Unpacking Meaning from Words: A Context-Centered Approach to Computational Lexicon Design

Hugo Liu

MIT Media Laboratory
20 Ames St., E15-320D
Cambridge, MA 02139, USA
hugo@media.mit.edu

Abstract. The traditional approach to computational lexicon design represents words as static encapsulations of purely lexical knowledge. This view imposes fundamental limitations on the ability of the lexicon to generate nuance-laden and context-sensitive meanings, because word boundaries are obstructive, and the impact of non-lexical knowledge on meaning is unaccounted for. To address these limitations, we present a context-centered approach called a Bubble Lexicon. Inspired by Ross Quillian's Semantic Memory System, we represent word-concepts as nodes on a symbolic-connectionist semantic network. In the Bubble Lexicon, a word's meaning is defined dynamically by a context bubble that is grown around a word's node, thus giving an account of systematic polysemy. Linguistic assembly tasks such as attribute attachment are also made sensitive to context, and demonstrate remarkable generative capability. In this paper, we present high-level arguments for a context-centered approach to lexicon design, and describe an implemented and evaluated Bubble Lexicon Architecture.

Keywords: Natural Language Semantics, Representing Context and Contextual Knowledge

1 Motivation

Packing meaning (semantic knowledge) into words (lexical items) has long been the knowledge representation tradition of lexical semantics. However, as the field of computational semantics becomes more mature, this paradigm is now called into question. Words, when computed as discrete and static encapsulations of meaning, cannot easily generate the range of nuance-laden and context-sensitive meanings that the human language faculty seems to be able to produce. Take one example: Miller and Fellbaum's popular machine-readable lexicon, WordNet [7], packages a small amount of dictionary-type knowledge into each *word sense*, which represents a specific meaning of a word. Word senses are partitioned *a priori*, and the lexicon does not provide an account of how senses are determined or how they may be systematically related, a phenomenon known as systematic polysemy. The result is a sometimes arbitrary partitioning of word meaning. For example, the WordNet entry for the noun form of "sleep" returns two senses, one which means "a slumber" (i.e. a long rest), and the other which means "a nap" (i.e. a brief rest). The systematic relation between these two senses is unaccounted for, and their existence as separate senses indistinguishable from homonymy gives the false impression that there is a no-man's land of meaning in between.

Trying to address the inflexibility of lexicons like WordNet, Pustejovsky's Generative Lexicon Theory (GLT) [19] attempts to pack a great deal more meaning into a word entity, including knowledge about how a word participates in various semantic roles known as "qualia," which dates back to Aristotle. The hope is that a densely packed word-entity will be able to generate a fuller range of nuance-laden meaning. In this model, the generative ability of a word is a function of the type and quantity of knowledge encoded inside that word. For example, the lexical compound "good rock" only makes sense because one of the functions encoded into "rock" is "to climb on," and associated with this function is some notion of "goodness." Like other lexicons, GLT still follows the paradigm of packaging meaning into discrete and predefined word structures. We argue, however, that this imposes two fundamental limitations on the lexicon's generative power:

1) **Artificial Word Boundary.** Because words are discrete objects with predefined meaning boundaries, lexicon designers are forced to make *a priori* thus rather arbitrary decisions about what knowledge to encode into a word, and what to leave out. This is a problem because practically, it is impossible to pack all the knowledge into a word that would be needed to anticipate all possible polysemies (multiplicity of meaning) of that word.
2) **Exclusion of non-lexical knowledge.** Given the present abstraction of a word as a predetermined, static encapsulation of meaning, it is common practice to encode only knowledge that is most directly relevant to the word, namely, *lexical knowledge.* In GLT, this lexical knowledge must fall into one of four qualia roles: formal (a narrow dictionary-typed definition), constitutive (what it is made of), telic (its functions), and agentive (how it came into being). We point out that not all relevant semantics will fall into these categories of lexical knowledge. Some knowlege arises out of larger contexts, such as general world knowledge. An illustrative example: if the proposition, "cheap apartments are rare" is considered true, then a set of entailments (its *interpretive context*) must also hold, such as, "they are rare because cheap apartments are in demand" and "these rare cheap apartments are in low supply." Although it is non-obvious how such non-lexical entailments are important to the semantic interpretation of the proposition, these contextual factors measurably nuance the interpretation of, *inter alia,* the word, "rare." The influence of such non-lexical knowledge on the word sense of "rare" is discussed extensively elsewhere [24]. Because non-lexical knowledge influences meaning, we argue that it must be accounted for in the lexicon.

We believe that these problems curtail the ability of such a lexicon to be used to produce more nuanced and context-sensitive meanings of words and compound expressions – something that the human mental lexicon seems to be capable of. This paper attempts to address these limitations by exploring a quite different approach to computational lexicon design based loosely on Ross Quillian's work on semantic memory [20], which we dub as: Bubble Lexicon.

Our approach questions the traditional representation of lexicons, namely: the word as a static encapsulation of lexical knowledge. We find encouragement from what is currently known about the human mental lexicon, where the meaning of words do not observe fixed boundaries, where knowledge is highly interconnected, and where no barriers separate lexical knowledge from non-lexical knowledge. Based on this understanding, we designed the Bubble Lexicon to eliminate boundaries around words, and those between lexical and non-lexical knowledge. A word is represented as a node on a semantic network, has no internal contents, and is simply meant as a reference point, or, *indexical feature,* as Jackendoff would call it [9]. The meaning of a word is distributed throughout the semantic neighborhood surrounding the word's reference node. And there are index nodes for more than words alone: explicit contexts and larger linguistic phrases can also have nodes. Without formal word boundaries, the "meaning" of a word becomes the dynamically chosen, flexible context bubble (hence the lexicon's name) around that word's node. The size and shape of the bubble varies according to the strength of association of knowledge and the influence of active contexts; thus, meaning is nuanced and made context-sensitive. Because both lexical knowledge and non-lexical are represented in the graph, both types of knowledge are considered in meaning determination.

To interpret compound lexical expressions with equal sensitivity to context, we perform some simulation over the Bubble Lexicon graph. The process of determining the meaning of a lexical compound such as "fast car" involves the generation of possible interpretations of how the "fast" and "car" nodes are conceptually related through dependency paths, followed by a valuation of each generated interpretation with regard to its structural plausibility and active contexts.

The organization of the rest of this paper is as follows. First, we present a more detailed overview of the Bubble Lexicon architecture and situate the knowledge representation in the literature. Second, we present mechanisms associated with this lexicon, such as context-sensitive interpretation of words and compounds. Third, we discuss an implementation of a Bubble Lexicon and present evaluation for the work through a context-sensitive linguistic interpretation task. Fourth, we briefly review related work. In our conclusion we revisit the bigger picture of the mental lexicon.

2 Bubble Lexicon Architecture

This section introduces the Bubble Lexicon Architecture (BLA) through several subsections. We begin by explaining the lexicon's knowledge representation. Next, we enumerate some tenets and assumptions of the BLA. Finally, we present an ontology of types for nodes, relations, and operators. Before we continue, we offer the following caveat to the reader: As the intention of this paper is to explore the semantics of the lexicon, our references to words are to words as concepts; as such, this work does not address words which are chiefly syntactic in nature, and we do not yet strive for a complete picture of the lexicon.

2.1 Knowledge Representation Considerations

A Bubble Lexicon is represented by a symbolic-connectionist semantic network specially purposed to serve as a computational lexicon. Nodes function as content-less *indices* for words, lexical compounds (linguistic units larger than words, such as phrases), and contexts (e.g. a discourse topic). Edges are labeled dually with a rather minimal set of conceptual dependency relations sufficient to describe the relationships between nodes, and with a numerical weight. Operators are special relations which can hold between nodes, between edges, and between operator relations themselves; they introduce boolean logic and the notion of ordering, which is necessary to represent certain types of knowledge (e.g. ordering is needed to represent verb-argument structures).

Different from traditional notions of the lexicon, the static graph is not meaningful by itself, and therefore, the semantic network's associated mechanisms are also an integral part of the lexicon. Words, as content-less nodes, have no core meaning *per se*, because a word does not encapsulate semantics. Words only have an *interpretive meaning*, arising out of some simulation of the graph. Whereas spreading activation, the act of meaning expansion via graph expansion (cf. [5]), in ordinary semantic networks is used to determine semantic proximity, our version of spreading activation outward from the word node is used to dynamically create a context bubble of interpretive meaning for the word. To model the influence of discourse contexts, which nuance meaning, our version of spreading activation appropriately accounts for all such active contexts when growing and shaping the bubble.

Some properties of the design warrant further explanation and are discussed in the following subsections.

Connectionist weights. Connectionism and lexicon design are not usually considered together. However, we believe that there are several reasons why a purely symbolic semantic network is insufficient to represent a dynamic and context-sensitive lexicon.

First, not all knowledge contributes equally to a word's meaning, so we need numerical weights on edges as an indication of semantic relevance. Some knowledge is more central to a word, but it is still of benefit to keep the peripheral knowledge about a word because it may be useful in certain cases. For example, if a context associated with some peripheral knowledge is active, the inclusion of that knowledge into the dynamic meaning bubble of the word will be more likely.

Second, connectionist weights lend the semantic network notions of memory and learning, exemplified in [16], [17], and [21]. For the purposes of creating and maintaining a computational lexicon, it may be desirable to perform supervised training on the lexicon to learn particular meaning bubbles for words under certain contexts. Learning can also be useful when importing existing lexicons into a Bubble Lexicon through an exposure process similar to semantic priming [1].

Third, connectionism gives the graph *intrinsic semantics*, meaning that even without symbolic labels on nodes and edges, the graded inter-connectedness of nodes is meaningful. This is useful in conceptual analogy over Bubble Lexicons. Goldstone and Rogosky [8] have demonstrated that it is possible to identify conceptual correspondences across two connectionist webs without symbolic identity. If we are also given symbolic labels on relations, as we are in BLA, the structure-mapping analogy making methodology described by Falkenhainer et al. [6] also becomes useful.

Finally, although not the focus of this paper, a memory-capable lexicon helps to support lexicon evolution tasks such as lexical acquisition (new word meanings), generalization (merging meanings), and individuation (cleaving meanings). A discussion of this can be found elsewhere [11].

Ontology of Conceptual Dependency Relations. In a Bubble Lexicon, edges are relations which hold between word, compound, and context nodes. In addition to having a numerical weight as discussed above, edges also have a symbolic label representing a dependency relation between the two words/concepts. The choice of the relational ontology represents an important tradeoff. Very relaxed ontologies that allow for arbitrary predicates like `bite(dog,mailman)` in Peirce's existential graphs [18] or node specific predicates as in Brachman's *description logics* system [2] are not capable of highly generalized reasoning. Efforts to engineer ontologies that enumerate *a priori* a complete set of primitive semantic relations, such as Ceccato's *correlational nets* [3], Masterman's primitive concept types [14], and Schank's Conceptual Dependency [22], show little agreement and are difficult to engineer. On the opposite end of the spectrum, overly impoverished relation types, such as the nymic subtyping relations of WordNet [7], severely curtail the expressive power of the lexicon.

Dependency relation in BLA are designed for generalized reasoning, and instead of enumerating a complete set of semantic relations, we enumerate a rather complete set of generic *structural* relationships which can hold between all concepts. For example, instead of `grow(tree,fast)`, we have `ability(tree,grow)` and `param(grow,fast)`. These relations are meant as a more fully expressive set of those found in Quillian's original Semantic Memory System. These structural relationships are useful to linguistic assembly tasks when building larger compound expressions from lexical items. They can be thought of as a sort of semantic grammar, dictating how concepts can assemble.

2.2 Tenets and Assumptions

Tenets. Graphically Figure 1 resembles an ordinary semantic network, but the differences will soon be evident. Here, we explain some principled differences.

1. **No coherent meaning without simulation.** From a static view of the Bubble Lexicon, lexical items are merely a collection of relations to a large number of other nodes, reflecting the many useful perspectives of a word, conflated onto a single network; therefore, lexical items hardly have any coherent meaning in the static view. When human minds think about what a word or phrase means, meaning is always evaluated in some context. Similarly, a word only becomes coherently meaningful in a bubble lexicon as a result of simulation (graph traversal) via spreading activation (edges are weighted, though figure 1 does not show the weights) from the origin node, toward some destination. The destination may be a context node, or if a larger lexical expression is being assembled, toward other lexical nodes. The destination node, no matter if it is a word or a context, can be seen as a *semantic primer*. It provides a context which helps us hammer down a more coherent and succinct meaning.
2. **Activated nodes in the context biases interpretation.** The meaning of a word, therefore, is the collection of nodes and relations it has traversed in moving toward its context destination. The meaning of a compound expression, such as an adjective-noun pair, is the collection of paths from the root noun to the adjec-

tival attribute. One path corresponds to one "word sense" (rather unambiguous meaning). The viability of a word sense path depends upon any context biases near the path which may boost the activation energy of that path. In this manner, semantic interpretation is very naturally influenced by context, as context prefers one interpretation by lending activation energy to its neighbors.

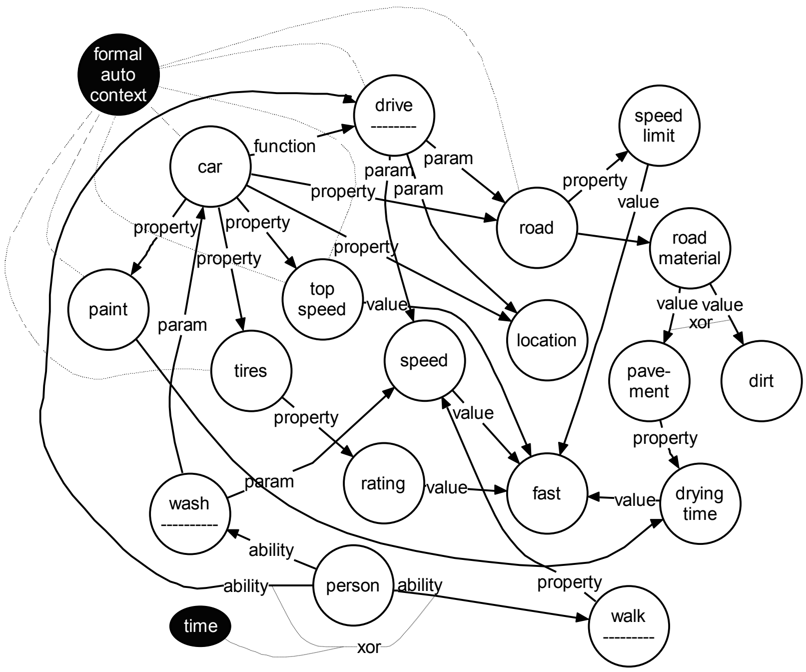

Fig. 1. A static view of a Bubble Lexicon excerpt. We selectively depict some nodes and edges relevant to the lexical items "car", "road", and "fast". Edge weights are not shown. Nodes cleaved in half by a straight line are trans-nodes, and have some notion of syntax. The black node is a context-activation node.

Assumptions. For convenience of representation, we make some assumptions.

1. Nodes are word-concepts. Although determiners, pronouns, and prepositions can be represented as nodes, we will not show these because they are connected to too many nodes.
2. Nodes may also be larger lexical expressions, such as "fast car," constructed through the process of encapsulation (explanation to follow), or they may be unnamed, intermediate concepts, though those are usually not shown.
3. In our examples, we show selected nodes and edges, although the success of such a lexicon design thrives on the network actually being very well-connected and dense.
4. Homonyms, which are non-systematic word senses (e.g. fast: not eat, vs. quick) are assumed to be represented by different nodes. Only polysemy is handled by the same node. We assume we can cleanly distinguish between these two classes of word senses. Compare to WordNet, where no effort is made to distinguish between systematic polysemy and homonymy.

5. Though not shown, relations are always numerically weighted between 0.0 and 1.0, in addition to the predicate label, and nodes also have a *stable activation energy*, to be discussed later.

2.3 Ontology of Nodes, Relations, and Operators

As shown in Figure 2, there are three types of nodes. **Normal nodes** may be word-concepts, or larger encapsulated lexical expressions. However, some kinds of meaning i.e. actions, beliefs, implications are difficult to represent. While most semantic networks have overcome this problem by introducing a causal relation [21], [17], we opted for a causal node called a **TransNode** because it offers a more precise account of causality as being *inherent* in some word-concepts, like actions. Because meaning determination is dynamic, TransNodes behave causally during simulation. TransNodes derive from Minsky's general interpretation [15] of Schankian transfer [22], and is explained more fully elsewhere [11].

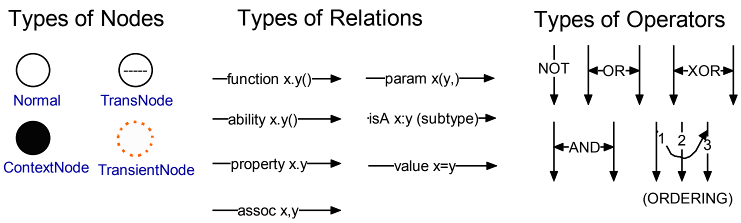

Fig. 2. Ontology of node, relation, and operator types.

ContextNodes are an interesting feature of the BLA that introduce the notion of contexts to the lexicon. Just as word nodes provide an index to an idea, ContextNodes provide an explicit index to more formal sorts of contexts (though contexts may also be informal). ContextNodes use the assoc (generic association) relation, along with operators, to cause the network to be in some state when they are activated. Meta-level ContextNodes that control a layer of ContextNodes are also possible. In Figure 1, the "formal auto context" ContextNode is meant to represent a person's stereotyped notion of what constitutes the domain of automotives. When it is activated, it loosely activates a set of other nodes. The next section will explain their activation. ContextNodes are not precisely defined and will not be identical across people; however, we posit that because knowledge of domains is a part of commonsense knowledge, these nodes will be conceptually similar across people.

Because ContextNodes help to group and organize nodes, they are useful in producing abstractions, just as a semantic frame might. Let us consider again the example of a car, as depicted in Figure 1. A car can be thought of as an assembly of its individual parts, or it can be thought of functionally as something that is a type of transportation that people use to drive from point A to point B. We need a mechanism to distinguish these two abstractions of a car. We could, of course, depend purely on lexical context from other parts of the larger lexical expression to help us pick the right abstractions; however, it is occasionally desirable to enforce more absolutely the abstraction boundaries. After all, we can view an *abstraction* as a type of packaged context. Along these lines, we can introduce two ContextNodes to define the appropriate abstractions.

So far we have only talked about nodes that are stable word-concepts and stable contexts in the lexicon. These can be thought of as being stable in memory, and very slowly changing. However, it is also desirable to represent more temporary concepts, such as those used in thought. For example, to reason about "fast cars", one might encapsulate one particular sense of fast car into a **TransientNode**. Or one can instantiate a concept and overload its meaning. TransientNodes explain how fleeting concepts in thought can be reconciled with the lexicon, which contains more *stable* elements. In the next section we illustrate the instantiation of a TransientNode.

It's clear that with the introduction of transient concepts to the graph, a Bubble Lexicon is far more than a static structure but rather, it is a platform over which linguistic assembly tasks can take place. We argue that this should not be a strange idea because in the human mental model, there is no line drawn between the lexicon and the ideas that are constructed out of them.

We present a small ontology of structural **relations** to represent fairly generic structural relations between concepts. Object-oriented programming notation is useful shorthand because the process of search in the network engages in *structural marker passing* of *relations*, where symbol binding occurs. It is also important to be reminded, at this point, that each edge carries not only a named relation, but also a numerical weight, indicating the strength of a relation. Numerical weights are critically important in all processes of Bubble Lexicons, especially spreading activation and learning.

Operators put certain conditions on relations. In Figure 1, road material may only take on the value of *pavement* or *dirt,* and not both at once. Some operators will only hold in a certain instantiation or a certain context; so operators can be conditionally activated by a context or node. For example, a person can drive and walk, but under the time context, a person can only drive XOR walk.

3 Bubble Lexicon Mechanisms

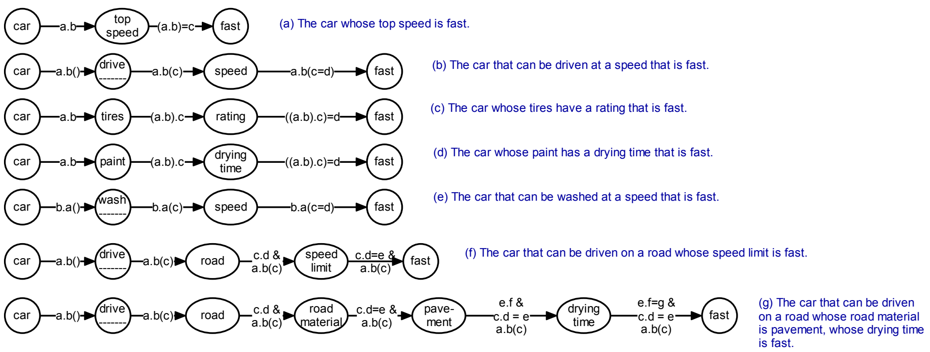

Fig. 3. Different meanings of "fast car," resulting from network traversal. Numerical weights and other context nodes are not shown. Edges are labeled with message passing, in OOP notation. The ith letter corresponds to the ith node in a traversal.

Having described the types of nodes, relations, and operators, we now explain the processes that are core themes of the Bubble Lexicon.

Meaning Determination. One of the important tenets of the lexicon's representation in Bubble Lexicons is that coherent meaning can only arise out of simulation. That is to say, out-of-context, word-concepts have so many possible meanings associated with each of them that we can only hope to make sense of a word by putting it into some context, be it a topic area (e.g. traversing from "car" toward the ContextNode of "transportation") or lexical context (e.g. traversing from "car" toward the normal node of "fast"). We motivate this meaning as simulation idea with the example of attribute attachment for "fast car", as depicted in Figure 1. Figure 3 shows some of the different interpretations of "fast car".

As illustrated in Figure 3, "fast car" produces many different interpretations given no other context. Novel to Bubble Lexicons, not only are numerical weights passed, structural messages are also passed. For example, in Figure 1, "drying time" will not always relate to "fast" in the same sense. It depends on whether or not pavement is drying or a washed car is drying. Therefore, *the history of traversal functions to nuance the meaning of the current node.* Unlike Charniak's notion of marker passing [4], whose purpose is to mark paths, structural marker passing in Bubble Lexicons is accretive, meaning that each node contributes to the message being passed.

Although graph traversal produces many meanings for "fast car," most of the senses will not be very *energetic*, that is to say, they are not very plausible in most contexts. The senses given in Figure 3 are ordered by plausibility. Plausibility is determined by the activation energy of the traversal path. Spreading activation across a traversal path is different than spreading activation in literature.

$$\mathrm{A}_{ij_x} = \sum_{n=i}^{j} \omega_{n-1,n}\alpha_n \quad \textbf{(1)} \qquad\qquad \mathrm{A}_{ij_x} = \sum_{n=i}^{j} \sum_{c}^{\substack{active\\contexts}} \omega_{n-1,n}\pi_{M_{n,n+1}}\alpha_n \mathrm{A}_{cn} \quad \textbf{(2)}$$

Equation (1) shows how a typical activation energy for the xth path between nodes i and j is calculated in classical spreading activation systems. It is the summation over all nodes in the path, of the product of the activation energy of each node n along the path, times the magnitude of the edge weight leading into node n. However, in a Bubble Lexicon, we would like to make use of extra information to arrive at a more precise evaluation of a path's activation energy, especially against all other paths between i and j. This can be thought of as meaning disambiguation, because in the end, we inhibit the incorrect paths which represent incorrect interpretations.

To perform this disambiguation, the influence of external contexts that are active (i.e. other parts of the lexical expression, relevant and active non-lexical knowledge, discourse context, and topic ContextNodes), and the plausibility of the structural message being passed are factored in.

If we are evaluating a traversal path in a larger context, such as a part of a sentence or larger discourse structure, or some topic is active, then there will likely be a set of word-concept nodes and ContextNodes which have remained active. These contexts are factored into our spreading activation evaluation function (2) as the summation over all active contexts *c* of all paths from *c* to *n*.

The plausibility of the structural message being passed $\pi_{M_{n,n+1}}$ is also important. Admittedly, for different linguistic assembly tasks, different heuristics will be needed. In attribute attachment (e.g. adj-noun compounds), the heuristic is fairly straightforward: The case in which the attribute characterizes the noun-concept directly is preferred, followed by the adjective characterizing the noun-concept's ability or use (e.g. Fig. 3(b)) or subpart (e.g. Fig. 3(a,c,d)), followed by the adjective characterizing some external manipulation of the noun-concept (e.g. Fig. 3(e)). What is not preferred is when the adjective characterizes another noun-concept that is a sister concept (e.g. Fig. 3(f,g)). Our spreading activation function (2) incorporates classic spreading activation considerations of node activation energy and edge weight, with context influence on every node in the path, and structural plausibility.

Recall that the plausibility ordering given in Figure 3 assumed no major active contexts. However, let's consider how the interpretation might change had the discourse context been a conversation at a car wash. In such a case, "car wash" might be an active ContextNode. So the meaning depicted in Fig. 3(e) would experience increased activation energy from the context term, $\mathrm{A}_{"car-wash",wash}$. This boost makes (e) a plausible, if not the preferred, interpretation.

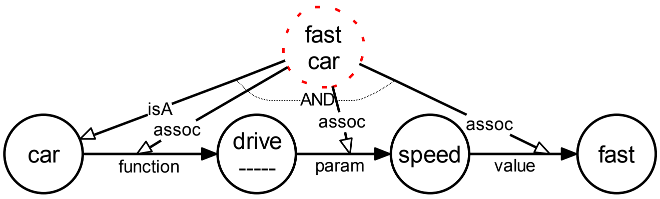

Fig. 4. Encapsulation. One meaning of "fast car" is encapsulated into a TransientNode, making it easy to reference and overload.

Encapsulation. Once a specific meaning is determined for a lexical compound, it may be desirable to refer to it, so, we can assign to it a new index. This happens by the process of encapsulation, in which a specific traversal of the network is captured into a new TransientNode. (Of course, if the node is used enough, over time, if may become a stable node, but lexical evolution is not the focus of this paper). The new node inherits just the specific relations present in the nodes along the traversal path. Figure 4 illustrates sense (b) of "fast car".

More than just lexical compounds can be encapsulated. For example, groupings of concepts (such as a group of specific cars) can be encapsulated, along with objects that share a set of properties or descriptive features (Jackendoff calls these *kinds* [9]), and even assertions and whole lines of reasoning can be encapsulated (with the help of the Boolean and ordering operators). And encapsulation is more than just a useful way of abstraction-making. Once a concept has been encapsulated, its meaning can be *overloaded*, evolving away from the original meaning. For example, we might instantiate "car" into "Mary's car," and then add a set of properties specific to Mary's car. We believe encapsulation, along with classical weight learning, supports accounts of lexical evolution, namely, it helps to explain how new concepts may be acquired, concepts may be generalized (concept intersection), or individuated (concept overloading). Lexical evolution mechanisms are discussed elsewhere [11].

Importing Existing Knowledge into the Bubble Lexicon. On a more practical note, one question which may be looming in the reader's mind is how a Bubble Lexicon might practically be constructed. A practical solution would be to bootstrap the network by learning frame knowledge from existing lexicons, such as GLT, or even Cyc [10], a database of lexical and non-lexical world knowledge. Taking the example of Cyc, we might map Cyc containers into nodes, predicates into TransNodes, and map micro-theories (Cyc's version of contexts) into ContextNodes which activate concepts within each micro-theory. Assertional knowledge can be encapsulated into new nodes. To learn the intrinsic weights on edges, supervised learning can be used to semantically prime the network to the knowledge being imported. Cyc suffers from the problem of rigidity, especially contextual rigidity, as exhibited by microtheories which predefine context boundaries. However, we believe that once frames are imported into a Bubble Lexicon, the notion of context will become much more flexible and dynamic, through the process of meaning determination. Contexts will also evolve, based on the notion of utility, not just predefinition.

4 Implementation

To test the ideas put forth in this paper, we implemented a Bubble Lexicon over a subset of the Open Mind Commonsense Semantic Network (OMCSNet) [13] based on the Open Mind Commonsense knowledge base [23], and using the adaptive weight training algorithm developed for a Commonsense Robust Inference System (CRIS) [12]. OMCSNet is a semantic network of commonsense knowledge, including both lexical and non-lexical knowledge. This goes toward our point that in designing a nuance-laden lexicon, it is desirable to include both lexical and non-lexical (world) knowledge.

Edge weights were assigned an *a priori* fixed value, based on the type of relation. The spreading activation evaluation function described in equation (2) was implemented. We planted 4 general ContextNodes through OMCSNet, based on proximity to the *hasCollocate* relation, which was translated into the *assoc* relation in the Bubble Lexicon. An experiment was run over four lexical compounds, alternatingly turning on each of the ContextNodes plus the null ContextNode. ContextNode activations were set to a very high value to elicit a context-sensitive meaning. Table 1 summarizes the results.

One difference between the attribute attachment example given earlier in the paper and the results of the evaluation is that assertional knowledge (e.g. "Gas money to work can be cheap") is an allowable part of the traversal path. Assertional knowledge is encapsulated as a node.

Table 1. Results of an experiment run to determine the effects of active context on attribute attachment in compounds.

Compound (context)	Top Interpretation (A_{ij_x} score in %)
Fast horse ( )	Horse that is fast. (30%)
Fast horse (money)	Horse that races, which wins money, is fast. (60%)
Fast horse (culture)	Horse that is fast (30%)
Fast horse (transportation)	Horse is used to ride, which can be fast. (55%)
Cheap apartment ( )	Apartment that has a cost which can be cheap. (22%)
Cheap apartment (money)	Apartment that has a cost which can be cheap. (80%)
Cheap apartment (culture)	Apartment is used for living, which is cheap in New York. (60%)
Cheap apartment (transportation)	Apartment that is near work; Gas money to work can be cheap (20%)
Love tree ( )	Tree is a part of nature, which can be loved (15%)
Love tree (money)	Buying a tree costs money; money is loved. (25%)
Love tree (culture)	People who are in love kiss under a tree. (25%)
Love tree (transportation)	Tree is a part of nature, which can be loved (20%)
Talk music ( )	Music is a language which has use to talk. (30%)
Talk music (money)	Music is used for advertisement, which is an ability of talk radio. (22%)
Talk music (culture)	Music that is classical is talked about by people. (30%)
Talk music (transportation)	Music is used in elevators where people talk. (30%)

As the results show, meaning interpretation can be very context-sensitive. With it comes several consequences. First, meaning interpretation is very sensitive to the sorts of concepts/relations/knowledge present in the lexicon. For example, in the last example in Table 1, "talk music" in the transportation context was interpreted as "music is used in elevators, where people talk." This interpretation came about, even though music is played in buses, cars, planes, and everywhere else in transportation. This has to do with the sparseness of relations in the test Bubble Lexicon. Although those other transportation concepts were present, they were not properly connected to "music". What this suggests is that meaning is not only influenced by what exists in the network, it is also heavily influenced by what is *absent* from the network, such as the absence of a relation that should exist.

Generally though, this preliminary evaluation of Bubble Lexicon for meaning determination is promising, as it shows how the meaning of lexical expressions can be made very sensitive to context. Compare this with traditional notions of polysemy as a phenomenon which is best handled through a fixed and predefined set of *senses*. But *senses* do not produce the more systematically graded generative behavior of the human mental lexicon, where meaning is naturally and subtly nuanced by context.

4 Related Work

Ross Quillian's Semantic Memory System [20] was the initial inspiration for this work. The main idea about meaning being distributed over a graph came from Quillian. In the semantic memory system, Quillian sought to demonstrate some basic semantic capabilities over a network of word-concepts, namely, comparing and contrasting words. The relations initially proposed represented minimal structural dependencies, only later to be augmented with some other relations including proximity, consequence, precedence, and similarity. The type of knowledge represented in the network was denotative and dictionary-like. The system did not attempt to explain the influence of context on meaning, or the impact of non-lexical knowledge on meaning. Linguistic assembly and sense interpretation tasks were not attempted. With the Bubble Lexicon research, we attempt to build on Quillian's work. We explain how such a semantic memory might be used to circumvent the limitations of traditional lexicons. We take a context-centered approach in modeling how meaning gets shaped, and how context impacts linguistic assembly and sense interpretation. We also introduce the connectionist semantics to the network as a vehicle for conceptual analogy and learning; the notion that non-lexical knowledge participates in shaping meaning; and the notion of structural marker passing and structural plausibility, as a method for valuating word and phrase senses.

5 Conclusion

In this research, we sought to address the inability of traditional computational lexicons to be pliable to context and to generate nuance-laden meaning. To overcome the limitations imposed by an artificial word boundary and the artificial exclusion of non-lexical knowledge from the lexicon, we explore a context-aware symbolic-connectionist semantic network called a Bubble Lexicon. Rather than representing words as static encapsulations of meaning, the Bubble Lexicon dynamically generates context bubbles of meaning which vary based on the active contexts. Explicit context nodes, the inclusion of non-lexical knowledge, and intrinsic weights on relations all serve to add nuance to meaning determination. Bubble Lexicon is more than a static structure; it is a potential platform for performing nuanceful linguistic assembly tasks such as context-sensitive attribute attachment (e.g. "fast car").

A brief implementation and evaluation of the Bubble Lexicon over a commonsense repository called OMCSNet yielded some interesting preliminary findings. The semantic interpretation of lexical compounds successfully demonstrated context-sensitivity. However, these preliminary findings also tell a cautionary tale. The accuracy of a semantic interpretation is heavily reliant on the concepts in the network being well-connected and densely packed. This makes the task of importing traditional lexicons into bubble networks all the more challenging, because traditional lexicons are typically sparse and not conceptually well-connected. However, we are optimistic that the large repositories of world knowledge being gathered recently will serve as a well-populated foundation for such a lexicon. The research described in this paper explores lexicons that approach the generative power of the human language faculty. We cannot help but note that as such a lexicon design grows toward its goal, it also approaches a comprehensive model of thought and semantic memory.

6 Acknowledgements

We are grateful to the following individuals for their helpful ideas in the course of this research: Push Singh, Andrea Lockerd, Deb Roy, Marvin Minsky, Henry Lieberman.

7 References

1. Becker, S., Moscovitch, M., Behrmann, M., & Joordens, S. (1997). Long-term semantic priming: A computational account and empirical evidence. Journal of Experimental Psychology: Learning, Memory, & Cognition, 23(5), 1059-1082.
2. Brachman, R. (1979) "On the epistemological status of semantic networks," in Findler 3-50.
3. Ceccato, Silvio (1961) Linguistic Analysis and Programming for Mechanical Translation, Gordon and Breach, New York.
4. Charniak, E. (1986) A Neat Theory of Marker Passing. AAAI 1986, 584-588
5. Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. Psychological Review, 82, 407--428.
6. Falkenhainer, B., Forbus, K.D. and Gentner, D. (1990). The structure-mapping engine: algorithm and examples. Artificial Intelligence, 41:1-63.
7. Fellbaum, C. (Ed.). (1998). WordNet: An electronic lexical database. Cambridge, MA: MIT Press.
8. Goldstone, R. L., Rogosky, B. J. (2002). Using relations within conceptual systems to translate across conceptual systems. Cognition 84, 295-320
9. Jackendoff, R. (2002). Reference and Truth. In, Jackendoff, R., Foundations of Language. Oxford University Press, 2002.
10. Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11).
11. Liu, H. (2002). Bubble Networks: A Context-Sensitive Lexicon Representation. MIT Media Lab Technical Report SA02-02. Available at: web.media.mit.edu/~hugo/publications.
12. Liu, H. and Lieberman, H. (2002). Robust photo retrieval using world semantics. Proceedings of LREC2002 Workshop: Creating and Using Semantics for IR, pp. 15-20, Las Palmas, Canary Islands.
13. Liu, H. and Singh, P. (2003). OMCSNet: A commonsense inference toolkit. Submitted to IJCAI 2003. Available at: web.media.mit.edu/~hugo/publications
14. Masterman, Margaret (1961) "Semantic message detection for machine translation, using an interlingua," in NPL (1961) pp. 438-475.
15. Minsky, M. (1986). Society of Mind. Simon and Schuster, New York.
16. Minsky, M. and Papert, S. Perceptrons: An Introduction to Computational Geometry. The MIT Press, 1969.
17. Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA.
18. Peirce, Charles Sanders (1885) "On the algebra of logic," American Journal of Mathematics 7, 180-202.
19. Pustejovsky, J. (1991) `The generative lexicon', Computational Linguistics, 17(4), 409--441.
20. Quillian, M. (1968) Semantic Memory. In M. Minsky, ed, Semantic Information Processing, 216-270. MIT Press, Cambridge, MA
21. Rieger, Chuck (1976) "An organization of knowledge for problem solving and language comprehension," Artificial Intelligence 7:2, 89-127.
22. Schank, R.C., Tesler, L.G. (1969) A Conceptual Parser for Natural Language. IJCAI 1969: 569-578
23. Singh, P. et al. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. In Proceedings of ODBASE'02. LNCS. Heidelberg: Springer-Verlag.
24. Various Authors. (2002) Cheap apartments are rare. Available at: http://agentsbbs.media.mit.edu:8080/carr/