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 knowledge representation tradition in computational lexicon design represents words as static encapsulations of purely lexical knowledge. We suggest that this view poses certain 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. Hoping to address these problematics, we explore a context-centered approach to lexicon design called a Bubble Lexicon. Inspired by Ross Quillian's Semantic Memory System, we represent word-concepts as nodes on a symbolic-connectionist network. In a Bubble Lexicon, a word's meaning is defined by a dynamically grown context-sensitive bubble; thus giving a more natural account of systematic polysemy. Linguistic assembly tasks such as attribute attachment are made context-sensitive, and the incorporation of general world knowledge improves generative capability. Indicative trials over an implementation of the Bubble Lexicon lends support to our hypothesis that unpacking meaning from predefined word structures is a step toward a more natural handling of context in language.

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, certain problematics of this paradigm are beginning to reveal themselves. 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 able to produce so effortlessly. 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

P. Blackburn et al. (Eds.): CONTEXT 2003, LNAI 2680, pp. 218–232, 2003.

their classification as separate senses indistinguishable from homonyms give the false impression that there is a no-man's land of meaning in between each predefined word sense.

Hoping to address the inflexibility of lexicons like WordNet, Pustejovsky's Generative Lexicon Theory (GLT) [19] packs 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 "to climb on" is some notion of "goodness." GLT improves upon the sophistication of previous models; however, as with previous models, GLT represents words as discrete and pre-defined packages of meaning. We argue that this underlying word-as-prepackaged-meaning paradigm poses certain limitations on the generative power of the lexicon. We describe two problematics below:

1) **Artificial word boundary.** By representing words as discrete objects with predefined meaning boundaries, lexicon designers must make *a priori* and sometimes arbitrary decisions about how to partition word senses, what knowledge to encode into a word, and what to leave out. This is problematic because it would not be feasible (or efficient) to pack into a word all the knowledge that would be needed to anticipate all possible intended meanings of that word.
2) **Exclusion of non-lexical knowledge.** When representing a word as a predetermined, static encapsulation of meaning, it is common practice to encode only knowledge that formally characterizes the word, namely, *lexical knowledge* (e.g. the qualia structure of GLT). We suggest that non-lexical knowledge such as *general world knowledge* also shapes the generative power and meaning of words. General world knowledge differs from lexical knowledge in at least two ways:
   a) First, general world knowledge is largely concerned with **defeasible** knowledge, describing relationships between concepts that *can* hold true or *often* holds true (connotative). By comparison, lexical knowledge is usually a more formal characterization of a word and therefore describes relationships between concepts that *usually* holds true (denotative). But the generative power of words and richness of natural language may lie in defeasible knowledge. For example, in interpreting the phrase *"funny punch,"* it is helpful to know that "*fruit punch can sometimes be spiked with alcohol."* Defeasible knowledge is largely missing from WordNet, which knows that a *"cat"* is a *"feline"*, *"carnivore"*, and *"mammal"*, but does not know that *"a cat is often a pet."* While some defeasible knowledge has crept into the qualia structures of GLT (e.g. *"a rock is often used to climb on*"), most defeasible knowledge does not naturally fit into any of GLT's lexically oriented qualia roles.
   b) Second, lexical knowledge by its nature characterizes only word-level concepts (e.g. *"kick"*), whereas general world knowledge characterizes both word-level and **higher-order concepts** (e.g. *"kick someone"*). Higher-order concepts can also add meaning to the word-level concepts. For example, knowing

that *"kicking someone may cause them to feel pain"* lends a particular interpretation to the phrase *"an evil kick."* WordNet and GLT do not address general world knowledge of higher-order concepts in the lexicon.

It is useful to think of the aforementioned problematics as issues of context. Word boundaries seem artificial because meaning lies either wholly inside the context of a word, or wholly outside. Non-lexical knowledge, defeasible and sometimes characterizing higher-order concepts, represents a context of connotation about a word, which serves to nuance the interpretation of words and lexical compounds. Considering these factors together, we suggest that a major weakness of the word-as-prepackaged-meaning paradigm lies in its inability to handle context gracefully.

Having posed the problematics of the word-as-prepackaged-meaning paradigm as an issue of context, we wonder how we might model the computational lexicon so that meaning contexts are more seamless and non-lexical knowledge participates in the meaning of words. We recognize that this is a difficult proposition with a scope extending beyond just lexicon design. The principle of modularity in computational structures has been so successful because encapsulations like frames and objects help researchers manage complexity when modeling problems. Removing word boundaries from the lexicon necessarily increases the complexity of the system. This notwithstanding, we adopt an experimental spirit and press on.

In this paper, we propose a context-centered model of the computational lexicon inspired by Ross Quillian's work on semantic memory [21], which we dub as a *Bubble Lexicon*. The Bubble Lexicon Architecture (BLA) is a symbolic connectionist network whose representation of meaning is distributed over nodes and edges. Nodes are labeled with a word-concept (our scheme does not consider certain classes of words such as, *inter alia*, determiners, prepositions and pronouns). Edges specify both the symbolic relation and connectionist strength of relation between nodes. A word-concept node has no internal meaning, and is simply meant as a reference point, or, *indexical feature*, (as Jackendoff would call it [9]) to which meaning is attached. 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. Defeasible knowledge can be represented in the graph with the help of the connectionist properties of the network. Non-lexical knowledge involving higher-order concepts (more than one word) are represented in the graph through special nodes called *encapsulations*, so that they may play a role in biasing meaning determination.

The nuanceful generative capability of the BLA is demonstrated through the linguistic assembly task of attribute attachment, which engages some simulation over the network. For example, 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 contextual plausibility. The proposed Bubble Lexicon is not being presented here as a perfect or complete solution to computational lexicon design, but rather, as the implementation and indicative trials illustrate, we hope Bubble Lexicon is a step toward a more elegant solution to the problem of context in language.

The organization of the rest of this paper is as follows. First, we present a more detailed overview of the Bubble Lexicon Architecture, situating the 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 Bubble Lexicon and present some evaluation for the work through some indicative trials. Fourth, we briefly review related work. In our conclusion we return to revisit the bigger picture of the mental lexicon.

2 Bubble Lexicon Architecture

This section introduces the proposed Bubble Lexicon Architecture (BLA) (Fig. 1) through several subsections. We begin by situating the lexicon's knowledge representation in the literature of symbolic connectionist networks. Next, we enumerate some tenets and assumptions of the proposed architecture. Finally, we discuss the ontology of types for nodes, relations, and operators.

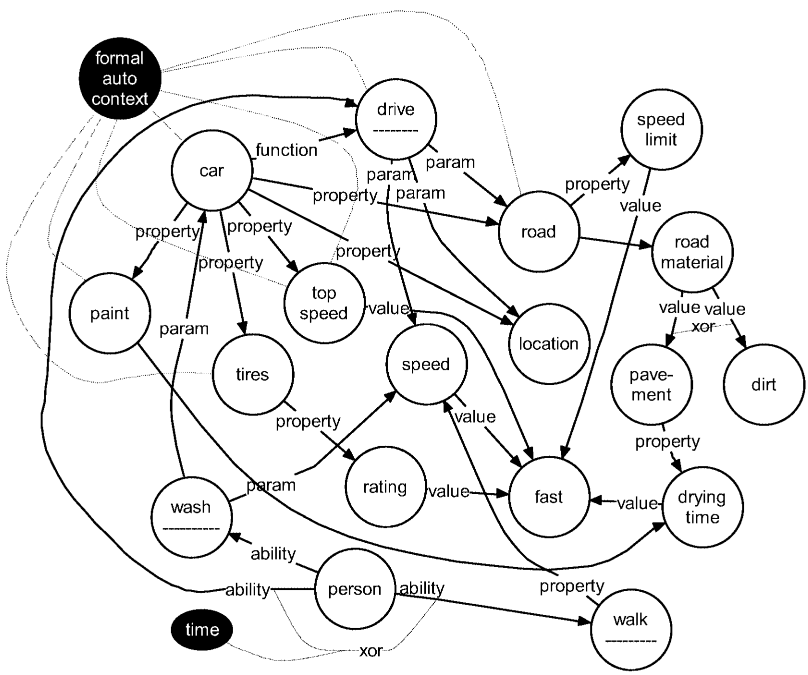

Fig. 1. A static snapshot of a Bubble Lexicon. We selectively depict some nodes and edges relevant to the lexical items "car", "road", and "fast". Edge weights are not shown. Nodes cleaved in half are causal trans-nodes. The black nodes are context-activation nodes.

2.1 Knowledge Representation Considerations

A Bubble Lexicon is represented by a symbolic-connectionist network specially purposed to serve as a computational lexicon. Nodes function as *indices* for words, lexical compounds (linguistic units larger than words, such as phrases), and formal contexts (e.g. a discourse topic). Edges are labeled dually with a minimal set of structural structural dependency relations 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 a sequence of events).

Because the meaning representation is distributed over the nodes and edges, words only have an *interpretive meaning*, arising out of some simulation of the graph. Spreading activation (cf. [5]) is ordinarily used in semantic networks to determine semantic proximity. We employ a version of spreading activation to dynamically create a context bubble of interpretive meaning for a word or lexical compound. In growing and shaping the bubble, our spreading activation algorithm tries to model the influence of active contexts (such as discourse topic), and of relevant non-lexical knowledge, both of which contribute to meaning.

Some properties of the representation are further discussed below.

Connectionist weights. Connectionism and lexicon design are not usually considered together because weights tend to introduce significant complexity to the lexicon. However, there are several reasons why connectionism is necessary to gracefully model the context problem in the 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, and to distinguish between certain from defeasible knowledge. Defeasible knowledge may in most cases be less central to a word's meaning, but in certain contexts, their influence is felt.

Second, connectionist weights lend the semantic network notions of memory and learning, exemplified in [16], [17], and [22]. For the purposes of growing 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] becomes possible.

Finally, although not the focus of this paper, a self-organizing connectionist lexicon would help to support lexicon evolution tasks such as lexical acquisition (new word meanings), generalization (merging meanings), and individuation (cleaving meanings). A discussion of this appears 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 suitable for 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 [23], show little agreement and are difficult to engineer. A small but insufficiently generic set of relations such as WordNet's nyms [7] could also severely curtail the expressive power of the lexicon.

Because lexicons emphasize *words,* we want to focus meaning around the word-concept nodes rather than on the edges. Thus we propose a small ontology of generic *structural* relations for the BLA. For example, instead of `grow(tree,fast)`, we have `ability(tree,grow)` and `parameter(grow,fast)`. These relations are meant as a more expressive set of those found in Quillian's original Semantic Memory System. These structural relations become 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. While the static graph of the BLA (Fig. 1) depicts the meaning representation, it is equally important to talk about the simulations over the graph, which are responsible for *meaning determination.* We give two tenets below:

1) **No coherent meaning without simulation.** In the Bubble Lexicon graph, different and possibly conflicting meanings can attach to each word-concept node; therefore, words hardly have any coherent meaning in the static view. We suggest that 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 Fig. 1 does not show the weights) from the origin node, toward some destination. This helps to exclude meaning attachments which are irrelevant in the current context, to hammer down a more coherent meaning.

2) **Activated nodes in the context biases interpretation.** The meaning of a word or phrase is the collection of nodes and relations it has "harvested" along the path toward its destination. However, there may be multiple paths representing different interpretations, perhaps each representing one "word sense". In BLA, the relevance of each word sense path depends upon context biases near the path which may boost the acti-

vation energy of that path. Thus meaning is naturally influenced by context, as context nodes prefers certain interpretations by activating certain paths.

Assumptions. We have made the following assumptions about our representation:
1) Nodes in BLA are word-concepts. We do not give any account of words like determiners, pronouns, and prepositions.
2) Nodes may also be higher-order concepts like "fast car," constructed through encapsulation. In lexical evolution, intermediate transient nodes also exist.
3) In our examples, we show selected nodes and edges, although the success of such a lexicon design thrives on the network being sufficiently well-connected and dense.
4) Homonyms, which are non-systematic word senses (e.g. fast: not eat, vs. quick) are represented by different nodes. Only systematic polysemy shares the same node. We assume we can cleanly distinguish between these two classes of word senses.
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*, which is a function of how often active a node is within the current discourse.

2.3 Ontology of Nodes, Relations, and Operators

We propose three types of nodes (Fig. 2). **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 because they have some notion of syntax. Some semantic networks have overcome this problem by introducing a causal relation [22], [17]. We opted for a causal node called a **TransNode** because we feel that it offers a more precise account of causality as being *inherent* in some word-concepts, like actions. This also allows us to maintain a generic structural relation ontology. Because meaning determination is dynamic, TransNodes behave causally during simulation. TransNodes derive from Minsky's general interpretation [15] of Schankian transfer [23], and is explained more fully elsewhere [11].

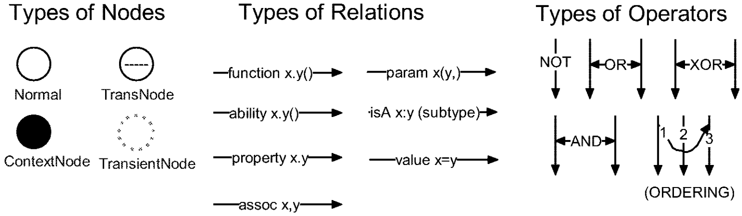

Fig. 2. Ontology of node, relation, and operator types.

While normal nodes can act as contexts when they are activated in the BLA, there is no formal definition to those groupings. We suggest that sometimes, human minds may employ more formal and explicit notions of context which define a topic or domain of discourse (e.g.: "automotives," "finance"). For example, the meaning of the formal context "finance" is somewhat different than the meaning that is attached to that word-concept node. For one, the formal context "finance" may be a well-defined term in the financial community. The external definition of certain concepts like formal contexts is supported by Putnam's idea of semantic externalism [20]. We introduce **ContextNodes** as an explicit representation of externally-defined formal contexts. ContextNodes use the assoc (generic association) relation, along with operators, to cause the network to be in some state when they are activated. They can be thought of as *externally grounded contexts.* Meta-level ContextNodes that control a layer of ContextNodes are also possible. In Figure 1, the "formal auto context" ContextNode is meant to represent formally the domain of automotives, to the best of a person's understanding of the community definition of that context.

Because ContextNodes help to group and organize nodes, they are also useful in representing perspectives, 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 can use ContextNodes to separate these two perspectives of a car. After all, we can view a perspective as a type of packaged context.

So far we have only talked about nodes which are stable word-concepts and stable contexts in the lexicon. These can be thought of as being stable in memory, and changing slowly. 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 path 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. The interaction of concepts and ideas constructed out of them should not be a strange idea because in the human mental model, there is no line drawn between them. In the next section we illustrate the instantiation of a TransientNode.

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 meaning determination in the network engages in *structural marker passing* of *relations*, where symbol binding occurs. It is also important to remember, that each edge carries not only a named relation, but also a numerical weight. Connectionist 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

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 formal 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 generated for "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 earlier notion of marker passing [4] used to mark paths, structural marker passing in Bubble Lexicons is accretive, meaning that each node contributes to the message being passed.

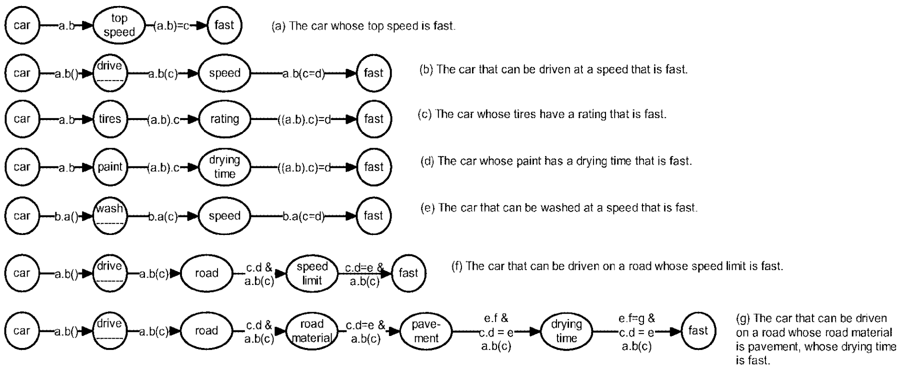

Fig. 3. Different meanings of "fast car," resulting from network traversal. Numerical weights and other context nodes are not shown. Edges are labeled with message passing, in OOP notation. The i^th letter corresponds to the i^th node in a traversal.

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 classical spreading activation from the 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 *x*th 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 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 valuation function (2) as the sum 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.

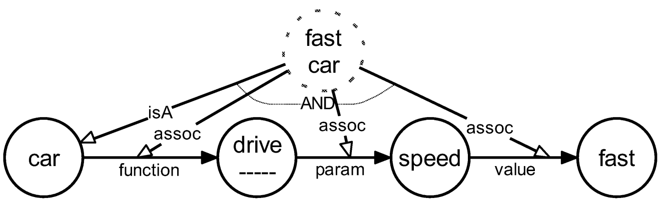

Fig. 4. Encapsulation. One meaning of "fast car" is encapsulated into a TransientNode, making it easy to reference and overload.

Encapsulation. Once a specific meaning is determined for a lexical compound, it may be desirable to refer to it, so, we assign to it a new index. This happens through a process called 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, it may become a stable node). 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. One question which may be looming in the reader's mind is how a Bubble Lexicon might be practically constructed. One practical solution is 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 pre-fix context boundaries. However, once imported into a Bubble Lexicon, meaning determination may become more dynamic and context-sensitive. Contexts will evolve, based on the notion of lexical utility, not just on predefinition.

4 Implementation

To test the ideas put forth in this paper, we implemented a Bubble Lexicon over a adapted subset of the Open Mind Commonsense Semantic Network (OMCSNet) [13] based on the Open Mind Commonsense knowledge base [24]. We use the adaptive weight training algorithm developed for a Commonsense Robust Inference System (CRIS) [12]. OMCSNet is a large-scale semantic network of 140,000 items of general world knowledge including lexical and non-lexical, certain and defeasible. Its scale provides BLA with a rich basis from which meaning can be drawn.

With the goal of running trials, 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 also labeled three existing nodes in OMCSNet as ContextNodes and translated the nodes' *hasCollocate* relations, into the *assoc* relation in the Bubble Lexicon. Predefining nodes, while not generally necessary, was done in this case to make it easier to observe the effects of context bias in indicative trials. Trials were run over four lexical compounds, alternatingly activating each of these ContextNodes plus the null ContextNode. Context activations were set to a very high value to exaggerate, for illustrative purposes, the effect of context on meaning determination. Table 1 summarizes the results.

One discrepancy between the proposed and implemented systems is that assertional knowledge (e.g. *"Gas money to work can be cheap")* in the implementation is allowed to be in the traversal path. Assertional knowledge is encapsulated as nodes.

The creative and nuanceful interpretations produced by the BLA demonstrate clearly the effects of active context on meaning determination. The incorporation of non-lexical knowledge into the phrasal meaning is visible (e.g. *"Horse that races, which wins money, is fast").* By comparison, WordNet and GLT would not have produced the varied and informal interpretations produced by BLA.

Table 1. Results of trials illustrate effects of active context on attribute attachment.

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%)

However, the implementation also reveals some difficulties associated with the BLA. Meaning interpretation is very sensitive to the quality and signal-to-noise ratio of concepts/relations/knowledge present in the lexicon, which in our case, amounts to knowledge present in OMCSNet. 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 singled out elevators, even though music is played in buses, cars, planes, and elsewhere in transportation. This has to do with the sparseness of relations in OMCSNet. Although those other transportation concepts existed, 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*, such as the absence of a relation that should exist.

Also, judging the relevance of meaning relies largely on the evolution of good numerical weights on edges; but admittedly, learning the proper weights is a difficult proposition: Though we point out that even a rough estimate of weights (for example, separating lexical and non-lexical knowledge by 0.5), as was employed in our implementation, vastly improved the performance of meaning determination.

Though the complexity and knowledge requirements remain lingering challenges for the BLA, the implementation and indicative trials do seem to support our hypothesis that unpacking meaning from predefined word structures is a step toward a more natural handling of nuance and context in language.

5 Related Work

Ross Quillian's Semantic Memory System [21] was the initial inspiration for this work, as it was one of the first to explore meaning being distributed over a graph. 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. With the Bubble Lexicon, 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 populate the network with lexical and non-lexical knowledge, and demonstrate their influences on meaning. We give an account of context-sensitive meaning determination by modifying spreading activation to account for contextual and structural plausibility; and introduce connectionism as a vehicle for conceptual analogy and learning.

6 Conclusion

In this paper, we examined certain limitations that the word-as-prepackaged-meaning paradigm imposes on the ability of the lexicon to generated highly nuanced interpretations. We formulated these problematics as issues of context, and hypothesized that a context-centered design of the computational lexicon would lend itself more to nuanced generation. We proposed a context-sensitive symbolic-connectionist 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 active contexts. The inclusion of non-lexical knowledge such as defeasible and higher-order conceptual knowledge, along with intrinsic weights on relations, all serve to nuance to meaning determination. More than a static structure, the Bubble Lexicon is a platform for performing nuanceful linguistic assembly tasks such as context-sensitive attribute attachment (e.g. "fast car").

An implementation of the Bubble Lexicon over a large repository of commonsense called OMCSNet yielded some promising findings. In indicative trials, context had a very clear effect in nuancing the interpretation of phrases, lending support to our hypothesis. However, these 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, and the numerical weights being properly learned. The task of building lexicons over symbolic connectionist networks will necessarily have to meet these needs and manage a great deal of complexity. 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 theory grows toward its goal, it also approaches a comprehensive model of thought and semantic memory.

Acknowledgements. We are indebited to the following individuals for their helpful ideas and comments in the course of this work: Deb Roy, Push Singh, Andrea Lockerd, and Marvin Minsky.

References

1. Becker, S. et al.: Long-term semantic priming: A computational account and empirical evidence. J. of Exp. Psych: Learning, Memory, & Cognition, 23(5), (1997) 1059–1082
2. Brachman, R. J.: On the epistemological status of semantic networks. In: Findler, N. V., (ed.), Associative Networks, pages 3–50. Academic Press (1979)
3. Ceccato, S.: Linguistic Analysis and Programming for Mechanical Translation, Gordon and Breach, New York (1961)
4. Charniak, E.: A Neat Theory of Marker Passing. Proceedings of AAAI, (1986) 584–588
5. Collins, A. M., and Loftus, E. F.: A spreading-activation theory of semantic processing. Psychological Review, 82, (1975) 407–428
6. Falkenhainer, B., Forbus, K. D. and Gentner, D.: The structure-mapping engine: algorithm and examples. Artificial Intelligence, 41:1–63. (1990).
7. Fellbaum, C. (Ed.).: WordNet: An electronic lexical database. MIT Press (1998).
8. Goldstone, R. L., Rogosky, B. J.: Using relations within conceptual systems to translate across conceptual systems. Cognition 84, (2002) 295–320
9. Jackendoff, R.: Chapter 10, Foundations of Language. Oxford University Press (2002)
10. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11) (1995).
11. Liu, H.: Bubble Networks: A Context-Sensitive Lexicon Representation. MIT Media Lab Technical Report SA02-02. (2002). Available at: web.media.mit.edu/~hugo/publications.
12. Liu, H. and Lieberman, H.: Robust photo retrieval using world semantics. Proceedings of LREC2002 Workshop: Using Semantics for IR, Canary Islands, (2002) 15–20
13. Liu, H. and Singh, P.: OMCSNet: A commonsense inference toolkit. *In Submission.* (2003) Available at: web.media.mit.edu/~hugo/publications
14. Masterman, M.: Semantic message detection for machine translation, using an interlingua. In: NPL (1961) 438–475
15. Minsky, M.: Society of Mind. Simon and Schuster, New York. (1986)
16. Minsky, M. and Papert, S.: Perceptrons. MIT Press (1969)
17. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA. (1988)

18. Peirce, C.S.: On the algebra of logic. American Journal of Math 7, (1885) 180–202
19. Pustejovsky, J.: The generative lexicon, Computational Linguistics, 17(4), (1991) 409–441
20. Putnam, H.: Representation and Reality, MIT Press (1988) 19–41
21. Quillian, M.: Semantic Memory. In: M. Minsky (Ed.): Semantic Information Processing, 216–270. MIT Press, Cambridge, MA (1968)
22. Rieger, C.: An organization of knowledge for problem solving and language comprehension. Artificial Intelligence 7:2, (1976) 89–127.
23. Schank, R.C., Tesler, L.G.: A Conceptual Parser for Natural Language. Proceedings of IJCAI, (1969) 569–578
24. Singh, P. et al.: Open Mind Common Sense: Knowledge acquisition from the general public. In Proceedings of ODBASE'02. LNCS. Heidelberg: Springer-Verlag (2002)