Knowledge Representation and Bayesian Inference for Response to Situations

Rakesh Gupta and **Vasco Calais Pedro***
Honda Research Institute USA, Inc.
800 California Street, Suite 300
Mountain View, CA 94041
rgupta@hra.com, vasco@cs.cmu.edu

Abstract

Commonsense reasoning is crucial in making humanoid robots capable of responding to situations in a human-like fashion. To address this challenge, we have used a Bayesian Network to compare different responses to find a likely response. This Bayesian Network is populated for the situation under consideration from a multidimensional semantic net, called the PraxiNet. PraxiNet is used to graphically represent all possible situations and responses. Instead of manually engineering the knowledge base for PraxiNet, we have used distributed knowledge capture techniques as the knowledge source for PraxiNet. We collect knowledge from volunteers over the web about causality and responses to situations. This knowledge is very noisy and is processed using Natural Language Processing (NLP) techniques including spell checking, heuristic-based pattern removal and chunking to improve the quality of knowledge. PraxiNet is expanded using WordNet and a thesaurus, and subsequently condensed by lemmatization, synonym and hypernym merging to increase the overlap of knowledge and the density of the network. Given a situation (or multiple situations) we extract the relevant part of PraxiNet into the Bayesian Network for computation of suitable responses. This approach is scalable and can handle millions of pieces of knowledge to find the common sense responses for a given situation.

Introduction

Over the past decades humanoid robots have evolved rapidly but most of the advances have been of a mechanical nature. As humanoid robots are built with an increasing range of physical abilities, the lack of world knowledge and sophisticated human-like reasoning algorithms become a major bottleneck. Mobile robots in homes and offices will be expected to respond to situations within their environment to satisfy the requested desires of their users. A key factor in meeting these expectations and smooth interaction between humans and robots is the robot's ability to behave correctly when faced with daily situations.

Given that humans already possess the necessary knowledge to deal with most situations, one promising way of dealing with this problem is to collect the relevant options for responding to the situation and to compare their probabilities. Bayesian Networks provide a framework to graphically represent and compute such response probabilities to find the most suitable response.

Inspired by work in semantic nets (Quillian 1967), the WordNet framework (Fellbaum 1998) and the OpenMind-based ConceptNet Initiative (Liu & Singh 2004) we have created PraxiNet to graphically represent and process knowledge about all possible situations and responses. Based on the concept of Praxis (*action* in Greek), PraxiNet is a multi-dimensional network where each dimension represents a type of semantic net. Unlike other conceptual graph systems that are based on objects and their properties, the distinguishing characteristic of PraxiNet is its knowledge representation which is based on situations and responses.

PraxiNet graph needs knowledge of situations and responses. In the past, knowledge has been manually engineered for the task or collected from volunteers over the web. Prior efforts to create large knowledge bases manually include ThoughtTreasure (Mueller 1998) and Cyc (Lenat & Guha 1990) using expert knowledge to hand-craft rules and knowledge. Hand-crafted knowledge and rules have several disadvantages. First, these rules require manual effort by specialists in the domain and in the rule representation language. Second, maintaining the consistency of the rule set becomes increasingly difficult as the number of rules grows. Finally, when retrieving the knowledge encoded in the knowledge base, the reasoning process is limited to matching preconditions of rules, which hinders its flexibility. Knowledge can also be gathered from non-specialist web users in the same fashion as the projects associated with the *OpenMind Initiative* pioneered by David Stork (Stork 1999). Efforts under this umbrella like MIT Media Lab OpenMind Common Sense (Liu & Singh 2004) contain broad and sparse common sense knowledge. Such sparse knowledge is not suitable for probabilistic reasoning.

We have generated our knowledge base using distributed knowledge capture. The OpenMind Indoor Common Sense (OMICS) database (Gupta & Kochenderfer 2004) contains a large set of common sense knowledge in the form of relational tables applicable to indoor home and office environments. Restriction on the domain gives dense knowledge suitable for probabilistic reasoning. However, collected knowledge is noisy and inconsistent, and converting

*Currently at the Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania,USA it into a usable form is a challenge. We have developed automated techniques to pre-process the OMICS data using linguistic techniques and tools like WordNet, so that all the knowledge can be efficiently represented in the PraxiNet. Linguistic Natural Language Processing (NLP) techniques include spell checking, heuristic-based pattern removal and chunking to improve the quality of knowledge. PraxiNet is expanded using WordNet and a thesaurus, and subsequently condensed by lemmatization, synonym and hypernym merging to increase the overlap of knowledge and the density of the network. Given a situation (or multiple situations) we extract the relevant part of PraxiNet into the Bayesian Network for computation of suitable response.

The remainder of this paper describes our approach in a bottom-up manner. In the next section we discuss OpenMind Indoor Common Sense data characteristics and explain the situations and response data. This is followed by preprocessing of data using linguistic techniques, followed by PraxiNet creation and techniques to increase its density. We then describe the creation of the Bayesian network and the discussion of the results, statistics and evaluation of PraxiNet and the reasoning approach, followed by Conclusions and Future Work.

Distributed Knowledge capture

The OpenMind Indoor Common Sense project (Gupta & Kochenderfer 2004) has successfully captured thousands of pieces of common sense knowledge about home and office environments in a relational database. Users enter words or phrases to complete natural language sentences. The semantic information comes directly from the templates used in data capture. The framework of the OMICS project is *object-centric* where actions taken by the robot are assumed to be grounded in the properties of objects in the world. Each piece of knowledge in OMICS is reviewed and only good knowledge is accepted in the database.

Situation and Response as Compound Objects

One of the key concepts in PraxiNet is the existence of compound objects representing situations and responses. For PraxiNet, a **situation** is the conjunction of an *object* and *property*, e.g. floor dirty, window broken, coffee cup empty. An **response** is the conjunction of an *action* and an *object*, e.g. clean floor, repair window, refill coffee cup. These situations and responses are schematically shown in Figure 1. We can create associations at the compound level, and specify causality and response relations to particular situations. As a result, we can reason at situations and responses level, rather than over objects and their properties.

Our work is different compared to other approaches such as ConceptNet and LifeNet (Singh & Williams 2003) in an important way. Since situations are specified with their components, PraxiNet has an intrinsic understanding of components in the compound node, which doesn't happen in conceptnet. This enables similarity determination between compound nodes based on the similarity of the component parts. For example, we are able to automatically determine that saying *open the map* is another way of saying *unfold the map* because open and unfold are synonyms. LifeNet and ConceptNet do not have the concept of compound nodes, and therefore cannot work with knowledge at a higher composite level.

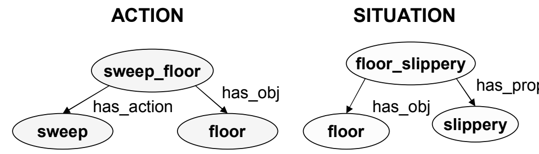

Figure 1: Kinds of OpenMind data used in PraxiNet

In this work we have used **Response** and **Causes** entries from the OMICS database. In Responses, we had entries of the kind *In situation A, one should do response B*. An example of Response entry is *If phone is ringing, one should answer the phone*. In Causes, we had entries of the kind *Situation C causes situation D*. An example of Causes entry is *Fan on causes room to be cool*.

Linguistic Preprocessing

The knowledge is extracted from the OMICS database, through a series of database queries that select the appropriate data. The goal in Linguistic Preprocessing is to reduce the noise and increase the usage and density of knowledge. To prepare the knowledge we first run a spell-check and correct the spellings. We evaluated three different spell checkers with Java API and chose WinterTree software. Overall 370 Words spellings are corrected automatically using the WinterTree software. Of these, 81% are manually checked to be correct replacements. Unchecked, these spelling errors create non existent concepts and increase the sparseness of the data by reducing the amount of valid data.

The next step is to strip down the non-essential text in the knowledge base using heuristic based pattern removal. Heuristic-based pattern removal is applied to object property and action fields to reduce the noise in the knowledge by removing non-essential words and patterns. We use JMontyLingua as a linguistic tool to tag and lemmatize. MontyLingua was developed by Hugo Liu at MIT and is based on the Brill Tagger enhanced by OpenMind commonsense (Liu 2004). In the objects, we typically want the noun reference rather than the descriptors. For example, *the cup from the supermarket* should collapse to *cup*. In the object field, the determiners are removed. For example, removing determiners from *the book* gives *book*. Other rules are used to extract the appropriate object of interest. Using the rule *A of B gives B* as object, we get *flowers* as object from *bunch of flowers*. If noun is preceded by an adjective, we remove the adjective to give object *car* from phrase *red car*.

To extract the relevant action in a given phrase describing the action, we apply heuristic-based pattern extraction using Syntactic Tagging and Chunking. All this preprocessing is done automatically without human intervention. Preprocessing of data is shown on the left side of Figure 2.

Untrustworthy sources lead to possible wrong knowledge in distributed capture, and this is handled by Bayesian inferencing based on frequency of the data. Frequency gives us

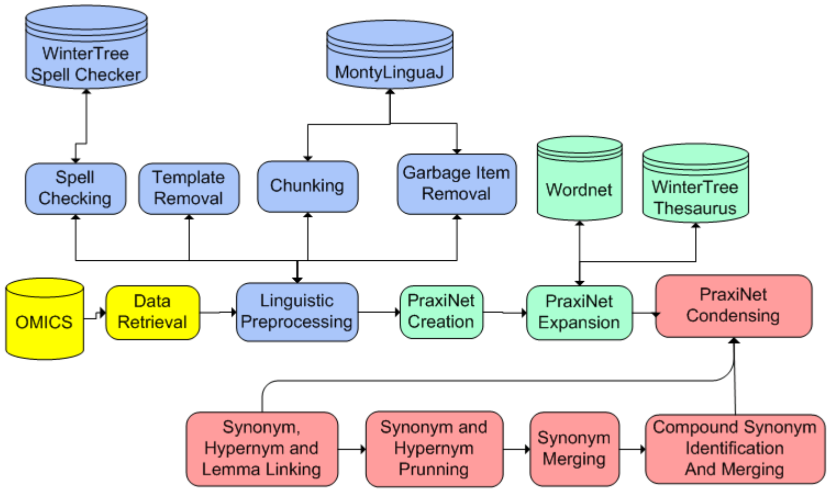

Figure 2: PraxiNet construction process

cumulative evidence to decide what is commonsense. Our system does not attempt to find the *right* response but the one that reflects the opinion of the majority. Given sufficient density of knowledge, wrong knowledge will appear as outliers, thus becoming irrelevant.

PraxiNet Creation

The PraxiNet is a directed labeled graph defined by $P = \{N, T, E, L, \alpha(x), \beta(y)\}$ where $N = \{n_1, n_2, n_3...n_x\}$ defines the node set, $T = \{t_1, t_2, t_3...t_x\}$ defines the node type set, $E = \{e_1, e_2, e_3...e_y\}$ defines the edge set, $L = \{l_1, l_2, l_3..l_y\}$ defines the edge label set, $\alpha(x)$ defines the type of a node such that $\forall x, \alpha(x) = t_x, t_x \in T$ and $\beta(y)$ defines the label of the edge where $\forall y, \beta(y) = l_y, l_y \in L$.

PraxiNet can also be construed as a multidimensional hyper-graph where each dimension is a graph where all the edges represent the same semantic type. Each dimension represents a type of semantic net. There are 24 possible relations in the OMICS database of which 10 are included in the current PraxiNet. Currently PraxiNet has the following 10 dimensions: has_object, has_action, has_property, response, patient, causes, synonym, hypernym, hyponym, lemmas. Figure 3 shows a sample region of the PraxiNet.

We can describe the structure of the PraxiNet in terms of node and edge types. The node types currently in PraxiNet are: object, property, action, situation and response. Currently, OMICS Causes and Responses generate the situations and responses in the PraxiNet. Cause edges connect situation to situation and Response edges connect situation to response. The edge types implemented in PraxiNet are: has_object, has_action, has_property, response, patient, causes, wordnet_synonym, wordnet_hyponym, wordnet_hypernym, and thesaurus_synonym. In the following sections, we describe the process regarding expansion and compaction of PraxiNet.

PraxiNet Expansion and Compaction

In OMICS, data sparsity is still an issue because of relatively small number of data entries (about 100,000) and use of multiple words to represent the same response or situation. Two people typically choose the same name for a single well-known object less than 20% of the time (Deerwester *et al.* 1990). It is also common for the same concept to be referred to by slightly different forms. For example, references to *floor dirty* and *floor unclean* should map to the same concept. We want to capture all information regarding the same concept together by condensing such information in an object centric way.

Data sparseness is addressed by expansion and compaction of the PraxiNet graph. PraxiNet attempts to increase the density of the net by using WordNet and the thesaurus semantic relations to establish new relations between existing nodes. We allow connections among existing concepts in the knowledge base that are hypernyms or synonyms of each other. Merging the synonym nodes increases the density of knowledge and makes the reasoning system robust to vocabulary differences among people.

For polysemous words with more than one different meaning (e.g. bank), the node will be represented as a weighted average of the different meanings. In our case polysemy is not an issue because the restriction on the domain heavily cuts down the number of terms with multiple relevant meanings in the domain.

WordNet and Thesaurus Expansion

Currently we are using three WordNet semantic relations for establishing new edges in WordNet—synonym, hypernym and hyponym. For each existing node in PraxiNet we extract the relevant words for these relations from WordNet. We create a new edge with the WordNet relation between the existing pair of nodes.

In expansion process, we add semantic links between existing nodes. For example, **synonyms** like infant and baby, **hypernyms** like knife and tableware, **hyponyms** like dog

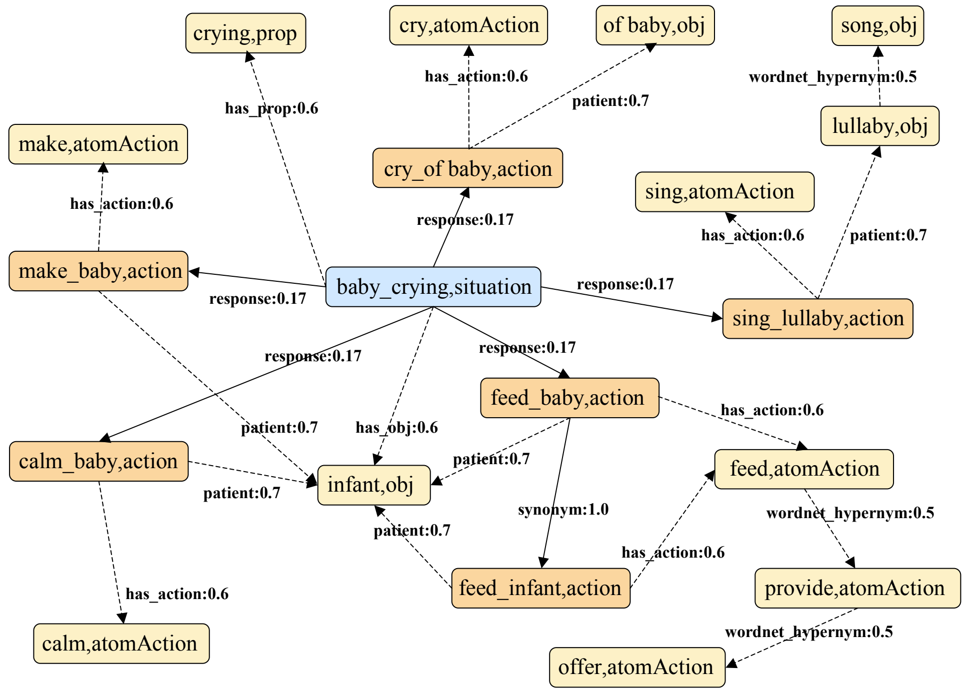

Figure 3: Sample region of the PraxiNet

and poodle are connected by new edges. In WordNet, only the first two synsets, direct hypernyms and hyponyms are considered to minimize noise. We also used the thesaurus from WinterTree software to create new synonym relations. MontyLingua is used to detect and add edges for lemma relations. Lemmatization maps word variants to the same feature (e.g. *dogs* and *dog* maps to same root word *dog*).

We perform expansion, pruning and merging operations in a multi pass approach. The advantage of multi pass approach is that it separates each of these steps, which simplifies the process of modifying either of them. Multipass means that we first extend the net as much as possible and then we prune it, one label at a time, followed by merging. There is no significant loss of speed from this approach, with the benefit of conceptual separation of each phase. This is described in the following subsections.

Synonym Pruning and Merging

After extending PraxiNet with these relations, we prune the uncertain synonyms and hypernym-hyponym pair edges, followed by collapse of valid synonyms and lemmas to further condense the net. We initiate the condensing process by pruning the synonyms and hypernyms that are not bidirectional (or hyponyms in the case of hypernyms) followed by the merge of the remaining synonyms. We do not collapse links between the hypernym-hyponym pairs in the current version.

It is important to maintain the original senses of merged words, So these are stored in hash table. For every pair of synonyms (n_1, n_2) we create an entry in the synonyms hash table $(n_2 \rightarrow n_1)$ and redirect the edges from n_2 to n_1. Figure 4 shows an example where *baby* and *infant* are merged as synonyms and replaced by one node in the graph.

Compound Synonym Identification and Merging

One of the main advantages of representing situations as compound nodes containing an object and a property is that by affecting one member of the compound nodes, the changes propagate to compound nodes. If two compound nodes have common intermediate nodes then they should be related. For example, if *baby* is synonym of *infant* then *baby crying* and *infant crying* should also be synonyms.

This is shown schematically in Figure 5. Here *baby* and *infant* are merged as synonyms and *baby crying* and *infant crying* are merged as compound synonyms.

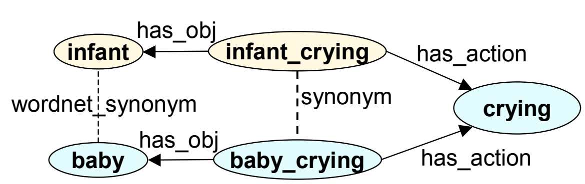

Figure 5: Compound relations example

As with compound synonyms, identifying the compound hypernyms is possible through the composing parts that have hypernyms. Currently compound hypernyms are found but not merged.

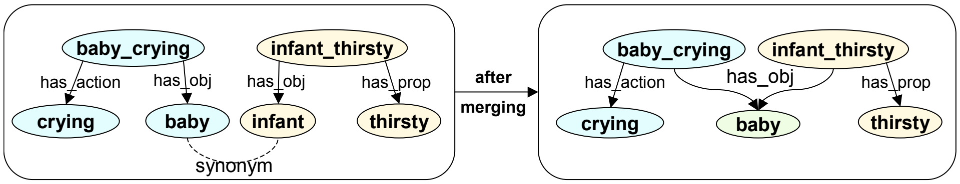

Figure 4: Example showing compaction of PraxiNet by merging of synonyms

Bayesian Network formulation

The goal of Bayesian inference is, given a situation, identify the most likely human response — in other words, the *commonsense* response to that situation.

Several methods have been used in the past for reasoning over semantic networks or graph oriented knowledge bases In LifeNet (Singh & Williams 2003), for example, reasoning was done using a local belief updating techniques with a *loopy* belief propagation model (Pearl 1988). Although it is an elegant technique, the reported time of several minutes makes it unsuitable for real time reasoning. In ConceptNet (Liu & Singh 2004) reasoning is done through graph traversal to find appropriate relations between nodes. Cyc (Lenat & Guha 1990) uses logical deductions for reasoning over its knowledge base. Spreading activation is another type of algorithm that is gaining renewed popularity for reasoning over graph-like structures in relational models (Bhalotia *et al.* 2002). These methods are successful in certain domains, but are unsuitable for our needs. Some of the mentioned algorithms are susceptible to node explosion, some are not real time, and neither is particularly sensitive to the possibility of starting from multiple nodes.

Other initiatives have tried to convert knowledge bases to Bayesian Networks, such as Wellman (Wellman, Breese, & Goldman 2002), Ngo (Ngo & Haddawy 1996) and Richardson (Richardson & Domingos 2003). Of these, only Richardson addresses the issue of creating a Bayesian network from distributed knowledge source rather than a hand crafted knowledge base. Furthermore none of these prior work describes an approach geared to real time reasoning, one of the focus of our algorithm.

It is likely that more than one situation is happening at the same time (e.g. the baby is crying and the room is hot). In such a scenario, we want to find the response that considers all the current situations. If no such response exists, we want to find one that satisfies a major part of current situations. Furthermore, we want the response in real time. Finally, as the number of nodes grows in PraxiNet, we need a robust algorithm that maintains near constant time.

For these reasons we decided to implement a localized Bayesian network as the basis of our reasoning method. Localized Bayesian network satisfy the handling of multiple situations, and near constant real time performance requirements. When multiple situations co occur, the goal becomes to find the best response that fits all situations. With a Bayesian approach we are able to boost the confidence of any response that can be reached by more than one situation, which maximizes the likelihood of selection of that response. Limiting the reasoning process to the relevant portion of the graph reduces the impact of the growth of the PraxiNet on the execution time allowing us to obtain a response in near real time for any node in the PraxiNet. This method is described in the following subsection.

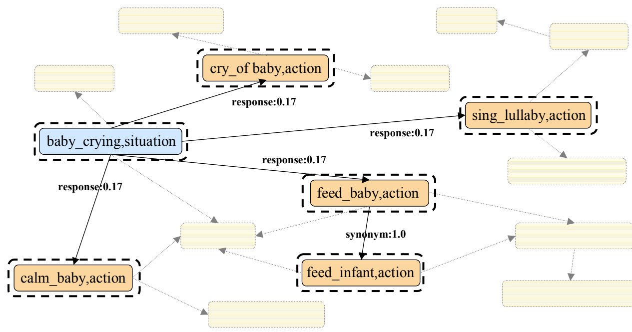

Figure 6: Selection of the relevant nodes to baby crying situation

Building the Bayesian Network

Given one or more situations, we start by identifying and selecting the corresponding nodes in PraxiNet as root nodes of our Bayesian network. We then proceed to collect all the associated nodes of the type *situation* or *response* and add them to the network as children of the root nodes. It is important to note that at this stage of the process we use only compound nodes because we already established all the compound level relations in the preprocessing stage. We use all the nodes connected by *cause*, *response*, *synonym*, *hypernym* and *hyponym*. We keep following the links until we have a subgraph that represents the local context, that is, all the responses reachable from the start nodes as illustrated in Figure 6.

We consider all synonym situations and synonym responses as representing the same knowledge respectively and merge them. Although it could be argued that this is not always strictly correct, restriction of the domain to indoor environments increases synonym knowledge overlap confidence.

At this point all the edges in the subgraph are either *response* edges or *cause* edges. We then raise the type of the edge to the lowest common semantic denominator, *leads to*. Basically, now the problem is formulated in the following manner: Given one or more situations as a start node, we have a Weighted Directed Acyclic Graph (WDAG) where the edges lead to either a *situation* node or an *response* node.

We are now looking for the response that has the highest probability of happening given the initial situations (our response). Instead of considering only direct responses to the initial situation, we are extending the domain to synonyms, hypernyms and responses to situations caused by the original nodes. Furthermore we are able to merge the different semantic edges into one simplified and elegant structure with which we can reason.

Inference using Bayesian Network

Since we are only looking for forward links in the original graph, we can assume independency between the children because we know that the initial nodes are true. The next and final step is to normalize the weights based on frequency of knowledge in the WDAG in the localized Bayesian network.

Figure 7 shows an example of combining causes and responses. The numbers show frequency of knowledge in OMICS database. The nodes should be read as a causal relation generating an expectation for nodes below. The calculations are as follows:

$P(S_1\text{=t}|S_0\text{=t}) = 0.6$
$P(R_1\text{=t}|S_0\text{=t}) = 0.4$
$P(R_1\text{=t}) = P(R_1\text{=t}|S_0\text{=t})P(S_0\text{=t})/P(S_0\text{=t}|R_1\text{=t}) = 0.4$
$P(S_1\text{=t}) = P(S_1\text{=t}|S_0\text{=t})P(S_0\text{=t})/P(S_0\text{=t}|S_1\text{=t}) = 0.6$
$P(R_2\text{=t}) = P(R_2\text{=t}|S_1\text{=t})P(S_1\text{=t})/P(S_1\text{=t}|R_2\text{=t}) = 0.48$
$P(R_3\text{=t}) = 0.12$

Considering the probability of all responses, the response to the situation *floor mat wet* is to choose the response with the highest probability which is *clean floor mat*. Other example situation-reponse pairs determined by the system include: response pick up telephone for situation telephone ringing; response calm baby for situation baby crying, response clean mug for situation dirty mug; response repair spectacles for situation broken spectacles. We currently have a fully functional system implemented that perform find the response in real time. The average response time is 0.1 seconds.

Evaluation and Results

Evaluating the validity of commonsense knowledge is a very subjective task. Given that the data in the OMICS database comes from human volunteers and each piece of data entered has been reviewed and manually verified to be valid by a reviewer, we have high confidence in the quality of our knowledge. If the initial data is overwhelmingly correct to begin with, then there is little to gain in trying to prove that the responses will make sense, because they will make sense insofar as the entries in OMICS database also make sense. So the question remains, how do we determine the degree of success of this approach?

Two indicators seem particularly relevant for the task at hand. First, how well are we able to convolute the different assertions representing the same knowledge together, and second can we retrieve an answer fast enough that could be used in real time.

The first indicator is particularly important due to widely different ways the same piece of knowledge can be referred to. In order to capture the commonsense response for a given situation we must be able to capture all the pieces of knowledge referring to a that particular situation. This is a challenging task due to the nature of language and its flexible power of expressiveness. The more we can conflate the knowledge and capture the different ways of saying the same thing, the more we increase the probability that the answer will reflect consensus.

The second indicator is necessary because if humanoid robots are to use this knowledge in the future, they must be able to do this in real-time. Creating fast ways of reasoning is an essential measure of the success of our approach.

Type	Raw	After Pre-process	After Expansion	After Compaction
Nodes	29049	24956	24956	23006
Edges	50397	47074	51215	48802

Table 1: PraxiNet Edge and Node Statistics

PraxiNet Statistics and Structure

PraxiNet is currently implemented in Java. Preprocessing, PraxiNet creation with expansion and compaction from OMICS database takes about 10 minutes (without any optimization). This has to performed only once whenever there is a new version of the knowledge base.

Currently we use only a small percentage of the data in the OMICS database. Of the 100,000 total entries, we had 9400 response entries and 900 causes entries. Half of the 100,000 entries are from the public version of the database that is freely available.

Table 1 shows how the number of nodes and edges vary during linguistic preprocessing and expansion and compaction of PraxiNet. During preprocessing, the number of nodes is reduced through spell checking and the elimination of non-essential components leading to merging of nodes. Entries that would create several nodes are now merged together. Fewer situations leads to a reduction in the edges since each situation has at least two edges. The number of nodes remains same while edges increase in PraxiNet Expansion, while both nodes and edges reduce during PraxiNet Compaction. During compaction, we have pruning of non-bidirectional edges, and removal of nodes and edges due to merging. 1782 synonym nodes and 168 lemma nodes are merged leading to an equivalent reduction in nodes and edges. In the expansion we only try to establish links between already existing nodes in the PraxiNet, so only the edges increase. During the compaction phase we prune edges and we combine nodes so we get a reduction in both edges and nodes.

Figure 8 shows variation of frequency of nodes in PraxiNet with number of words in a node. The plot shows a drastic reduction in the Nodal word length after the linguistic processing and the PraxiNet condensing. The main reason is the elimination of the non-essential components in nodes in the the linguistic preprocessing phase. Fewer words in a node increase the probability of finding a correlation with other nodes. So the reduction in the number of words in the

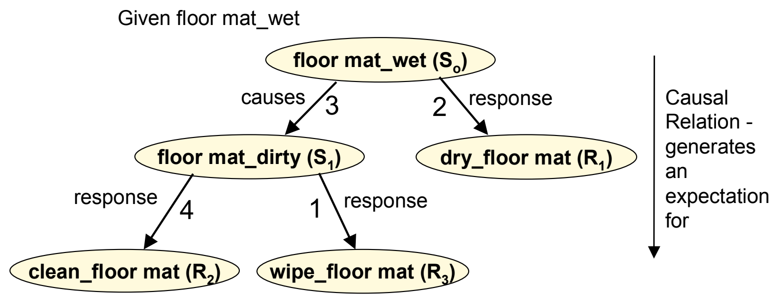

Figure 7: Bayes Inference in floormat example

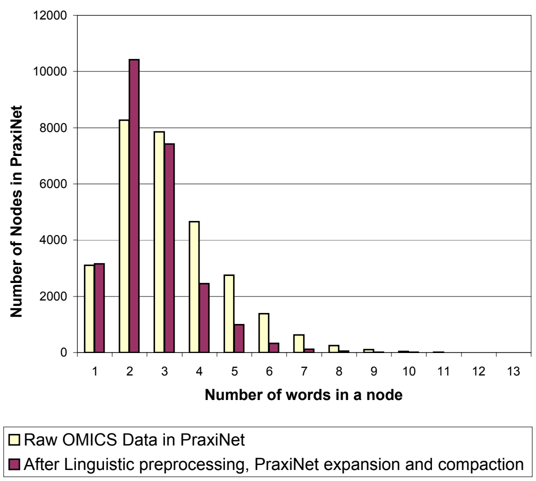

Figure 8: Complexity of PraxiNet nodes relative to number of words in the node

nodes contribute to cleaner and more manageable data.

The Figure 9 shows variation of number of outgoing edges from a node with expansion and compaction in PraxiNet. We observe that initially there are no nodes with one outgoing edge. This is because the initial edges connect the situation node to their objects and properties. Since a situation has at least one object and one property, we either have no outgoing edge or at least two. There is no change in number of one outgoing edges after linguistic preprocessing but a reduction in nodes with zero, two or more outgoing edges because of combining of nodes. In later stages, when we extend the PraxiNet and combine nodes, we start having nodes with one outgoing edge.

These curves reflect an increase in the PraxiNet density during its creation, which contributes to the increase in the amount of available evidence for a given situation/response pair.

Our current system contains a Java application where the Bayesian network can be displayed with computed probabilities of responses to the given situation. PraxiNet structure is based on JgraphT [1], a Java based graph library able to support millions of nodes effectively. JGraphT as been used in large efforts such as the Stanford Java NLP toolkit (Stanford 2005). The response determination to a given situation using this Bayesian Net runs in real time. Preliminary results point to an average iteration time of under 0.1 seconds to find a response to a given situation.

Conclusions and Future Work

In this paper we have shown how we can use distributed knowledge in conjunction with Bayesian reasoning to endow humanoid robots with the ability to respond to situations within their environments. Given a situation, the relevant part of the PraxiNet is extracted in a localized Bayesian Network to compute the response to a given situation. We applied these Bayesian Networks to reason in real time with promising results.

We used a distributed database created by collaborative effort, the OMICS database, to create PraxiNet, a multi-dimensional semantic network and used Bayesian Networks to reason over PraxiNet represnted knowledge. PraxiNet is used to represent distributed knowledge at the level of situations and responses. Density of data in PraxiNet is improved further using Natural language Processing and lexical resources like WordNet and thesaurus to find and merge synonyms to compact the PraxiNet. Conflicts in knowledge are handled by statistical techniques and Bayesian computation to tolerate errors in data as well as to determine consensus. Our approach is robust against noisy data which leads to scalability of our work to millions of pieces of knowledge.

In contrast to prior work, our work is the first to reason at the composite level of situations and responses. Prior work was geared towards general level commonsense relations and dealt with information at the objects and properties level. To our knowledge, this is also the first work to extract a Bayesian Networks from distributed knowledge for *real time reasoning*. In future, we would like to address is the identification of synonyms between large noun phrases using resources like WordNet. With more data that we will collect in the future, the density of PraxiNet will increase which will further improve the quality of our inference. We are also interested in the creation of learning algorithms for knowledge extension using dialogue and other commonsense databases like ThoughtTreasure.

[1] available at: jgrapht.sourceforge.net

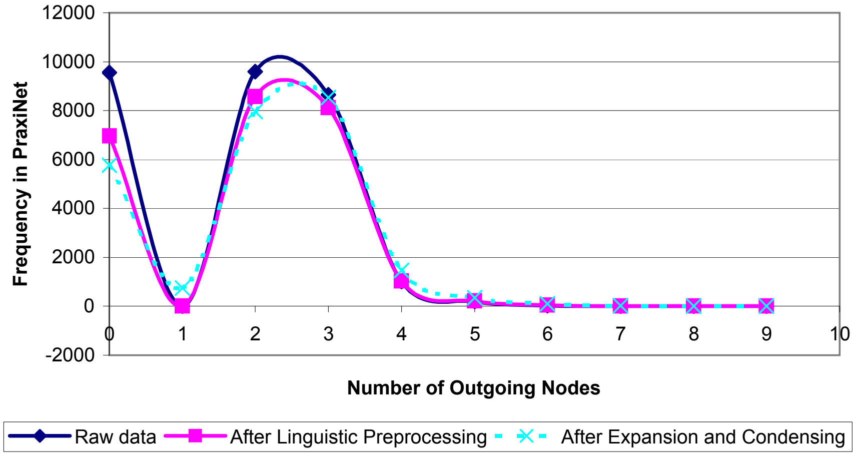

Figure 9: Variation of number of outgoing edges from a node with each step in PraxiNet processing

References

Bhalotia, G.; Nakhe, C.; Hulgeri, A.; Chakrabarti, S.; and Sudarshan, S. 2002. Keyword searching and browsing in databases using BANKS. In *Proceedings of the 18th International Conference on Data Engineering*.

Deerwester, S. C.; Dumais, S. T.; Landauer, T. K.; Furnas, G. W.; and Harshman, R. A. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6):391–407.

Fellbaum, C. 1998. Wordnet: An electronic lexical database.

Gupta, R., and Kochenderfer, M. J. 2004. Common sense data acquisition for indoor mobile robots. In *Nineteenth National Conference on Artificial Intelligence (AAAI-04)*.

Lenat, D. B., and Guha, R. V. 1990. *Building Large KnowledgeBased Systems: Representation and Inference in the Cyc Project*. Reading, Massachusetts: Addison-Wesley.

Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22.

Liu, H. 2004. Montylingua: An end-to-end natural language processor with common sense. Available at: web.media.mit.edu/ hugo/montylingua.

Mueller, E. T. 1998. *Natural language processing with ThoughtTreasure*. New York: Signiform.

Ngo, L., and Haddawy, P. 1996. A knowledge-based model construction approach to medical decision making. In *Proceedings of 1996 AMIA Annual Fall Symposium*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference*. Morgan Kaufman.

Quillian, M. R. 1967. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Sciences* 12:410–430.

Richardson, and Domingos. 2003. Building large knowledges bases by mass collaboration. In *Proceedings of the international conference on Knowledge capture*, 129–137.

Singh, P., and Williams, W. 2003. LifeNet: A propositional model of ordinary human activity. In *Knowledge Capture (DC-KCAP 03)*.

Stanford. 2005. Java nlp toolkit, available at http://www-nlp.stanford.edu/javanlp/.

Stork, D. G. 1999. The Open Mind Initiative. *IEEE Expert Systems and Their Applications* 14(3):19–20.

Wellman, M. P.; Breese, J. S.; and Goldman, R. P. 2002. From knowledge bases to decision models. *Knowledge Engineering Review* 7(1):35–53.