conform to preferred syntactic patterns which fall into three general classes: Noun Phrases (things, places, people), Attributes (modifiers), and Activity Phrases (actions and actions compounded with a noun phrase or prepositional phrase, *e.g.*: "turn on water," "wash hair."). In the normalization process, verbs are stripped to their base form, the count of nouns is stripped, and parts-of-speech which have lesser semantic value like determiners (*e.g.* "a", "the", "two") and modals (*e.g.* "might", "could", "may") are stripped. A portion of the concept grammar is given as part-of-speech patterns in Table 1.

Table 1. Grammar for Partially Structuring Natural Language Concepts

Node class	A portion of the grammar	Examples of valid nodes
Noun Phrases	NOUN; NOUN NOUN; ADJ NOUN; NOUN PREP NOUN	"apple"; "San Francisco"; "fast car"; "life of party"
Attributes	ADJ; ADV ADJ	"red"; "very red"
Activity Phrases	VERB; VERB NOUN; ADV VERB; VERB PREP NOUN; VERB NOUN PREP NOUN	"eat"; "eat cookie"; "quickly eat"; "get into accident"; "eat food with fork"

ConceptNet edges are described by an ontology of at present 19 binary relations shown below in Table 2. These relations were chosen because the original OMCS corpus was built largely through its users filling in the blanks of templates like 'a hammer is for _____'. Thus the relations we chose to extract largely reflect the original choice of templates used on the OMCS web site.

Table 2. Semantic Relation Types currently in ConceptNet

Category	Semantic Relations – *(and an explanation)*
Things	IsA – *(corresponds loosely to hypernym in WordNet)* PropertyOf – (*e.g. (PropertyOf "apple" "healthy"))* PartOf – *(corresponds loosely to holonym in WordNet)* MadeOf – *(e.g. (MadeOf "bottle" "plastic"))*
Events	FirstSubeventOf, LastSubeventOf – *(e.g. (FirstSubeventOf "act in play" "learn script"))* EventForGoalEvent – *(e.g. (EventForGoalEvent "drive to grocery store" "buy food"))* EventForGoalState – *(e.g. (EventForGoalState "meditate" "enlightenment"))* EventRequiresObject – *(e.g. (EventRequiresObject "apply for job" "resume"))*
Actions	EffectOf – (*e.g. (EffectOf "commit perjury" "go to jail"))* EffectOfIsState – *(e.g. (EffectOfIsState "commit perjury" "criminal prosecution"))* CapableOf – *(e.g. (CapableOf "police officer" "make arrest"))*
Spatial	OftenNear – *(e.g. (OftenNear "sailboat" "marina"))* LocationOf – *(e.g. (LocationOf "money" "in bank account"))*

Goals	DesiresEvent, DesiresNotEvent – *(e.g. (DesiresEvent "child" "be loved"))*
Functions	UsedFor – *(e.g. (UsedFor "whistle" "attract attention"))*
Generic	CanDo – *(e.g. (CanDo "ball" "bounce"))* ConceptuallyRelatedTo – *(e.g. (ConceptuallyRelatedTo "wedding" "bride" "groom")*

As illustrated by the examples in Table 1, the semantics of the predicate relations in ConceptNet are quite informal. Even for a particular semantic relation, the syntactic and/or semantic type of the arguments are not formally constrained, though some predicate names imply some typing (*e.g.* EventForGoalEvent, EventForGoalState, EventRequiresObject). In general, the usage and scope of each semantic relation can be best and most intuitively ascertained by looking at the original choice of templates used on the OMCS web site (At: http://openmind.media.mit.edu)

3 Commonsense Reasoning in Natural Language?

In this section, we discuss some of the strengths and weaknesses of representing and reasoning with commonsense knowledge in natural language.

3.1 Where logic excels

There is an important representational tradeoff between logic and natural language. Logic is precise – its symbols are unambiguous, and inference amounts to deductive theorem proving. Its strength lies in its stable and systematic way of evaluating and maintaining the truth of expressions. In technical domains with little ambiguity where precision is important, logic is a superb framework. But what about the vague notion of a "common sense" domain? Cyc, for example, represents its commonsense knowledge in a language called CycL, which is essentially a second-order logical language with second-order features such as quantification over predicates. John McCarthy first outlined the basic approach of representing commonsense knowledge with predicate logic in his classic paper, "Programs with Common Sense" (1958). Examples given in the paper seem quite appealing for their elegance. For example, you can represent the statement, "if you are at your car and you drive from home to the airport, then you are at the airport," using the following logical statement: *canachult*(*at*(*I*,*car*), *go*(*home*,*airport*,*driving*), *at*(*I*,*airport*)). Yet the development and application of commonsense reasoning systems using the logical approach has turned out to be not so straightforward as this example might suggest. To make this example really work requires a substantial amount of additional logical scaffolding, to precisely define the terms used in the statement and their interrelationships, which has turned out to be a task of daunting complexity.

3.2 How do people represent and reason?

There seems to be a divergence between the logical approach to reasoning and what is known about how people reason. Human commonsense reasoning has a number of properties that distinguish it from traditional logical reasoning, and that have inspired various extensions to the logical approach. Most notably, commonsense knowledge is largely defeasible and context-sensitive. People have no problem believing that "birds can fly," even though they know that "penguins are birds who cannot fly", and that "birds with broken wings cannot fly."

One explanation for why human reasoning is defeasible and context-sensitive is Lakoff and Johnson's prototype theory of human categorization (1980), which argues that people define categories based on its most central and characteristic (and often caracaturistic) example or prototype. Whereas ontologists would like to define all members in a category as being equal in membership, define a category as having sharp boundaries, and define membership by a set of common features, empirical work on human categorization shows this to not be true. People confer membership in a category to varying extents, often draw fuzzy boundaries for a category, and group members into a category based on how members resemble each other. Fehr & Russell's empirical study of common sense emotions reveals the extent that prototypes play in our categorization of emotions (1984). Logical notation is rigorous, and is completely amenable to ontologies, strict boundaries, and clean definitions, but has trouble with the inexactness and highly nuanced nature of human prototype categorization.

In addition, whereas logical reasoning is deductive, human reasoning is largely inductive, abductive, and empirical, where (over-)generalizations from known experiences plays a prominent role. One reason why humans are such successful inductive reasoners is that we collect a very large number of features in observing objects and events, thus providing a wealth of information from which patterns can emerge. Whereas the logical tradition excels at deductive reasoning, it has had much difficulty formalizing induction, and attempts to do so have generally involved great complexity, *cf.* (Flach, 1995). While logic is a highly precise and expressive language, it has difficulty modeling the imprecise way that human categorize and compare things based on prototypes, and also difficulty emulating human reasoning which is largely inductive and highly associative.

3.3 Natural language as a lingua for common sense

Such concerns led us to consider using natural language expressions as more central components of a commonsense knowledge representation. In our work with ConceptNet, we are exploring natural language as a lingua for representing common sense. In many ways, natural language fragments are a much more flexible and accessible representation for common sense. Whereas logical symbols have no a priori meaning outside of their immediate definition, words and expressions automatically inherit meaning from the way they used in our culture. Because we

don't have to first read off the axioms in which they play part to interpret word symbols, the representation becomes much simpler to author and inspect. Consider that Cyc employs logicians to author its knowledge, while ConceptNet was automatically mined from a knowledge base of English sentences contributed by 14,000 people over the web, many of whom have no computer science background. Cyc, posed in logical notation, does indeed contain very high quality knowledge. Still, accessibility and lowered complexity of authoring is a great boon, because it enables new kinds of knowledge acquisition, like the OMCS web site. It may be appropriate to give the caveat at this point in the paper that unlike logic, natural language and thus, ConceptNet's reasoning, will not have a complete formal semantics, evading exactness and absolute truths.

By posing common sense in natural language, we can benefit from the implicit conceptual framework of human language. For example, WordNet is a lexical-conceptual framework which gives us the subsumption relationships between words. This lexical hierarchy allows us to heuristically infer the related between two nodes. For example, in ConceptNet, "buy food" and "purchase groceries" are two intrinsically similar nodes. Computational resources like WordNet, Longman's Dictionary of Contempory English (LDOCE), Beth Levin's English Verb Classes (1993), and FrameNet (Fillmore & Baker, 2001) reveal the various synonym and subsumption relationships between "buy" and "purchase" and between "food" and "groceries." Allowing us to quantify the semantic similarity between two nodes or symbols affords us the ability to reason inductively over concepts, and almost by the very nature of representing knowledge in natural language, we are categorizing objects and events like people do. Whereas logic is a *synthetic representation*, creating and manipulating symbols in a closed world, natural language is an *empirical representation*, juggling concepts that are already defined and related in a human language.

Ambiguity is a particular aspect that needs to be dealt with when reasoning in natural language. In logic, it is approached as something negative, to be eradicated. But human language is full of ambiguity, and perhaps ambiguity is one of the greatest strengths to human language. Remember that the reason why there is ambiguity in words is because we have such a wealth of knowledge backing various interpretations of that word, and having this background knowledge for free is hardly a bad situation. What *is* necessary is a way to bound and manage the ambiguity of natural language fragments so that unambiguous computer algorithms can manipulate them in a systematic way. In the next section, we will present a methodology which prescribes ways of managing the ambiguity of concepts represented in natural language.

Natural language has its weaknesses as a representation for common sense knowledge. Whereas a logical symbol is concise, there may exist many different natural language fragments which mean essentially the same thing, and may seem equally suitable to include, and so in this sense logic can be seen as more economical. It is also sometimes more difficult to be precise in natural language. For example, what is the precise color of a "red apple?" In logic, we might be able to formally represent the range in the color spectrum corresponding to a "red apple," but in natural language, the word "red" is imprecise and has various interpretations.

Consider the differing colors which map to "red apple" versus "red wine" versus "red hair." WordNet has tried to address this issue of *semantic leakage* by imposing boundaries on word called *word senses.* In many cases, such boundaries are very clear, as in the case of homonyms (*e.g.* river *bank* versus financial *bank*), but in the case of more *systematic polysemies* (*e.g.* WordNet has different senses for a short *sleep* versus a long *sleep*), it is clear that such boundaries are artificial.

Thus far, we have argued that representing common sense in natural language is a good idea because the implicit conceptual framework of human language makes nodes and symbols meaningful by default, gives us a way to quantify the similarity of nodes and symbols, and is thus more amenable to inductive reasoning. We have argued that natural language is a more accessible representation for authoring. In addition to these points, we would like to add that natural language as a representation is highly desirable when the goal of the application is to reason over text. One of the primary applications of commonsense knowledge bases is to draw inferences about text. Using a logical framework of commonsense like Cyc to reason about text is quite complex. Text, which is inherently ambiguous, must first be mapped into Cyc's unambiguous logic, which is often problematic. There must be rules to map every variety of textual expression into an ontology of logical concepts, and this generally requires a very deep (and very hard to build) semantic parser. By maintaining a natural language knowledge representation, we can more readily reason about text, requiring only a surface parse of the text to extract all the contained concepts. Now all the diverse ways of expressing the same concept come in handy for concept recognition.

We now move on to the next section, where we present a methodology for reasoning over semi-structured natural language fragments. This will flesh out some of the discussion in this section.

4 Methodology for Reasoning over Natural Language Concepts

In this section, we present some methodology for reasoning over semi-structured natural language fragments used in ConceptNet. All natural language concepts in ConceptNet possess a rich *intrinsic semantics*, based on the meaning they inherit from human language. For example, creating the concept "fast car," and accepting the caveat that we consider chiefly the most common interpretation, we instantly know (tempered with an uncertainly model) that this is a type of "fast vehicle," a fast form of transportation, a car with a speed, a concept that bears family resemblance to "slow car" and to "fast bullet," and a myriad of other inferences could be made from the very certain to wildly speculative. If we can tolerate the interpretational ambiguity by taking the most common interpretation, we can situate the concept within the conceptual framework of human language and see how the concept bears similarities to other concepts in numerous ways – similarities that can be computed using computational resources. The following subsections present methodology for computing pair-wise similarities between concepts, and flexible inferencing.

4.1 Computing conceptual similarity

The basic premise of concepts as natural language fragments is that their interpretation is situated within the conceptual framework of language. For each concept, there is a way to decompose that concept into first-order atomic concepts by applying a surface parse. For the structure of concepts in ConceptNet, please refer to the syntactic grammar of concepts given in Table 1. The first-order atomic concepts will consist of simple noun phrases (note that the grammatical class "Noun Phrases" from Table 1 contains both simple and compound members), simple prepositional phrases, simple attributes, and simple verbs.

To compute the meaning of a concept, the concept is first decomposed into first-order atomic concepts, while preserving the dependence relationships between the concepts. For example: "buy good cheese" decomposes into "buy," "good," and "cheese" where "good" is an attribute of "cheese" and "cheese" plays the thematic patient role to "buy." Note that this is only a surface parse. Liu's Bubble Lexicon (2003) would support deeper parses and more highly nuanced interpretations (*e.g.* "fast car" can be interpreted variously with "fast" describing the car's top speed, the car's speed of motion, the speeding rating of the car's tires, et cetera), but this is left for future work. After a concept is decomposed, each atom is situated within the conceptual frameworks of WordNet, Longman's Dictionary of Contempory English (LDOCE), Beth Levin's English Verb Classes, and FrameNet. We chose to maintain these multiple representations because we are concerned that the inferential distance within any single resource will be overly biased. For example, the atoms "good" and "cheese" and "buy" are mapped onto the lexical entry for cheese in WordNet and LDOCE, and the verb "buy" is mapped into the lexicons of Levin's Verb Classes, and FrameNet. The meaning of a concept is represented as a collection of pointers from the decomposed concept's atoms into each of the semantic resources of WordNet, LDOCE, Levin Verb Classes, and FrameNet.

To compute the similarity of two concepts, we produce a heuristic score by comparing corresponding atoms (verb matching verb, noun modifier matching noun modifier, etc.) of the two decomposed concepts using each of the semantic resources. First, within each resource, a similarity score is produced for each pair of corresponding atoms. WordNet, LDOCE, and FrameNet's inheritance structure for verbs can be structured as semantic networks in which inferential distance is given to be proportional to the number of hops away (we heuristically weight isA and synonym links differently). For example, in WordNet's subsumption hierarchy, the inferential distance between "apple" and "food" is proportional to 3, because "apple" isA "edible fruit" isA "produce" isA "food". LDOCE also gives morphological relationships, which helps with action/state variations such as "relax" versus "feel relaxation." A similarity score should be inversely proportional to inference distance. In Levin's Verb Classes, each verb belongs to multiple alternation classes. Inferential distance here is proportional to the percentage of alternation classes shared. The weighted sum of the similarity scores is produced for each atom using each of the resources is taken. Weights on each semantic resource should be proportional to the predictive accuracy of that resource. Weights on

atoms should be proportional to the relative importance of the different atom types. For example, a noun modifier is generally not as important as the noun it modifies.

The particular coefficients used in heuristic similarity vary with different semantic resources, and change depending on the context of the reasoning task. Some similarity percentages of concepts are computed in ConceptNet as given in Table 3. (Assuming default importance weights on verbs, modifiers, noun phrases, and prepositional phrases.)

Table 3. Some pairwise similarities in ConceptNet

"apple" ~ "red apple" (76%)	"buy food" ~ "purchase groceries" (69%)
"big dog" ~ "animal" (53%)	"relax" ~ "feel relaxation" (72%)
"red" ~ "red apple" (36%)	"have accident" ~ "get into accident" (64%)

Of course computing conceptual similarity using lexical inferential distance is very difficult, as demonstrated in Table 3. Without additional insight into how a concept is generally interpreted by default (which would require a difficult, deep parse), we can only make heuristic approximations as to the relative contributions of the verb, noun phrase, attribute, and prepositional phrase to the meaning of a concept. In future work, we hope to further exploit knowledge in WordNet glosses and FrameNet frames to further nuance similarity scoring. However, our knowledge-based scoring of concepts based on inferential distance already goes beyond some previous work in reconciliation of natural language fragments, such as notably, William Cohen's WHIRL system (2000), which uses TF-IDF, a statistical vector similarity metric.

4.2 Flexible Inference

One of the strengths of representing concepts in natural language is the ability to add flexibility and fuzziness to improve inference. We again give the caveat that inferences in semantic networks are not logical deductions like in Cyc, but rather are based on graph reasoning methods like spreading activation (Collins & Loftus, 1975), structure mapping (Gentner, 1983), and network traversal. Graph-based reasoning is associative and thus not as expressive, exact, or certain as logical inferences, but it is much more straightforward to perform, and useful for reasoning practically over text. In this section, we demonstrate three kinds of flexible inference in ConceptNet: context finding, inference chaining, and conceptual analogy.

Context finding. One task useful across many textual reasoning applications is determining the context around a concept, or around the intersection of several concepts. The GetContext() feature in the API makes this easy. For example, computing the top ten concepts in the contextual neighborhood of "go to bed" yields "take off clothes," "go to sleep," "sleep," "lie down," "lay down," "close eye," "turn off light," "dream," "brush tooth," and "snore."

The contextual neighborhood around a node is found by performing spreading activation from that source node, radiating outwardly to include other concepts. The relatedness of any particular node is not just a function of the number of links away it is, but also considers how many paths there are from that node to the source node, and the directionality of the edge. In addition, pairwise similarity of nodes indicates the mutual information between the two nodes, allowing similar nodes to be aggregated, leading to a more accurate estimation of contextual neighborhood. For example, in the above example, the co-presence of "sleep" with "go to sleep" and "lay down" with "lie down" mutually promote each other higher up the list of relevant concepts.

How has GetContext() been applied for practical commonsense reasoning? Musa *et al.*'s GloBuddy system (2003) is a dynamic Berlitz phrase book that uses ConceptNet's GetContext() feature to generate a topical collection of phrases paired with their translations. For example, entering "restaurant" would return phrases like "order food" and "waiter" and "menu," and their translations into the target language. Now suppose we feed in all the extracted concepts in a particular passage of text into GetContext() and take their intersection. GetContext() used in this way serves as a "topic spotter" of sorts. Eagle *et al.* (2003) used ConceptNet and this method to gist conversation topics from overheard conversations.

Inference chaining. Another basic type of inference that can be done on a graph is building inference chains: Traversing the graph from one node to another node via some path of connectedness. This is not logical inference *per se* but a simplification of modus ponens transitive reasoning. The FindPathsBetween() feature in the ConceptNet Practical Reasoning API supports building inference chains. Temporal and spatial chains are particularly good examples. For example, ConceptNet can generate all the temporal chains between "buy food" and "fall asleep." One chain may be: "buy food" → "have food" → "eat food" → "feel full" → "feel sleepy" → "fall asleep." Each of these chains can be seen as being akin to a "script." Being able to compute the pairwise conceptual similarity is particularly crucial to the robustness of inference chaining, because it makes these chains "fuzzy." Suppose that we started with "buy steak" instead of "buy food," and suppose there is no temporal knowledge about what happens after "buy steak." By knowing that "buy steak" is a special case of "buy food," since "food" subsumes "steak," we can now make the inference "fall asleep."

Liu *et al.*'s Emotus Ponens (2003) system performs affective text classification using a slightly different representation than ConceptNet. It uses essentially inference chaining for assessing the affect of a concept. Consider that a small subset of the concepts in ConceptNet are first affectively classified into one of six affect categories (happy, sad, angry, fearful, disgusted, surprised). The affect of any unclassified concept can be assessed by finding all the paths which lead to each of these six affectively known categories, and then judging the strength and frequency of each set of paths. This is the graph-based equivalent of a *k-nearest-neighbor* classifier.

Conceptual analogy. A third practical textual inference task is finding concepts which are structurally analogous. In the ConceptNet Practical Reasoning API, there

is a GetAnalogousConcepts() feature that returns a list of structurally analogous concepts given a source concept. Structural analogy is not just a measure of semantic distance. For example, "wedding" and "bride" are semantically close but structurally unlike. Structurally, a "funeral" is much more like a "wedding." Here is another example. Typing "couch" into the GetAnalogousConcepts(), examples of top results returned include "sofa," "chair," "bed," "seat" because they share similar properties and have similar functions. We are employing structure-mapping methods (Gentner, 1983) over the ConceptNet graph to generate these simple conceptual analogies. Just like we've done with the GetContext() feature, it is also easy to contextually bias the GetAnalogousConcepts() feature. We can prefer to see analogous concepts which fall within a particular domain (defined by another GetContext()), or by biasing the numerical weights of particular semantic relations, we can emphasize *functional similarity* versus *object attribute similarity* versus *temporal similarity.* As with context finding and inference chaining, conceptual analogy is make flexible by using computed node similarity as *glue* to prevent missed structural similarities. For example, if functionOf("massage", "feel relaxation") and functionOf("meditation", "unwind"), knowing that "feel relaxation" and "unwind" are very similar prevents GetAnalogousConcepts() from overlooking this shared property of "massage" and "meditation."

Liu *et al.* are using this same idea of finding what concepts have in common to augment the aforementioned Emotus Ponens system. The basic idea behind this augmentation is that certain kinds of structural analogy, such as concepts sharing PropertyOf's, IsA's, and UsedFor's, can be predictive of affective similarity. They hope that expanding concepts with analogous concepts can expand the coverage of the system and thus improve the performance of the affective classification.

A word on evaluation. Traditionally, it is quite difficult to produce useful standalone objective evaluations of knowledgebase quality. While we have performed some evaluative analysis over ConceptNet and written about it in (Liu & Singh, 2004a), it is often equally insightful to see evaluations of ConceptNet in the context of how they improve intelligent applications. We and others at our lab have developed a host of applications using early versions of ConceptNet. We survey these applications in (Lieberman *et al.*, 2004). These seem to be entirely new kinds of applications, in that it is difficult to imagine how they could possibly be built without making use of commonsense inferences over natural language text. Many of these projects are evaluated and we invite the reader to follow the literature if he/she is interested in these in-context evaluations of ConceptNet.

5 Conclusion

Presently the largest freely available commonsense resource, ConceptNet comes with a knowledge browser, and a preliminary set of tools to support several kinds of practical inferences over text. ConceptNet follows the easy-to-use semantic network

structure of WordNet, but incorporates a greater diversity of relations and concepts inspired by Cyc.

To maintain an easy-to-use knowledge representation, while at the same time incorporating more complex higher-order commonsense concepts and relations, we chose to represent concepts as semi-structured natural language fragments. This novel use of language as knowledge representation can very elegantly represent both first-order (*e.g.* "apple pie") and second-order concepts (*e.g.* "bake apple pie"), and unlike logical symbols, the *a priori* meaning of the words make it possible to quantify the implicit similarity of two concepts.

In this paper we presented some novel methodology for computing the pairwise similarity of concepts using a variety of lexical resources such as WordNet, LDOCE, FrameNet, and Levin Verb Classes. We showed how computing the similarities between concepts enables more flexible and robust inferences. We also looked more broadly at the knowledge representational tradeoffs between formal logic and semi-structured natural language, and concluded that the flexibility afforded by natural language made it a highly suitable representation for a system whose goal is to make practical inferences over text.

That ConceptNet is already being widely used in a number of research projects such as those surveyed in (Lieberman *et al.*, 2004) is testament to the resource's *practicality* and usefulness to researchers with no background in linguistics or commonsense reasoning. We hope that this paper has encouraged the reader to consider using ConceptNet within their own projects, and that it will spur further thinking about semi-structured natural language as a serious representation for reasoning.

Acknowledgements

We extend our thanks to the many people at the Media Lab who have used ConceptNet in their projects, especially Barbara Barry, Nathan Eagle, Henry Lieberman, and Austin Wang, and especially to the 14,000 people across the web who contributed some of their common sense to the Open Mind Common Sense web site. This work was supported by the many sponsors of the Media Lab.

References

1. Cohen, W. (2000). WHIRL: A word-based information representation language. Journal of Artificial Intelligence, 118 (163-196).
2. Collins, A. and Loftus, E. (1975). A Spreading-Activation Theory of Semantic Processing. Psychological Review, 82(6):407-428.
3. Eagle, N., Singh, P., and Pentland, A. (2003). Common sense conversations: understanding casual conversation using a common sense database. Proceedings of the Artificial Intelligence, Information Access, and Mobile Computing Workshop (IJCAI 2003).
4. Fehr, B., & Russell, J. A. (1984). Concept of emotion viewed from a prototype perspective. Journal of Experimental Psychology: General, 113, 464-486.
5. Fellbaum, C. (Ed.). (1998). WordNet: An electronic lexical database. MIT Press.

6. Fillmore, C., & Baker, C.F. (2001). Frame semantics for text understanding. Proceedings of WordNet and Other Lexical Resources Workshop, NAACL.
7. Flach, P.A. (1995). Conjectures. An inquiry concerning the logic of induction. PhD thesis, Katholieke Universiteit Brabant.
8. Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7, pp 155-170.
9. Lakoff, G. & Johnson, M. (1980). Metaphors We Live by. University of Chicago Press.
10. Lenat, D.B. (1995). CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11)
11. Levin, B. (1993). English Verb Classes and Alternations: A Preliminary Investigation. MIT Press.
12. Lieberman, H., Liu, H., Singh, P., Barry, B. (2004). Beating Some Common Sense Into Interactive Applications. To Appear in AI Magazine.
13. Liu, H. (2003). Unpacking meaning from words: A context-centered approach to computational lexicon design. In Modeling and Using Context, 4th International and Interdisciplinary Conference, Proceedings, CONTEXT 2003. pp. 218-232. LNCS. Springer.
14. Liu, H. and Lieberman, H. (2002). Robust photo retrieval using world semantics. Proceedings of LREC2002 Workshop: Using Semantics for IR, Canary Islands, 15-20
15. Liu, H., Lieberman, H., Selker, T. (2003). A Model of Textual Affect Sensing using Real-World Knowledge. In Proceedings of IUI 2003. Miami, Florida.
16. Liu, H., Singh, P. (2004a). ConceptNet: A Practical Commonsense Reasoning Toolkit. To appear in BT Technology Journal. At: http://web.media.mit.edu/~push/ConceptNet.pdf
17. McCarthy, J. (1958). Programs with Common Sense. Proceedings of the Teddington Conference on the Mechanization of Thought Processes
18. Musa, R., Scheidegger, M., Kulas, A., Anguilet, Y. (2003) GloBuddy, a Dynamic Broad Context Phrase Book. In Proceedings of CONTEXT 2003, pp. 467-474. LNCS. Springer.
19. Singh, P. et al. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. In Proceedings of ODBASE'02. LNCS. Heidelberg: Springer-Verlag