The Ontology of Commonsense Knowledge

Walid S. Saba

Knowledge and Language Engineering Group
School of Computer Science, University of Windsor, Windsor, ON N9B-3P4 Canada
saba@cs.uwindsor.ca

Abstract. It is by now widely accepted that a number of tasks in natural language understanding (NLU) require the storage of and reasoning with a vast amount of background (commonsense) knowledge. While several efforts have been made to build such ontologies, a consensus on a scientific methodology for ontological design is yet to emerge. In this paper we suggest an approach to building a commonsense ontology for language understanding using language itself as a design guide. The idea is rooted in Frege's conception of compositional semantics and is related to the idea of type inferences in strongly-typed, polymorphic programming languages. The method proposed seems to (*i*) resolve the problem of multiple inheritance; (*ii*) suggest an explanation for polysemy and metaphor; and (*iii*) provide a step towards establishing a systematic approach to ontological design.

1 Introduction

Recent work in natural language understanding (NLU) seems to be slowly embracing what we like to call the 'understanding as reasoning' paradigm, as it is quite clear by now that understanding natural language is, for the most part, a commonsense reasoning process at the pragmatic level, for example in such tasks as reference resolution, plan recognition, lexical disambiguation, prepositional phrase attachments, temporal coherence, and the resolution of quantifier scope ambiguities. For instance, consider the resolution of 'He' in the following:

John shot a policeman. He immediately
a) *fled away.* (1)
b) *fell down.*

It is quite difficult to imagine how children effortlessly resolve such references, if not by recourse to the commonsense facts that, typically, when *shot*(x,y) holds between some x and some y, x is the more likely subject to flee and y is the more likely subject to fall down. Other examples of commonsense reasoning in language understanding involve the resolution of quantifier scope ambiguities. Consider the following:

$$\left\{\begin{array}{c} \textit{every} \\ \textit{few} \\ \textit{two} \end{array}\right\} \textit{graduate student(s) at MIT submitted a paper to ACL'99} \qquad (2)$$

We argue that the plausibility of wide scope *a* (implying a single paper) increases as the number of students involved in the relation decreases. Lacking a syntactic or a semantic explanation, this inference must be a function of our commonsense knowledge of how the 'submit' relation between students and 'papers' is typically manifested in the real world. Specifically, this inference is based on our commonsense belief that the *submit* relation between a *student* and a *paper* is typically $[1..m]$-to-1, where m is some small number. Moreover, different individuals seem to have a slightly different value for m, which is consistent with the findings of Kurtzman and MacDonald (1993) that different individuals seem to have different scope preferences in the same textual context.[1]

The 'understanding as reasoning' paradigm is certainly not entirely new in NLU research. Within the AI community, this paradigm was implicitly embraced by a number of authors (e.g., see Charniak, 1986; Hirst, 1986; Wilks, 1975; Schank, 1982). Unfortunately, however, these approaches were largely based on *ad hoc* algorithms built on top of *informal* knowledge representations. Due to the lack of formality, these procedures were hopelessly unscalable, and scalability was for the most part attempted by pushing the problem from the procedures to the data; which consequently led to the so-called *knowledge bottleneck*. The lack of progress in solving the *knowledge bottleneck* problem generally led AI researchers to either abandon inferential and knowledge-based approaches in favor of more quantitative approaches (e.g., Charniak, 1993), or to focus almost exclusively on the development of large commonsense knowledge bases (e.g., Lenet and Ghua, 1990). Within linguistics and formal semantics, one the other hand, little or no attention was paid to the issue of commonsense reasoning at the pragmatic level. Indeed, the prevailing wisdom (which might be partly due to lack of progress in AI-based NLU) was that a number of NLU tasks require the storage of and reasoning with a vast amount of background knowledge (van Deemter, 1996), an opinion that led some (e.g., Reinhart, 1997) to conclude that pragmatic approaches are 'highly undecidable'.

In our view both trends were partly misguided. In particular, we hold the view that while language understanding is for the most part a commonsense reasoning process at the pragmatic level, this reasoning process and the underlying knowledge structures that it utilizes must be formalized if we ever hope to build scalable systems. In this light we believe the work on integrating logical and commonsense reasoning in language understanding (e.g., Allen, 1987; Pereira & Pollack, 1991; Zadrozny & Jensen, 1991; Hobbs, 1985; Hobbs *et al.*, 1993; and more recently Asher & Lascarides, 1998; and Saba & Corriveua, 1997) is of paramount importance.

---

[1] An inferencing strategy that models individual preferences in the resolution of scope ambiguities at the pragmatic level has been suggested in (Saba and Corriveau, 2001).

Much of this work is directed towards formulating commonsense inferencing strategies to resolve a number of ambiguities at the pragmatic level. Although it has been shown (see Saba & Corriveau, 2001) that these inferences do not always require the storage of and reasoning with a vast amount of background knowledge, it is clear that a number of tasks do require such a knowledgebase. Indeed, substantial effort has been made towards building ontologies of commonsense knowledge (e.g., Lenat & Ghua, 1990; Mahesh & Nirenburg, 1995; Sowa, 1995), and a number of promising trends that advocate ontological design based on sound linguistic and logical foundations have started to emerge in recent years (e.g., Guarino & Welty, 2000; Pustejovsky, 2001). However, a systematic and objective approach to ontological design is still lacking. In particular, we believe that an ontology for commonsense knowledge must be *discovered* rather than *invented*, and thus it is not sufficient to establish some principles for ontological design, but that a strategy by which a commonsense ontology might be **systematically** and **objectively** designed must be developed. In this paper we propose such a strategy.

2 Language Use as Guide to Ontological Design

Our basic strategy for designing an ontology of commonsense knowledge is rooted in Frege's conception of Compositionality. According to Frege (see Dummett, 1981, pp. 4-7), the sense of any given sentence is derived from our previous knowledge of the senses of the words that compose it, together with our observation of the way in which they are combined in that sentence. The cornerstone of this paradigm, however, is an observation that has not been fully appreciated regarding the manner in which words are supposed to acquire a sense. In particular, the principle of Compositionality is rooted in the thesis that "our understanding of [those] words consists in our grasp of the way in which they may figure in sentences in general, and how, in general, they combine to determine the truth-conditions of those sentences." (Dummett, 1981, pp. 5).

This simple idea forms the basis of our strategy in designing an ontology for commonsense knowledge: what language allows one to say about a concept, tells us a lot about the concept under consideration. In other words, the meanings of words (i.e., the concepts), can be discovered from the manner in which the words are **used** in everyday language. As Bateman (1995) has suggested, *language* is the best-known theory on everyday knowledge. Assuming that language reflects thought, therefore, analyzing patterns of everyday language 'use' should provide useful clues to the structure of commonsense knowledge. As a motivating example, consider the nouns *table* and *elephant*, and the adjectives *smart* and *large*, out of which four syntactically well formed and semantically valid adjective-noun combinations can be made. One of these combinations, namely *smart table,* is typically rejected on pragmatic grounds, as it is at odds with our commonsense view of what tables are[2]. In particular, while it is sensible to say *large elephant* and *large table*, a *table* is not the kind of

[2] For the moment we are not concerned with metaphor.

object for which *smart* applies. This analysis results in the fragment hierarchy shown in figure in 1 below. Note that this kind of analysis is not much different from the type inferencing process that occurs in strongly typed, polymorphic programming languages[3].

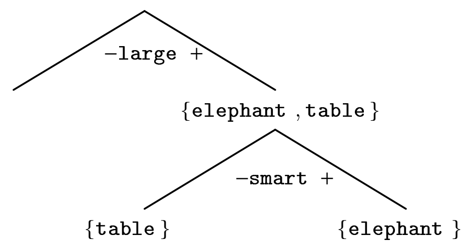


Figure 1. A simple analysis of four adjective-noun combinations.

For example, consider the linguistic patterns and the corresponding type inferences shown in table 1. From $x + 3$, for example, one can infer that x is a `number` since numbers are the "kinds of things" that can be added to 3. In general, the most generic type possible is inferred (i.e., these operations are assumed to be polymorphic).

Linguistic Pattern	Type Inference
$x + 3$	x is `number`
`reverse`(x)	x is a `sequence`
`insert`(x,y)	x is an `object`; y is `sequence` of x objects
`head`(x)	x is a `sequence`
`even`(x)	x is `number`

Table 1. Linguistic patterns and the corresponding type inferences.

For example, all that can be inferred from *reverse*(x) is that x is the generic type `sequence`, which could be a `list`, a `string` (a sequence of `characters`), a `vector`, etc. Note also that in addition to actions (methods), properties (truth-valued functions) can also be used to infer the type of an object. For example, from *even*(x) one can infer that x is a `number`, since lists, sequences, etc. are not the kinds of objects which can be described by the predicate *even*. In general, given a domain (a set of concepts) $C = \{c_1, \ldots, c_m\}$ and a set of actions (and properties) $P = \{p_1, \ldots, p_n\}$, a predicate $app(p,c)$ where $c \in C$ and $p \in P$ can be defined such that the action (or property) p applies to (or makes sense of) objects of type c. For each property $p \in P$, a set $C_p = \{c \mid app(p,c)\}$, denoting all concepts c for which the property p is applicable, can be generated. A concept hierarchy is then systematically discovered by

[3] This is reminiscent of (Montague, 1970): "I reject the contention that there is an important theoretical difference between formal and natural languages".

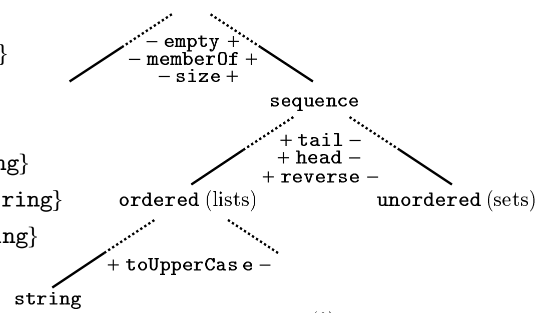
analyzing the subset relationship among the various sets generated. To illustrate how this type of analysis could (systematically) yield a type hierarchy, consider a set of concepts C and a set P of properties (or actions) that may or may not be sensibly applied to concepts in C:

$$C = \{\texttt{list}, \texttt{string}, \texttt{set}\}$$
$$P = \{\texttt{empty}, \texttt{memberOf}, \texttt{size}, \texttt{tail}, \texttt{head}, \texttt{reverse}, \texttt{toUpperCas e}\}$$

Shown in figure 2 below is a number of sets that are generated by $app(p,c)$ (figure 2a) and the concept hierarchy implied by the subset relationship among these sets (figure 2b). Note that each (unique) set corresponds to a class in the hierarchy. Equal sets (e.g. $C_{\texttt{tail}}$ and $C_{\texttt{head}}$) correspond to the same class. A class could be given any meaningful label that intuitively represents all the concepts in the class. For example, in figure 2b `sequence` is used to collectively refer to sets, strings and lists.

$$C_{\texttt{reverse}} = \{\texttt{list}, \texttt{string}\}$$
$$C_{\texttt{tail}} = \{\texttt{list}, \texttt{string}\}$$
$$C_{\texttt{head}} = \{\texttt{list}, \texttt{string}\}$$
$$C_{\texttt{size}} = \{\texttt{list}, \texttt{set}, \texttt{string}\}$$
$$C_{\texttt{memberOf}} = \{\texttt{list}, \texttt{set}, \texttt{string}\}$$
$$C_{\texttt{empty}} = \{\texttt{list}, \texttt{set}, \texttt{string}\}$$
$$C_{\texttt{toUpperCase}} = \{\texttt{string}\}$$

(a)



(b)

Figure 2. Sets generated by $app(p,c)$ and the hierarchical structure implied by them.

Clearly, there are a number of rules that can be established from the concept hierarchy shown in figure 2. For example, one can state the following:

$$(\forall c)(app(reverse,c) \supset app(size,c)) \tag{3}$$
$$(\exists c)(app(size,c) \wedge \neg app(reverse,c)) \tag{4}$$
$$(\forall c)(app(tail,c) \equiv app(head,c)) \tag{5}$$

Here (3) states that whenever *it makes sense* to *reverse* an object *c*, then it also makes sense to ask for the *size* of *c*. This essentially means that an object to which the *size* operation can be applied must be a parent of an object to which the *reverse* operation can be applied. (4), on the other hand, states that there are objects for which the *size* operation applies, but for which the *reverse* operation does not apply. Finally, (5) states that whenever it makes sense to ask for the *head* of an object then it also makes sense to ask for its *tail*, and vice versa. Thus while there must be at least one property that defines a concept, there could be many (we will have more to say about this below.) Finally, it must be noted that in performing this analysis we have assumed that the predicate $app(p,c)$ is a Boolean-valued function, which has the consequence that the type hierarchy is a strict binary tree. In fact, this is one of the main characteristics of our method, and has led to two important results: (*i*) multiple inheritance is completely avoided; and (*ii*) by not allowing any ambiguity in the interpretation of $app(p,c)$, lexical ambiguity, polysemy and metaphor are explicitly represented in the hierarchy. This will be discussed below.

3 Language and Commonsense Knowledge

The work described here was motivated by the following two assumptions: (*i*) the process of language understanding is for the most part a commonsense reasoning process at the pragmatic level; and (*ii*) since children master spoken language at a very young age, children must be performing commonsense reasoning at the pragmatic level, and consequently, they must posses all the commonsense knowledge required to understand spoken language[4]. In other words, we are assuming that deciding on a particular $app(v,c)$ should not be controversial, and that children can easily and consistently answer simple questions such as *do elephants fly*, *do mountains talk*, *do books run*, etc. Note that in answering these questions it is clear that one has to be coconscious of metaphor. For example, while tables, people, and feelings can be strong (i.e., it is quite meaningful to say *strong table, strong person, strong feeling*), it is clear that the senses of *strong* in these three cases are quite distinct. In fact, the various metaphorical derivations of a lexeme are eventually discovered by the process we describe here, as will become evident in the next sections. The point here is that all that matters, initially, is to consider posing queries such as $app(smart,elephant)$ to a five-year old. Furthermore, in asking such a query we are not asking whether or not every elephant is smart, nor how smart elephants can be, but whether or not it is meaningful to say 'smart elephant'. We believe that such queries are binary-valued. In other words, while at the quantitative (or a data-level) it could be a matter of degree as to how smart a specific elephant might be, for example, the qualitative question of whether or not it is meaningful to say 'smart elephant' is not a matter of degree[5]. With this in mind, our

[4] It may very well be the case that "everything we know we learned in kindergarten"!

[5] We will not dwell on this issue too much here except to say that as Elkan (1993) has convincingly argued, to avoid certain contradictions logical reasoning must at some level collapse to a binary logic. While Elkan's argument seemed to be susceptible to some criticism (e.g., Dubois *et al.* (1994)), there are more convincing arguments supporting the same result. For example, consider the following:

(1) *John likes every famous actress*
(2) *Liz is a famous actress*
(3) *John likes Liz*

Clearly, (1) and (2) should entail (3), regardless of how famous Liz actually is. Using basic approach to discovering the ontology of commonsense knowledge can be summarized as follows:

- Select a set of adjectives and verbs, $V = \{v_1, \ldots, v_m\}$.
- Select a set of nouns $C = \{c_1, \ldots, c_n\}$.
- Generate sets $C_i = \{c \in C \mid app(v_i, c)\}, 1 \le i \le m$ for every $v_i \in V$
- Analyse the subset relationship between all sets $C_i \in \{C_1, \ldots, C_m\}$

As an initial example, consider the set of verbs V = {`move`, `walk`, `run`, `talk`, `reason`} and the set of nouns C = {`Rational`, `Bird`, `Elephant`, `Shark`, `Animal`, `Ameba`}. By repeatedly applying *app*(*v*,*c*) the following sets are generated:

$C_{\texttt{move}} = \{\texttt{Rational}, \texttt{Animal}, \texttt{Bird}, \texttt{Elephant}, \texttt{Shark}, \texttt{Ameba}\}$
$C_{\texttt{talk}} = \{\texttt{Rational}\}$
$C_{\texttt{reason}} = \{\texttt{Rational}\}$
$C_{\texttt{think}} = \{\texttt{Animal}\}$
$C_{\texttt{walk}} = \{\texttt{Rational}, \texttt{Bird}, \texttt{Elephant}\}$
$C_{\texttt{run}} = \{\texttt{Rational}, \texttt{Bird}, \texttt{Elephant}\}$

First we note that while some decisions could 'technically' be questioned (say by a biologist), our strategy was to simply consider the question from the point of view of commonsense. In deciding on a particular *app*(*v*,*c*) we considered the query poised to a five-year old: do elephants fly, do they run, do they talk, etc. Questionable situations were simply ignored. This initial process resulted in the hierarchy shown in figure 3 below. Some of the sets indicating positive left and right attributes are given in figure 4 below. Note that some powerful inferential patterns that can be used in language processing are implicit in the structure shown in figure 3. For example, what does not think does not hurt (L_7), what walks also runs (L_8), anything that lives evolves (L_3 and L_4), etc.

4 Knowledge-Level: Being *Locked* to a Property

Motivating our approach to *discovering* (as opposed to *inventing*) the structure of commonsense knowledge is the idea that common language must reflect commonsense, and thus, differences and similarities between concepts must be reflected in our everyday conversation. Technically, the

---

any quantitative model (such as fuzzy logic), this intuitive entailment cannot be produced (we leave the details of formulating this in fuzzy logic as an exercise!) The problem here is that at the qualitative level the truth-value of *famous*(x) must collapse to either true or false, since at that level all that matters is whether or not Liz is famous, not how famous she actually is.

claim we are making here can be stated as follows: every concept at the knowledge- (or commonsense-) level must 'own' some unique property, and this must also be linguistically reflected by some verb or adjective. This might be similar to what Fodor (1998, p. 126) meant by "having a concept is being locked to a property." In fact, it seems that this is one way to test the demarcation line between commonsense and domain specific knowledge. In particular, it seems that domain-specific concepts are those concepts that are not uniquely locked to any word in the language.

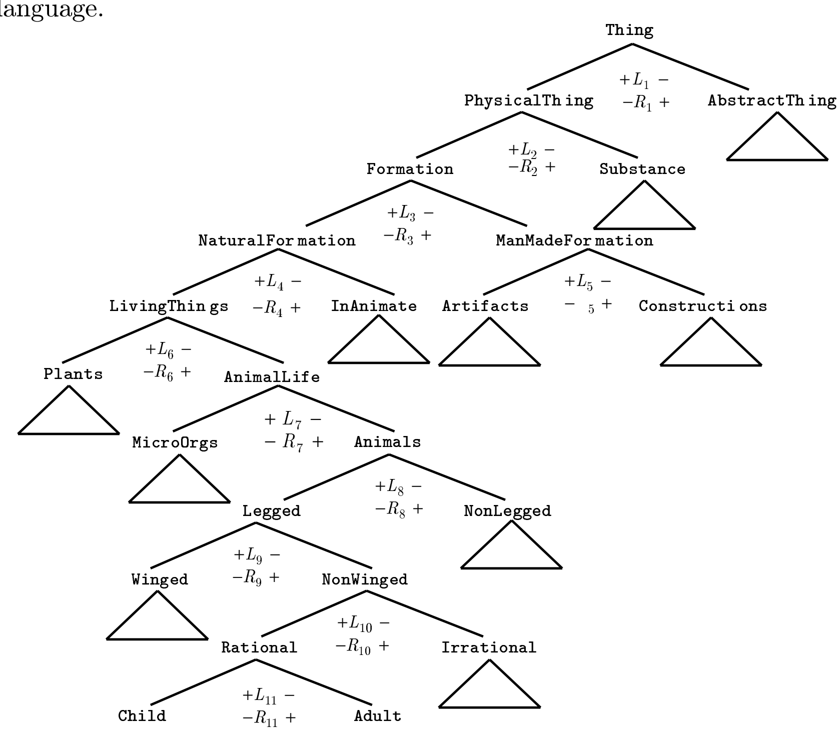


Figure 3. An adult is a physical, living thing that is formed. It evolves, it grows, it develops, moves, it can walk, run, hear, see, talk, think, reason, etc.

$$+L_2- = \{develop, form\} \qquad +L_3- = \{evolve\}$$
$$+L_4- = \{live, die, born, grow\} \qquad +L_6- = \{branch\}$$
$$-R_6+ = \{move\} \qquad -R_7+ = \{sleep, rest, eat, digest, bleed, hurt, think\}$$
$$+L_8- = \{sit, jump, walk, run\} \qquad -R_9+ = \{roar, cry\}$$
$$+L_{10}- = \{talk\} \qquad -R_{11}+ = \{reason\}$$

Figure 4. Sets shown in figure 3.

To illustrate this point further let us analyse a fragment of the hierarchy shown in figure 3 in some detail. The concept `LivingThing`, one can argue, is the concept for which one can say the following:

app(`LivingThing`,`Grow`)
app(`LivingThing`,`Develop`)
app(`LivingThing`,`Live`)
app(`LivingThing`,`Die`)

That is, living things grow, develop, live and die (the concept `LivingThing` is actually 'locked' to several other properties, and thus to several other verbs and adjectives.) What is important to note here is that in order to classify `LivingThing` further one must find some conceptual difference between all living things; a difference which must be somehow reflected in language. As it happens, the word 'move' could be used on all living things except plants. One could, therefore, suggest a classification based on that verb, as shown in figure 3.

Thus all living things are either plant-like things (`Plants`), which are the immobile living organisms, or animal-like things (`AnimalLife`), which are the mobile living organisms. We are now again faced with the same situation, namely further classification of `Plants` and `AnimalLife`. This in turn requires us to find some conceptual difference between potential subtypes of these two concepts, which must translate into a difference in ordinary language. This process is repeated until no further classification based on linguistic differences is possible. In figure 3, for example, `Winged`, which is the class of animals that are legged and have wings was introduced based on the property of *flying*, a difference that translates in language to the lexeme 'fly'. While a biologist can list numerous differences between all birds that fly, classification on such grounds is domain-specific, and does not belong to the knowledge (or commonsense) level, since there does not seem to be a linguistic difference between all birds that fly. This must be, therefore, the knowledge-level, and any further classification beyond this level is based on domain-specific knowledge.

5 Polysemy and Metaphor

In our approach the occurrence of a verb/adjective at any place and at any level in the hierarchy always refers to a *unique sense* of that verb/adjective. Therefore one expects similar senses of a lexeme to apply to concepts along the same path, albeit at different levels in the hierarchy. In particular, one would expect that highly ambiguous verbs to apply to concepts higher-up in the hierarchy, where various similar senses of a verb v should end-up applying at various levels below v.

Consider for example `form` and `formulate`, in the sense of forming and formulating ideas. Since our method is based on the idea of using such verbs to discover the nature of concepts, `form` and `formulate` must both apply to ideas. Note that if everything that can be 'formed' can be 'formulated' and vice versa, then these two verbs would be synonymous.

However, in this case this is not so, since there are things that can be formed but not formulated. For example, consider the small fragment shown in figure 5, where it is shown that 'developing', 'formulating', 'forming', etc. are all specific ways of 'making' (in other words, one sense of 'make' is 'develop'). Note the eventual split however. In particular, while we `make`, `form`, and `develop` both `ideas` and `feelings`, `ideas` are `formulated` while `feelings` are `fostered`.

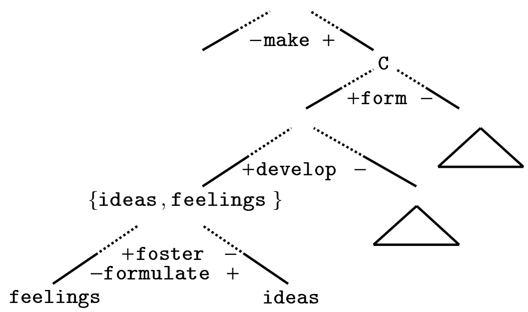


Figure 5. An explanation of polysemy.

While the occurrence of similar senses of verbs at various levels in the hierarchy indicates polysemy, the occurrence of the same verb (the same lexeme) at structurally isomorphic places in the hierarchy indicates metaphorical derivations. Consider the following:

$$app(\texttt{run},\texttt{LeggedThing}) \tag{6}$$
$$app(\texttt{run},\texttt{Machine}) \tag{7}$$
$$app(\texttt{run},\texttt{Show}) \tag{8}$$

(6) through (8) state that we can speak of a legged thing, a machine and a show running. Clearly, however, these examples involve three different senses of the verb `run`. It could be argued that the senses of `run` that are implied by (7) and (8) correspond to a metaphorical derivation of the actual running of natural kinds, the sense implied by (6). It is also interesting to note that these metaphorical derivations occur at various levels: first from natural kinds to artifacts; and then from physical to abstract. Moreover, the mass/count distinction on the physical side seems to have a mirror image of a mass/count on the abstract side. For example, note the following similarity between `water` (physical substance) and `information` (abstract substance):

- `water/information` flows, can be diverted, filtered, processed, etc.
- we can be flooded by, or drown in `water/information`
- a little bit of water/information is (still) water/information

One interesting aspect of these findings is to further investigate the exact nature of this metaphorical mapping and whether the map is consistent throughout; that is, whether same-level hierarchies are structurally isomorphic, as the case appears to be so far (see figure 6)[6].

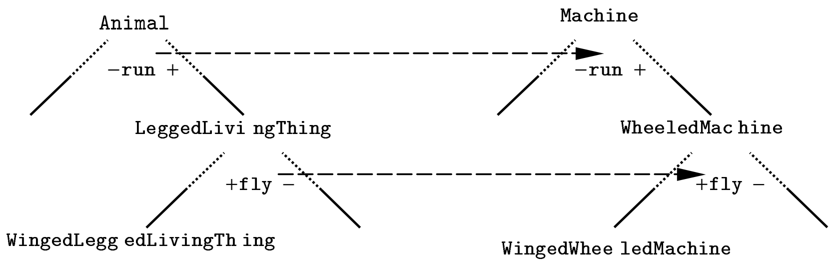


Figure 6. Isomorphic structures explaining metaphors.

6 Negation, Immutable Features and Surprise

The model proposed here allows us to have a very interesting model of the negation. To illustrate, consider the following propositions implied by the concept hierarchy given in figure 3, namely that generally animals move, and people talk:

$$\mathit{app}(\texttt{Move},\texttt{Animal}) \tag{9}$$
$$\mathit{app}(\texttt{Talk},\texttt{Rational}) \tag{10}$$

What is interesting to consider here is how one interprets the negation of such propositions. In particular, there are two possible answers to the query $\neg \mathit{app}(\texttt{Move},?\texttt{X})$, i.e., to the query "what objects do not move?" One can simply provide (`U - Animal`) as an answer, where `U` is the set of all concepts in the universe of discourse. This is the set of all concepts excluding those for which $\mathit{app}(\texttt{Move},\texttt{Animal})$ holds. Thus, plants and all non-living things do not move (see figure 7 below). This is strong negation, since it simply returns the complement with respect to the entire structure. However, we argue that there is a subtle difference between the following queries:

Do mountains talk? (11)
Do elephants talk? (12)

Although a rational agent would answer "no" in both cases, one might imagine a child replying "nah, mountains do not talk" in response to (11). This must be function of the following: elephants fall directly under the negative polarity of `talk`; while this property is not even applicable to mountains (see figure 7 below). From a Gricean point of view, it seems that "elephants do not talk" is somewhat more meaningful than "mountains do not talk." This subtle difference in the two cases of negation is crucial in performing commonsense reasoning in language

[6] Conservatively, the mapping might be a homomorphism and not an isomorphism.

understanding. This is also related to the notion of the *immutability* of a feature (Sloman *et al.*, 1998), which is thought to reflect the degree to which a concept depends on a certain feature (or, conversely, how central is a certain feature to the definition of a concept). For example, for some person p, $\neg Run(p)$ is arguably less surprising than $\neg Walk(p)$, which in turn is less surprising than $\neg Move(p)$. That is, it is much harder to assume the immobility of a person that to assume that a certain person might not be able to walk, or run. In our model such measures (including a measure of surprise) could be easily computed.

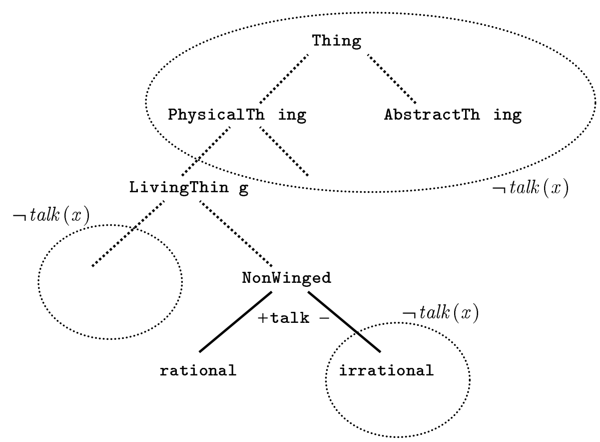


Figure 7. Mountains, elephants, and poems do not talk.

7 Reasoning with Commonsense Knowledge

What we are suggesting in this paper is a process that would hopefully lead to the *discovery* of the ontology of commonsense knowledge. This alone would clearly do little to building natural language understanding systems unless an inferencing strategy that utilizes this ontology is properly formulated. While the ontology provides the synthetic knowledge that an NLU system might need, an NLU system must clearly use quite a bit of analytic knowledge. A typical example would be the following:

$$(\forall p, c_1, c_2)(app(p, c_1) \wedge isa(c_1, c_2) \supset app(p, c_1)) \tag{13}$$

That is, any property that applies to a concept applies to all its subtypes. Clearly there are numerous other such rules that could be added. Another important observation here is that the system that will eventually emerge will yield a much richer type structure than the (flat) type systems typically assumed in formal semantics (e.g., Montague, 1973). For example, form the hierarchy in figure 3 one can clearly establish the following:

$$\texttt{Walk} : (e_{\texttt{LeggedThing}} \rightarrow t) \tag{14}$$

$$\texttt{Write} : (e_{\texttt{Human}} \rightarrow (e_{\texttt{Content}} \rightarrow t)) \tag{15}$$

That is, 'write' is not simply a relation between two entities, but a relation between two specific types of entities. Note the importance of this step (of combining formal semantics with a rich type hierarchy), however. For example, (15) states that *write*(x,y) is well-typed as long as $\texttt{isa}(x,\texttt{Human})$ and $\texttt{isa}(y,\texttt{Content})$. In general,

$$\mathit{wellTyped}(\texttt{v}(e)) \equiv_{df} \mathit{type}(\texttt{v}, (e_m \rightarrow t)) \wedge \mathit{type}(e, a) \wedge \texttt{isa}(a, m)$$

$$\mathit{wellTyped}(\texttt{v}(e_1, e_2)) \equiv_{df} \mathit{type}(\texttt{v}, (e_m \rightarrow (e_n \rightarrow t))) \wedge \mathit{type}(e_1, a) \wedge \mathit{type}(e_2, b) \wedge \texttt{isa}(a, m) \wedge \texttt{isa}(b, n)$$

More importantly, however, the combination of a rich type hierarchy and a rigorous semantics should shed some light on the semantics of compound nominals. For instance, type information might explain why removing the middle noun form (16) changes the subject considerably while the same is not true in (17).

Computer book sale (16)
Information management system (17)

Such rules are important in a variety of language processing tasks, and in particular in topic-based information retrieval. A compositional semantics that exploits a rich type hierarchy should therefore facilitate the development of a meaning algebra; for example to explain why *fake gun* is not (exactly) a gun, whereas *imported gun* is very much a gun. These are precisely the kinds of issues that have prompted this work, and much of this is currently under development.

8 Concluding Remarks

In this paper we argued for and presented a new approach to the systematic design of ontologies of commonsense knowledge. The method is based on the basic assumption that "language use" can guide the classification process. This idea is in turn rooted in Frege's principle of Compositionality and is similar to the idea of type inference in strongly-typed, polymorphic programming languages. The experiment we conducted shows this approach to be quite promising as it seems to have answered a number of questions simultaneously. In particular, the approach seems to (*i*) completely remove the need for multiple inheritance; (*ii*) provide a good model for lexical ambiguity and polysemy; and (*iii*) suggest a plausible explanation of metaphor in natural language.

Much of what we presented here is work in progress, more so than a final result. Therefore, we are well aware that it might be quite ambitious to expect this process to yield a complete classification in a strict binary tree (no multiple inheritance, and no lexical ambiguity). We must also note that a number of other aspects of this work were not discussed here, such as the part-whole relationship. In particular, it seems that some, but not all, verbs that apply to a concept apply to their parts. For example, *grow* in *app*(*grow*,*leg*) and *app*(*grow*,*arm*) is very much related to *grow* in *app*(*grow*,*person*). That is, when we refer to a person growing, aging, etc. we are indirectly referring to the growing or aging of the parts. Another important part of this work is to also discover the nature of the relationship between (genuine) types (e.g., `Human`) and roles that concepts play (e.g., `Teacher`, `Father`, etc.) In this regard a number of temporal aspects must also be formalized.

A great deal of work is still needed to formalize the entire approach as well as work out the various inference rules that will eventually be needed in a natural language understanding system. We have already successfully used some of the ideas presented here in NLU tasks, such as developing an efficient and cognitively plausible inferencing strategy to resolve quantifier scope ambiguities at the pragmatic level (see Saba & Corriveau, 2001). While our immediate goal is to discover the ontology of commonsense knowledge, our ultimate goal is to build systems that can *understand* spoken language. This task has proven to be more challenging than has ever been imagined. Turing might have had it right all along: a machine that can converse in spoken language, must be an intelligent machine!

References

Allen, J. (1987), *Natural Language Understanding*, Benjamin/Cummings, Menlo Park, CA.

Asher, N. and Lascarides, A. (1998), The Semantics & Pragmatics of Presupposition, *Journal of Semantics*, 15:239-299.

Bateman, J. A. 1995. On the Relationship Between Ontology Construction and Natural Language: a Socio-Semiotic View. *International Journal of Human-Computer Studies*, **43**: 929-944

Charniak, E. (1993), *Statistical Language Learning*, Cambridge, MA: MIT Press.

Charniak, E. (1986), A neat theory of marker passing. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 584--588, Los Altos, CA, Morgan Kaufmann.

Fuzzy logic vs. possibilistic logic, *IEEE Expert*, 9(4), pp. 15-19

Dummett, M. (1981), *Frege: Philosophy of Language*, Harvard Univ. Press, Cambridge: MA.

Elkan, C. (1993), The Paradoxical Success of Fuzzy Logic, In *Proceedings of the 11th National Conference on Artificial Intelligence,* AAAI-93, pp. 698-703. AAAI Press.

Fodor, J. A. (1998), *Concepts - where cognitive science went wrong*, Oxford University Press: NY.

Guarino, N. and Welty, C. (2000), A Formal Ontology of Properties, In *Proceedings of the 12th Int. Conf. on Knowledge Engineering and Knowledge Management, LNCS,* Verlag.

Hirst, G. (1986). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge

Hobbs, J. (1985), Ontological Promiscuity, In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 61-69, Chicago, Illinois, 1985. ACL.

Hobbs, J. R., *et al*. (1993), Interpretation as Abduction, *Artificial Intelligence*, **63**:69-142.

Kurtzman, H. and MacDonald, M. (1993), Resolution of Quantifier Scope Ambiguities, *Cognition*, 48, 243-279.

Lenat, D. B. and Guha, R.V. (1990), *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Addison-Wesley.

Mahesh, K. and Nirenburg, S. (1995), A Situated Ontology for Practical NLP, In *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, IJCAI-95, August 1995, Montreal, Canada.

Montague, R. (1973). The Proper Treatment of Quantification in Ordinary English, In Thomason (1974).

Montague, R. (1970), English as a Formal Language, In Thomason (1974).

Pereira, F. C. N. and Pollack, M. E. (1991), Incremental Interpretation, *Artificial Intelligence*, **50**:37-82.

Pustejovsky, J. (2001), Type Construction and the Logic of Concepts, In P. Bouillon and F. Busa (eds.), *The Syntax of Word Meanings*, Cambridge University Press.

Reinhart, T. (1997), Quantifier Scope: How Labor is Divided between QR and Choice Functions, *Linguistics and Philosophy*, **20**(4): 335-397

Saba, W. S. and Corriveau, J.-P. (1997), A Pragmatic Treatment of Quantification in Natural Language, In *Proceedings of the 1997 National Conference on Artificial Intelligence* (AAAI-97), pp. 610-615, Morgan Kaufmann.

Saba, W. S. and J-P. Corriveau (2001), Plausible Reasoning and the Resolution of Quantifier Scope Ambiguities, *Studia Logica* (Special Issue on Commonsense Reasoning), **67**(2):1-19

Schank, R. C. (1982), *Dynamic Memory: A Theory of Learning in Computers and People*, Cambridge University Press, New York.

Sloman, AS., Love, B. and Ahn, W-K. (1998), Feature Similarity and Conceptual Coherence, *Cognitive Science*, **22**(2):189-228.

Sowa, J. F. (1995). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Boston, MA: PWS Publishing Company.

Thomason, R. (1974), Editor with an Introduction, *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press.

van Deemter, K. (1996), Towards a Logic of Ambiguous Expressions, In K. van Deemter and S. Peters (eds.), *Semantic Ambiguity and Underspecification*, pp. 55-76, CSLI, Stanford, CA.

Wilks, Y. (1975), A Preferential, Pattern-Seeking, Semantics for Natural Language Interface, *Artificial Intelligence*, **6**:53-74.

Zadronzy, W. and Jensen K. (1991), Semantics of Paragraphs, *Computational Linguistics*, **17**(2):171-209.