Learning Commonsense Human-language Descriptions from Temporal and Spatial Sensor-network Data

by

Bo Morgan

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006

Author .....
Program in Media Arts and Sciences,
School of Architecture and Planning
September 1, 2006

Certified by .....
Walter Bender
President, One Laptop Per Child;
Senior Research Scientist, Media Lab, MIT
Thesis Supervisor

Accepted by .....
Andrew B. Lippman
Chairman, Department Committee on Graduate Students

Learning Commonsense Human-language Descriptions from Temporal and Spatial Sensor-network Data

by

Bo Morgan

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on September 1, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Embedded-sensor platforms are advancing toward such sophistication that they can differentiate between subtle actions. For example, when placed in a wristwatch, such platforms can tell whether a person is shaking hands or turning a doorknob. Sensors placed on objects in the environment now report many parameters, including object location, movement, sound, and temperature. A persistent problem, however, is the description of these sense data in meaningful human-language. This is an important problem that appears across domains ranging from organizational security surveillance to individual activity journaling.

Previous models of activity recognition pigeon-hole descriptions into small, formal categories specified in advance; for example, location is often categorized as "at home" or "at the office." These models have not been able to adapt to the wider range of complex, dynamic, and idiosyncratic human activities. We hypothesize that the commonsense, semantically related, knowledge bases can be used to bootstrap learning algorithms for classifying and recognizing human activities from sensors.

Our system, LifeNet, is a first-person commonsense inference model, which consists of a graph with nodes drawn from a large repository of commonsense assertions expressed in human-language phrases. LifeNet is used to construct a mapping between streams of sensor data and partially ordered sequences of events, co-located in time and space. Further, by gathering sensor data *in vivo*, we are able to validate and extend the commonsense knowledge from which LifeNet is derived.

LifeNet is evaluated in the context of its performance on a sensor-network platform distributed in an office environment. We hypothesize that mapping sensor data into LifeNet will act as a "semantic mirror" to meaningfully interpret sensory data into cohesive patterns in order to understand and predict human action.

Thesis Supervisor: Walter Bender
Title: President, One Laptop Per Child; Senior Research Scientist, Media Lab, MIT

Certified by .....

Joseph A. Paradiso
Associate Professor, Media Lab, MIT
Thesis Reader

Certified by .....

Whitman Richards
Professor, Media Lab, MIT
Professor, Brain and Cognitive Sciences, MIT
Thesis Reader

Acknowledgments

This work is supported in part by a grant from *France Télécom R&D.*

My parents, Carolyn Spinner and Greg Morgan, have taught me how life is fun by providing an invaluable example. Marvin Minsky has taught me both directly and indirectly how to build models of thinking. The folks at Brambleberry have been invaluable for all of their productive conversations and ideas, especially Marguerite Hutchinson. Everyone in the Commonsense Computing and Electronic Publishing Groups have become an extended family: Walter Bender, surrounded by his motley crew of graduate students, surrounded by their UROP students. Thanks to Sandy Pentland, Wen Dong, and Jon Gips for access to the the wearable sensor-network MIThril data. Thanks to Josh Lifton for teaching me about sensor-networks and for helping me to interface my project with his Plug sensor-network. Thanks to Henry Lieberman for helping me to organize my thesis as a "slaying the dragon" story. This thesis has been inspired by and written in loving memory of a kind friend and a leader of the field of artificial intelligence, Push Singh.

Contents

Chapter 1

Introduction

1.1 Problem: Sensors are not meaningful to people

We are entering a world in which it will become common for sensors on everyday objects throughout the environment to report things like location, movement, sound, temperature, etc. to computers. Research has resulted in examples of sensor-rich environments, which are starting to evolve into real-world installations (McFarland et al. (1998), Moore & Kennedy (2000), Tapia et al. (2004), Fulford-Jones et al. (2004), Wyatt et al. (2005), Eagle & Pentland (2005), Lifton et al. (2005), Thiemjarus et al. (2006), Edmison et al. (2006), Luprano et al. (2006)) due to their lower cost, lower power, and smaller size—a trend that will continue.

Commonsense computing is a vision of computation where computers have the set of general knowledge and ways of reasoning that a given community shares, so that computers can have a deeper understanding of humans and become a more integral component of daily life. Sensors do not speak human-language and do not communicate using means that a human would consider commonsense; sensors generally produce unintelligible streams of numbers. We propose trying to make sensors speak human-language by using commonsense representations, such as the recently developed LifeNet representation, to perform context-expansion to understand full-

vocabulary (Mohri et al. 1998) human-language understanding of arbitrary sensor streams.

LifeNet (Singh & Williams 2003) is a model that functions as a computational model of human life and attempts to anticipate and predict what humans do in the world from a first-person point of view. LifeNet utilizes a commonsense knowledge base (Singh et al. 2002) gathered from assertions about the world input by the web community at large. In this work, we extend this commonsense knowledge with sensor data gathered *in vivo*. By adding these sensor-network data to LifeNet, we are enabling a bidirectional learning process: both bottom-up segregation of sensor data and top-down conceptual constraint propagation, thus correcting current metric assumptions in the LifeNet phenomenological model by using sensor measurements. Also, in addition to having LifeNet learning general commonsense metrics of physical time and space, it will also learn metrics to a specific lab space, the Third Floor of the Media Lab at MIT, and recognize specific individual human activities. These computational abilities will provide opportunities for making object-oriented spatial and temporal inferences, such as predicting how many people are in a given room and what they might be doing.

We hypothesize that the commonsense semantically related language-data that have been gathered from the public, such as the OpenMind Commonsense knowledge base, can be used to bootstrap quicker learning algorithms for classifying and recognizing sensor events and in turn common human activities.

1.2 Solution: Commonsense semantic knowledge bases

For technology to ultimately be helpful, it needs to be able to describe these events in terms meaningful to people; for example, expressing the coordinates from an accelerometer in a watch as the difference between shaking hands and opening a doorknob.

There are many applications and motivations for learning commonsense from raw sensor data and most of these relate to having a deeper ability to self-reflect on our world and the activities that are occurring. Whether we are wearing a sensor network and getting feedback throughout the day, or if we are reviewing the monthly online bulletin of occurrences within our local community, we are using this ability to self-reflect on our past in order to plan our future. Within this theme, we will discuss the general area of commonsense activity recognition as well as the more specific domains of body sensor networks that are worn throughout the day, such as cell phones and wristwatches, and also the environmental sensors that our body networks can use as an information infrastructure, such as audio and video recorders in buildings, temperature, humidity, and smoke detectors, etc. This polarity draws the distinction between the personal sensor network and the environmental sensor network.

1.3 Previous models: Too small and formal

Previous attempts at activity recognition force all descriptions into a very few categories specified in advance (Madabhushi & Aggarwal (1999), Luprano et al. (2006), (Edmison et al. 2006)); for example, location can be: at home, the office, or elsewhere (Eagle & Pentland 2005). Although these models perform very well at putting activities into these small sets of categories, these models don't adapt well to the very rich, dynamic, and idiosyncratic range of human activities.

1.4 LifeNet: A large adaptive first-person model

We developed a new representation called LifeNet as a representation for sensor understanding. LifeNet gathers descriptions of commonsense events from a large community of web volunteers, and creates a first-person model of events co-located in time and space. Given partial knowledge of a situation, LifeNet models commonsense expectations people might have about spatial and temporal context of the event. We can use

those expectations both for interpreting sensor data and learning new descriptions from the data.

LifeNet is a first-person commonsense inference model, which consists of a graph with nodes of commonsense human-language phrases gathered from OpenMind Commonsense (Singh et al. 2002), ConceptNet (Liu & Singh 2004), Multilingual ConceptNet (Chung et al. 2006) (English, Japanese, Brazilian), PlaceLab data (Tapia et al. 2004), and Honda's indoor commonsense data (Kochenderfer & Gupta 2004). Examples of commonsense knowledge from OpenMind include: "washing your hair produces clean hair"; "shampoo is for washing your hair"; "you can find shampoo in a shower"; etc. This knowledge is related in three ways: logical existential relationships, temporal probabilistic distributions, and spatial probabilistic distributions. LifeNet might infer that "I am washing my hair" *before* "My hair is clean." A **concept** is a human-language Unicode string representing a human-language phrase, which functions as the primary mode of indexing the ConceptNet reasoning algorithm. A **phenomenon** (Heidegger 1962) is a more general sense of the ConceptNet "text phrase" type of knowledge and forms the basic index to the LifeNet reasoning algorithm. The set of LifeNet phenomena includes all ConceptNet concepts as well as groups of sensor data. A recognized mode of text or sensor datum is a phenomenon functioning as a percept, while a contextual mode of text or sensor datum functions as a top-down projection phenomenon. A **commonsense phenomenon** is a mental state that a given "club" or group of people share; for example, a specific sensory experience that one might be able to express in conceptual human-language terms. Any given group of people will most likely share language capabilities that provide the ability to recall large sets of shared commonsense phenomena that are not necessary human-language concepts themselves. The connotation of the word phenomenon also leads the LifeNet algorithm closer to perceptual foundations in phenomenology and phenomenological, ontological understanding of perception, which is an underdeveloped branch of artificial intelligence that we hope to pursue in future research.

All of the reasoning in LifeNet is currently based on probabilistic propositional logic (Markov random fields and Bayesian mixtures of Gaussians); the benefits of

this design include: (1) probability eliminates the need to debug very large databases containing millions of strict logical relationships; and (2) higher-order logical representations, such as finite first-order logic and object-relative probability models, can often be compiled into a propositional form before inference routines are performed, so this compilation feature could be an extension to LifeNet. A basic visualization of the graph structure of LifeNet is shown in Figure 1-1.

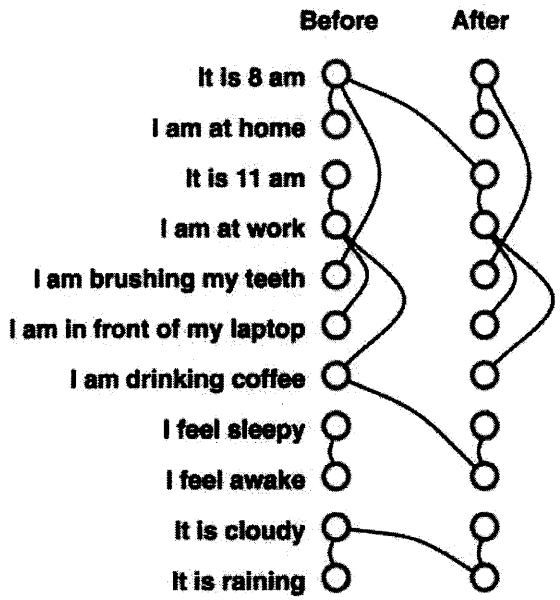


Figure 1-1: **LifeNet, a graph of nodes connected by probabilistic relationships** This dynamic Markov random field (similar to some dynamic Bayesian networks) consists of nodes of human-language phrases that are connected by tabular probabilistic relationships that is duplicated in two temporal slices that can predict sequences of events in simple sequential stories. The benefits of this simple view of LifeNet is that it is a very efficient and simple representation for computing the existence of phenomena; but one of the drawbacks of using only this representation is that it cannot calculate specific distances between events in time, and it does not predict where events will occur in space. This view is basically the same graphical representation of human-language nodes with probabilistic relationships specified as edges.

1.5 Performance evaluation

We evaluate the LifeNet critical reasoning algorithm on two very different knowledge bases: (1) Commonsense objects in a research office environment, and (2) Ubiqui-

tous Plug sensor-network platform audio streams. We demonstrate that using commonsense knowledge bases and inference tools, such as LifeNet, improves traditional bottom-up machine learning performance at understanding the human patterns inherent in these knowledge base. LifeNet can construct a mapping between sensor streams and commonsense stories. A **LifeNet story** is a partially ordered sequence of events expressed as conceptual human-language phrases (Unicode strings). Optionally, some of the sensor streams can be annotated with story events. LifeNet uses an analogy mechanism to perform logical belief propagation with a Markov random field and mixtures of Gaussians.

1.6 Contributions

The contributions of this thesis are:

1. The algorithm for making inferences over continuous temporal-spatial distributions;

2. The commonsense propositional representation (using first-person English phrases);

3. The connection to sensors to enable commonsense inference over sensor data;

4. The way analogy is being used to learn probabilistic inference is similar to analogies in Gentner (1983), but it is novel in the way it is being used to learn probabilistic-inference graph structures;

5. An algorithm that measures the similarity between two probability distributions using mutual information and modal similarity.

1.6.1 Future directions

We hypothesize that sensor data cannot be simply understood by machine learning algorithms that do not have a human-level language description of what is going on. In order to predict and understand human behavior in a sensor-rich environment, sensor-networks will need to incorporate frameworks like LifeNet that contain first-person

commonsense phenomenological models of human behavior. We also hypothesize that once human behavior can be explained by these top-down commonsense constraints, more specific commonsense patterns can be bootstrapped from this initial mapping of sensor data to human behaviors, leading to the codification and extraction of typical patterns of human behaviors, which would not be possible without the initial top-down commonsense language constraints.

Chapter 2

Problem: Sensors are not meaningful to people

Although, there is a technical problem with using all of this sensor technology for helping individuals to keep track of their daily lives: humans cannot naturally understand streams of sensor data, which often come in packets of cryptic numbers that are each labeled with their origin in time and space. The necessary problem of translating raw sensor data to a format that humans can understand involves mapping the data into forms that are natural for human understanding. One of the most powerful methods of natural social reference is some form of symbolic human-language, such as English. Other forms of human understanding, such as translating sensor data into a form natural to visual human understanding are also briefly discussed in this thesis, but areas for future research involve more immersive human experiences in sensor data. We hypothesize that one way to intimately extend this idea of **virtual existence** within sensor data will eventually include adaptive bidirectional neural interfaces to spatial and temporal world-model representations (Eden 2004).

We are entering a world where embedded-sensor platforms in everyday physical objects report things like location, movement, sound, temperature, etc. to computers. **Commonsense object** is an physical object that a human might use commonly to solve everyday problems, such as a "stapler" solving the common problem of keeping papers together in a document. Like all commonsense, these objects are specific to

the social cultures, groups and clubs to which the individual belongs. In general, a commonsense object is an object that all people within a context would consider to be a common everyday object. Sensors that detect temperature, acceleration, chemical composition, sound, as well as video are now very common, not only as a part of the working and public environments of our society, but also as part of our tools that we carry with us everywhere we go, such as cell phones, hearing aids, prosthetics, portable digital assistants (PDAs), laptops, etc. Creating networks of information-sharing applications that operate within security protocols have only begun to be established between this variety of systems as new protocols for low-power and short-range radio communication have become more emphasized in addition to global standards for how these devices communicate (e.g. I.E.E.E. (2003) personal low-power radio standard and other ISM radio bands). These international public frequency ranges and standards are not owned or controlled by national or corporate interests, so the individual citizen is free to use these ranges and protocols, and cell phones that run common computer languages are functioning as the current ubiquitous platform for these technologies. These sensitive and open-information environments allow the development of applications for the individual user that allow free access to the individual user to use these pervasive embedded sensor platforms in order to enhance their own self-reflection on personal activity patterns that may not have been obvious without such aids.

For example, a homeowner might find it inconvenient to look through endless video footage of many video cameras that they could potentially put around their home to identify a burglar after the fact of experiencing an illegal entry into their homes. We propose that one very natural way for people to understand sensor data is to provide human-language transcriptions of sensor data automatically. Given the ability to search through very large amounts of sensor data, such as all of the sensor data collected in a personal home through a human-language search capability is a natural way for people to perform a focused search within a very large corpus of collected sensor data about their personal life, which provides a powerful new means of self-reflection As neural interfaces become more intimate, this form of indexing

sensor memories by streams of symbolic phenomena, of which human-language is a subset, will become more natural and helpful experience-based memory indexing—one possible example of human-thought augmentation. human-language is not necessary for this memory indexing to take place, and one can imagine directly indexing previous experience directly by using a neural signal as the index to previous sensor data. Simply put, sifting through packets of numbered sensor data is not a natural or useful way for humans to interact with sensors.

2.1 Giving commonsense to computers

Commonsense computing is a vision of computation where computers have a general knowledge capability and ways of reasoning that are analogous to humans, so that computers then become a more integral component of daily life.

LifeNet (Singh & Williams 2003) is a model that functions as a computational model of human life that attempts to anticipate and predict what humans do in the world from a first-person point of view. LifeNet utilizes a commonsense knowledge base (Singh et al. 2002) gathered by the web community at large. In this work, we extend this commonsense knowledge with sensor data gathered *in vivo*. By adding these sensor-network data to LifeNet, we enable a bidirectional learning process: both bottom-up segregation of sensor data and top-down phenomenological constraint propagation, thus correcting current metric assumptions in the LifeNet phenomenological model by using sensor measurements. Also, in addition to having LifeNet learn commonsense metrics of physical time and space, it has also been used to learn metrics of a specific lab space, the Third Floor of the Media Lab at MIT, while recognizing specific individual human activities. Thus LifeNet is able to make both general and specific spatial and temporal inferences, such as predicting how many people are in a given room and what they might be doing.

The recent emergence of large semantic networks of human commonsense has led to a variety of applications. All of these applications provide an easier way for a human to interact with a computer through the semantic relationships between commonly

used and shared human-languages that have been gathered actively through online webpages. LifeNet is a reasoning system that uses spatial and temporal relationships within other commonsense knowledge bases (Liu & Singh (2004), Kochenderfer & Gupta (2004)) in order to begin to make guesses as to the positions and times of human-centered events. These relationships are initialized as being weak $\frac{1}{1}$ probabilities, but we will learn these relationships from data that is gathered from sensors embedded in experimental powerstrips that can sense nine different modalities. These are discussed in detail in the "Plug Sensor-network" Section in the Evaluation Chapter.

2.2 Ubiquitous computing

The eight prototypical sensor networks of the 1970s (Kahn 1978) and the origins of ubiquitous computing in the late 1980s (Weiser et al. (1999), Weiser (1991)) establish a vision of computation where computers are so deeply integrated into our lives that they become both invisible and everywhere. Realizing this vision requires building computer systems that exist in our environment and on our bodies; it poses two distinct directions for research: (1) the "human-out"—the influence of humanity's needs on technological developments; and (2) the "technology-in"—the influence of new technology on humanity. For example, the telephone can be considered as human-out by considering our social need to speak to one another; text messaging on cell phones can be considered as technology-in, since a new technology has affected the way that we express our humanity. Much sensor-network research emphasizes the technology-in direction; the work discussed in this thesis attempts to add models of human understanding to sensor networks emphasizing a human-out direction.

2.3 Top-down constraint hypothesis

Merleau-Ponty expresses a point relevant to this thesis:

> It is, then, diffused with a power of objectification, a 'symbolical func-

> tion', a 'representative function', a power of 'projection' which is, moreover, already at work in forming 'things'. It consists in treating sense-data as mutually representative, and also collectively representative of an 'eidos'; in giving a meaning to these data, in breathing a spirit into them, in systematizing them, in centering a plurality of experiences round one intelligible core, in bringing to light in them an identifiable unity when seen in different perspectives. To sum up, it consists in placing beneath the flow of impressions an explanatory invariant, and in giving a form to the stuff of experience.
>
> —Maurice Merleau-Ponty (Merleau-Ponty 1962)

Note that when Merleau-Ponty refers to as an "eidos" he stresses the importance of the structured forms that comprise the meaning of sensory data, which are also used in the activities of perceptual "projection"— top-down perceptual influences. These structural forms of perception are similar to Jepson & Richards (1994) "modes", and LifeNet uses the latter terminology to refer to the functional structures that provide generative explanations for the meaning of the data, and attempt at a generative meaning, as in the answer to the question: "How were these data generated?" Future research will deal with more complex generative processes as well as object-oriented approaches to try to answer more difficult questions, such as the following:

- "What generated these data?"
- "Who generated these data?"

Also, social and intentional object-oriented models of generative processes might provide hints toward how to answer the following questions:

- "What goals was this person pursuing while generating these data?"
- "What was this person thinking about that person when these data were generated?"

We hypothesize that the commonsense data that have been gathered from the public, such as the OpenMind Commonsense knowledge base, can be used to bootstrap quicker learning algorithms for recognizing sensor events and for classifying human activities. It has been shown that a two-slice bigram probabilistic model of sequential events in time can learn to predict commonsense data that is provided by sensors (Wyatt et al. 2005), given that the user is wearing an RFID reader on their wrist that can scan RFID labeled objects. Our approach does not use RFID tagged objects or a sliced Markov model of time, but instead considers time to be just another data dimension that can be reasoned over in the same way as any other sensor dimension, such as audio, power, or heat. In other words, instead of using a constant time-interval sampling technique, we use a technique of recognition that we call a Perception Lattice, which provides a way of recognizing phenomena of arbitrary lengths of time based on a symbolic sampling method (See Chapter 5). This method of recognizing phenomena of arbitrary lengths is reasoned over using mixtures of Gaussians, which are not limited to reasoning over fixed distances between phenomena in the same way that sliced bigram models are limited. We hypothesize that this ability for LifeNet to symbolically as well as numerically reason over the relationships between commonsense phenomena will allow humans to easily annotate numerical sensor data using symbolic human-language, allowing a process of self-reflection on the sensor data in their environment.

Chapter 3

Solution: Commonsense semantic knowledge bases

For technology to ultimately be helpful to the user, it needs to be able to describe these events in terms meaningful to a person. LifeNet has the ability to provide a human-language index into vast quantities of sensor data, allowing users to search through the histories of their homes and communities. In this section of this thesis, we present a prototype application of the LifeNet inference architecture: a cell phone diary application that allows users to annotate their body-sensor network data using a diary application. The knowledge for this diary application was seeded by the commonsense semantic knowledge bases that have been gathered from the web community at large ((Singh et al. 2002), Kochenderfer & Gupta (2004), Chung et al. (2006)).

3.1 Commonsense activity recognition

Because LifeNet has already incorporated millions of semantic relationships from other commonsense knowledge databases, the existing context that this semantic knowledge will provide in the learned relationships between sensor events will be the novel aspect of our approach to the problem of sensor network event recognition. Incorporating sensory data into LifeNet's commonsense knowledge will provide a rich source of temporal event sequences and concurrencies. LifeNet will use what limited

context it can infer from the raw sensor data in order to guide further sensing of the environment. This technique of using context to narrow the scope of the sensor network could focus the battery energy of the sensor network on specific sensor modalities at a certain times that would be important for a type of resource limited top-down inference to take place.

By way of example, let us consider a jogger that wants to use a device that can be carried or otherwise worn in order to remember a commonsense description in human-language of what is going on around her. Simple sensors do not tell us this information directly. Simple sensors on the human body can detect a number of dimensions of data in time, such as temperature, light level, sounds, vibrations, accelerations, and also electrical measurements (e.g. EKG, EEG, BCI, GSR, EMG). So, when she wants to see at the end of the day when she was "jogging" the system can respond with when the sensor data most likely reflects "jogging" as it is related to other commonsense concepts, which are in turn related to raw sensor data. As will be shown in Chapter 5, LifeNet can infer these relations from a small amount of supervised data. The remainder of this Chapter will focus on an online application for gathering such supervised data from cell phone users.

3.2 Cell phone diary application

The Reality Mining diary application (See Figure 3-1) provides large life-pattern self-reflection. The Reality Mining diary application was a project that augmented systems of data collection (Eagle & Pentland 2005) that runs on a cell phone, communicating with a centralized server in order to create a centralized database that consists of a person's daily activities correlated with textual commonsense descriptions of what they have done during that day. This information is displayed in the form of an online webpage diary that not only allows users to enter new information about themselves and descriptions about their activities, but also uses LifeNet in order to understand how the person has moved through space and time and what events would likely be the context of those events. An obvious benefit is that the user

can effectively search their daily lives for commonsense events of special interest. An example of a commonsense search might be "Tell me the time when I met that girl." The LifeNet commonsense engine would accept this query and augment the context "meet girl" with "party" and "drink wine." LifeNet knows that parties are loud and involve many human voices so searching for audio power spectra that contain a lot of energy as well as matching the power-spectrum signature of a collection of human voices. LifeNet will consider the related events in both human-language stories and raw sensor data in order to add even more context to this search. The LifeNet online cellphone diary allows these methods of automatic self-reflection and life summarization. Also, the diary allows a method for the user to add personalized commonsense information about their day that they would normally add to a conventional diary and that LifeNet would use in recognizing trends in other, analogous parts of that person's life.

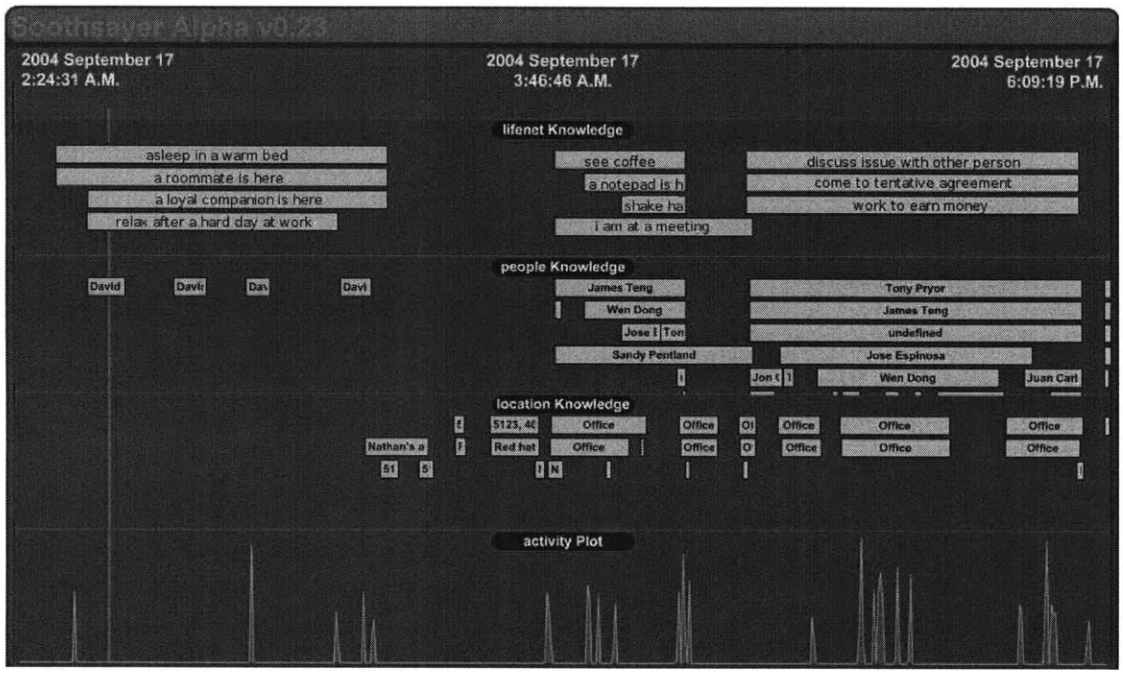


Figure 3-1: **Automatically generated cell phone diary application** This application was developed to show how people can use an online diary in order to interact with data that has been gathered by their cell phone throughout the day. 150 cell phones were programmed to log nearby cell phones (through Bluetooth identification) every 5 minutes,

and this information was presented along with approximate cell phone location identified by recognizing user-provided cell-tower IDs. Cell phone usage was also displayed as the bottom graph in this application. The relevant aspect of this application was the LifeNet interface (top), where the user could provide human-language text labels in order to create a commonsense diary. It could, in theory, be automatically generated, but the LifeNet inference routine was never fully connected to the reality mining project (Eagle & Pentland 2005).

Chapter 4

Previous models: Too small and formal

One of the major roadblocks facing full-vocabulary human-activity recognition is the fact that most models that are used to represent a human life are too *small* and too *formal*. For example, many models attempt to categorize human-activities into 3–30 categories, such as "I am outside" versus "I am inside", or "I am walking" versus "I am sitting." The development of body sensor networks that gather the data for these categorization activity-recognition applications has initially attracted the use of simple single-representation models of human-activity recognition. For example Eagle & Pentland (2005) have used Hierarchical Hidden Markov Models (HHMMs), and SenSys-151 (2006) have used Probabilistic Generative Grammars (PGG). Although these are very powerful probabilistic techniques, we hypothesize that developing reasoning architectures that reason about different aspects of the probabilistic model independently and incrementally combine their inferences will allow concurrent algorithms to propagate belief constraints about much larger state spaces than a single probabilistic representation would allow; the dynamic combination of many different representations and reasoning techniques will be necessary in order to represent and reason about the complexity of human life. LifeNet combines Markov Random Fields (MRFs) for reasoning about complex logical existential and truth relationships, Mixtures of Gaussians for reasoning about numbers in real-numbered dimensions such as

time and space, Dynamic Bayesian Networks for reasoning about sequences such as discrete time-steps, Vector Quantizing Neural Networks (VQNN) (Kohonen 1995) for the online learning of equiprobable symbolic mappings for streams of numerical sensor data, and our Temporal Perception Lattice (Functional Generative Grammar) for learning functional generative processes for perceptual data as a means for inference and recognizing similarity between very large sets of partially-ordered data.

4.1 LifeNet does not assume specific sensor types

LifeNet contains many different forms of learning algorithms for different fundamental types of sensor data. This allows us to use each of LifeNet's algorithms for processing more than one modality of data. Table 4.1 illustrates the types of data that LifeNet can process and what algorithms process these types of data along with the modalities that can be represented by each of these types.

Data Type	**Algorithm Type**	**Modality Type**
Partially-ordered Sets, Perception Lattices	Greedy Compression Search, Belief Propagation	Text Stories, Symbolic Sensor Streams
Real-numbered Tuples	Vector Quantizing Neural Network	Audio Power Spectra, Acceleration Power Spectra
Bayesian Nets, Markov Random Fields	Belief Propagation	Existence of Phenomena, Truth of Phenomena
Mixtures of Gaussians	Belief Propagation	Spatial Positions of Phenomena, Temporal Positions of Phenomena

Table 4.1: **The LifeNet internal data representation types and algorithms** These data and algorithms are very general in their applicability to specific modalities that LifeNet has been tested on thus far. Note the wide range of modality types that can be processed by a few core data and algorithm types. Because LifeNet can transfer probabilistic inferences and similarity calculations between different algorithm types, these interfaces can enable critical cross-modal learning between these reasoning systems relatively easily.

Previous work has focused on using specific sensors in order to perform activity recognition. For example, in Wyatt et al. (2005) 100 RFID sensors were placed on

a number of different objects within a house, while a wrist-worn RFID reader was used by the subject, while the subject performed one of 26 activities. This technique of activity recognition requires a large effort on the part of the user because the user must purchase and constantly wear a cumbersome wrist-worn RFID reader and all of the objects in their home must have RFID tags on them. This technique is a good technique for gathering data about what objects are used in commonsense human activities, but it is an awkward technique for gathering a large amount of commonsense data from many people. The Plug sensor-network (Lifton et al. 2005) is a good example of a possibility for a sensor-network that could be easily and cost effectively replace household power-strips. LifeNet is able to take advantage of the Plug sensor-network and other networks that produce streams of real-numbered or symbolic data distributed in space and time.

4.2 LifeNet does not assume specific temporal granularities or hierarchies

Many previous activity recognition techniques have focused on the use of Hierarchical Hidden Markov Models (HHMMs) in order to model sequences of events at multiple time scales. While this is a sound technique in theory, in practice the techniques used have considered only a fixed number of hierarchical levels of temporal granularity (usually 2–4); also within each of these levels, a fixed time-step length is used. Often also, as an additional assumption, these purely temporal hierarchical techniques assume a type of activity at each layer, such as a user's (1) goal, (2) trip, (3) transportation mode, and (4) edge transition (Liao et al. 2004). Others use multiple layers of Hidden Markov Models with each layer being a predefined fixed length of time (Oliver et al. 2002).

The advantage of not assuming a specific temporal granularity for the LifeNet architecture is that without this assumption LifeNet can answer questions about the a wide range of temporal scales in the same query. LifeNet can learn relationships

and perform inferences about temporal events on a wide temporal scale ($x : 10^{-308} < x < 10^{307}$ seconds).

4.3 Learning large numbers of natural language activities

LifeNet inherits large numbers of semantic language relationships from the ConceptNet projects (Liu & Singh (2004), Chung et al. (2006)). These projects bring more than 300,000 semantically related English phrases and 30,000 phrases in other languages (Korean, Japanese, and Portuguese) to LifeNet's range of textual phenomena. Also, the OMICS (Kochenderfer & Gupta 2004) knowledge base contains stories that link the English phrases together into 3,000 5–7 step sequences. Using this information along with partially labeled streams of sensor network data distributed in space and time gives us the learning and inference representations to develop algorithms for recognizing human-activities using an unlimited full human-vocabulary descriptions.

Projects have previously focused on developing learning algorithms that learn to categorize a small number of discrete categories (usually 3–30). For example Madabhushi & Aggarwal (1999) used their body sensor-network to categorize the activities "sitting down", "getting up", "hugging", "squatting", "rising from a squatting position", "bending sideways", "falling backward", and "walking." Eagle & Pentland (2005) developed an algorithm for activity recognition based on a cell phone as a sensor that categorized the locations of "office", "home", or "elsewhere." Luprano et al. (2006) developed a body sensor-network platform that could detect the activity categories of "resting", "lying", "walking", "running", and "going up or down stairs." Recently, Edmison et al. (2006) built a body sensor-network platform that detected the activities of "walking", "running", or "lying down." Also, Liao et al. (2005) have developed an activity recognition algorithm using a GPS sensor that not only predicts the location where a subject is, "home", "work", "bus stop", "parking lot", or "friend's house", but also predicts categories for how the subject transitions

from place to place, "walk", "drive", "visit", "sleep", "pickup", and "get on bus." Thiemjarus et al. (2006) used a sensor network to recognize different categories of 11 exercise activities including "sitting in a chair", "standing", "steps", "sitting on the floor", "demi-plie", "galloping left", "skipping", "galloping right", "side kick", "front kick", and "walking." As far as we are aware, this thesis is the first sensor-network activity recognition research that demonstrates algorithms that use full-vocabulary human-language to recognize human activities.

4.4 Using many reasoning critics to tractably reason over very large state-spaces

LifeNet has many different numerical and symbolic reasoning techniques in order to reason over not only semantic relevance but also contextualized existence, spatial and temporal relations with modal forms as well as functional generative similarity by using a Temporal Perception Lattice. This combination of many critical reasoning algorithms that judge the inferences of one another in iterative belief propagation between fundamentally different data representations is the key to LifeNet's ability to very approximately reason over the incredibly large state space (300,000 binary language variables results in $2^{300000} = 10^{90309}$ states) by relying on distributed critical algorithms using heavily contextualized evidence. Previous work that has dealt very effectively with a large state space was (Dong & Pentland 2006), who achieved relatively good categorization of subject activities, including eight location variables, six speaking variables, seven posture variables, and eight high-level activity variables resulting in their claim of being able to reason effectively over $8 \times 6 \times 7 \times 8 = 2588$ existential states. However, in order to not get caught up in number games, we emphasize that LifeNet's goal is to adapt to the language of the user and learn not only the sensor patterns of specific users but also the language that the subject uses to describe his or her activities to the system. Not only do people engage in very different activities, perhaps warranting the ability reason over very large state spaces, but also people are

probably only interested in a very small subset in the overall possible state space. For example, one user may be interested in when they are "eating in a restaurant", while another user may be interested in something completely different such as when she is "buying a cup of coffee." Subramanya et al. (2006) have recently combined spatial and temporal inference over GPS and partially labeled sensor data using an assumed map of labeled locations, but the activities that are recognized using this approach are limited to environmental context, "indoors", "outdoors", or "in a vehicle", and subject motion type, "stop", "walk", "run", "drive", and "up/down." The combinations of multiple critical reasoning algorithms operating in different modalities and incrementally combining the results of their individual inferences through belief propagation allows critical contextualized focus for the LifeNet reasoning algorithm that narrows the search to only consider an intersecting subset of the possible states that would be considered in individual modalities separately.

Chapter 5

LifeNet: A large adaptive first-person model

We use LifeNet as a representation for sensor understanding. LifeNet gathers descriptions of commonsense events from a large community of Web volunteers and creates a first-person model of events co-located in time and space. Given partial knowledge of a situation, LifeNet models commonsense expectations people might have about spatial and temporal context of the event. We use those expectations both for interpreting sensor data and learning new descriptions from the data.

LifeNet has been built as a piecewise-reasoning system in the tradition of the Society of Mind architecture (Minsky 1985), specifically the critic-selector model for emotional reasoning. A **critic-selector model** is a theory of how humans perceive, reason, and act. Minsky (2006) introduced the critic-selector model as a hierarchical implementation of the agents within the society of mind where critics and selectors are two specific types of agents. The model fits within a six-layered model of human intelligence, which has increasing levels of abstraction from the peripheral aspects of intelligence that interface directly with the physical world and the human sensations and motor-control. LifeNet is meant to function as a robust human-scale implementation of the most peripheral and lowest layer of this *Model-6* architecture, the *Reactive Layer*, which handles only the simplest levels of intelligence that do not rely on the self-reflective aspects of the subsequent layers, such as the *Reflective Layer* and the

Deliberative Layer. See Singh (2005) for an implementation of the lower three layers of the *Model-6* architecture in a very limited reasoning domain

The LifeNet algorithm is divided into different reasoning critics that process different types of data, while sharing constraining relationships between specific elements of data. For example, the spatial reasoning algorithm can infer where an object is most likely to be at a given time, while concurrently a temporal reasoning critic can check for asynchronicities in the spatial inferences. This type of concurrent constraint propagation between mental realms occurs between all critics in LifeNet. The shared phenomena that exist in more than one reasoning critic are referred to as the shared commonsense elements between these critics. These shared representational information structures allow for efficient information flow between probabilistic constraints that are processed in parallel (See Section 5.3.1 for details on automatically optimizing this information flow). The different reasoning critics that LifeNet is composed of are as follows:

- temporal-distance critic
- spatial-distance critic
- existential critic
- superficial temporal-redundancy critic
- sensor percept-alignment critic

These critics operate over knowledge bases that are specific to these critics and constraints between these knowledge bases provide the commonsense information flow between operating critics.

- sequential human-language stories
- sensor power-spectrum streams
- sensor stories

- analogical
  - human-language analogical stories
  - sensor analogical stories

Most of the relationships that are stored within LifeNet are arranged on the time, t, axis because this is the most general axis for considering sensor-data streams and human- text stories in any language, which is currently the primary application for the LifeNet reasoning system. LifeNet has also inherited the semantic knowledge from ConceptNet, so LifeNet can make general assumptions about distances in three-dimensional space, including the dimensions of longitude, latitude, and altitude, which are referred throughout this document as x, y, and z.

5.1 Converting real data to symbolic data

Although LifeNet uses a variety of real-numbered inference and learning algorithms, the abstract patterns that LifeNet is optimized to find in data are derived from partially ordered sets of symbolic data. The incoming sensor data, which is initially streams of real-numbers, must be converted to streams of symbols before LifeNet can find abstract patterns and in turn abstract similarities between these data. LifeNet uses a simple technique of considering a fixed-range power spectrum for each sensor stream and categorizing these power spectra into streams of symbols for subsequent stages of more abstract processing.

5.1.1 Computing power spectrum streams

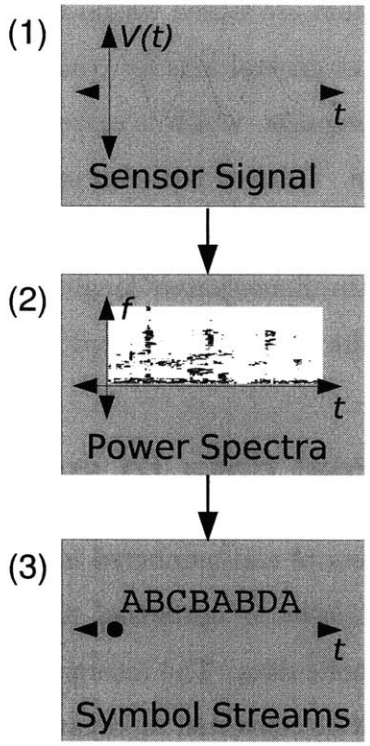


Figure 5-1: **Symbolic sensor percepts** These percepts start as streams of potential energy (voltages) (1) that exist with 100% certainty at specific points in space at specific points in time. These potential energies are filtered through a **fast Fourier transform (FFT)** algorithm, which performs approximately the same function as the human cochlear hairs that transduce resonant signals to the human brain in order to detect specific frequencies. This frequency power spectrum (2) is computed in time and is fed into an online-learning algorithm that is also modeled after the human perceptual system called a Kohonen neural network (Kohonen 1995), which turns the flow of frequency power spectra into a given number of categorical percepts that provide us with a stream of symbols (3) that the LifeNet analogical redundancy critics can operate over. This symbolic manipulation of sensory data operates over both commonsense language streams as well as sensor-signal streams in this way.

In order to compute a power spectrum stream, a frequency bandpass range chosen for each different sensor type, resulting in a set of frequencies with associated power amplitudes that change over time. For example, the Plug (Lifton et al. 2005) audio-stream data was high-pass filtered at 20 Hz and low-pass filtered at 1000 Hz. A fast

Fourier transform (FFT) was performed on non-overlapping windows of audio data in order to generate power spectra for each time-segment window of audio data. This audio data was then saved to a bitmap file with the horizontal axis representing time and the vertical axis representing frequency. Each horizontal pixel in this bitmap represents 1/16 of a second—time increasing to the right—and each vertical pixel represents each integral frequency band from DC to 512 Hz—frequency increasing from top to bottom. The intensity of each pixel is calculated as the power, $P = A^2$ (amplitude squared), of the voltage signal induced by the sound waves. See Figure 5-1 for an overview of the process of converting numerical sensor streams to symbolic streams.

5.1.2 Biological inspiration for using power spectra

Biology exploits power spectra throughout neural perceptual systems (Rosenweig et al. 1999). For example, a similar usage of frequency power spectrum analysis is in the human cochlea, where hairs called stereocilia select for specific frequency ranges depending on frequency penetration depth within the tapering cochlea. A similar but not nearly as sensitive frequency specificity of sensation appears in the tactile or somatosensory afferents of the skin, where four main skin receptors operate based on different temporal and spatial frequency ranges, resulting in responses to high-frequency and low-frequency vibrational stimuli as well as the high-frequency and low-frequency spatial distribution. Color vision is also a type of frequency power spectrum analysis usually using three (and sometimes four in tetrachromats) visual receptors that are each tuned to become excited within specific frequency ranges of the electromagnetic power spectrum. These biological examples of how power spectra provide an important aspect of the human perceptual system have encouraged us to implement basic power spectra recognition as a part of LifeNet's sensor stream recognition process.

5.1.3 Converting power spectra to symbols

In order to turn analog sensor power spectra into symbolic streams of data, which are directly compatible with the analogy matching algorithm within LifeNet, an unsupervised categorization algorithm—one of Kohonen's earlier neural networks, a vector quantization neural network (VQNN) (Kohonen (1995), Hecht-Nielsen (1990)), is used. VQNNs are also referred to as unsupervised density estimators or autoassociators and are closely related to k-means clustering. Actually, Desieno's version (Warren 2004) of Kohonen's VQNN algorithm was used in LifeNet. Desieno's version includes "Desieno's conscience" factor to ensure equiprobability of clusters independent of possible distribution irregularities. The VQNN was used for a number of reasons:

1. incremental online learning;
2. human mental compatibility;
3. simple to implement;
4. efficient when implemented in hierarchical binary-decision-tree form.

An incremental online learning algorithm is important for an application that must be deployed into an environment where the algorithm must adapt to learning the current sensor power spectrum surroundings. The human mental compatibility of the LifeNet algorithm will become more important as LifeNet becomes a better first-person model of human mental activity through modeling and mapping low-level neural activity with high-level human-language concepts and stories. Perhaps LifeNet can eventually function as a mental prosthesis for storing, indexing, and recalling personal memories. A shared mental commonsense of neural activities could begin to be recognized as long as artificially intelligent algorithms maintain a mental compatibility for understanding human minds.

The basic VQNN algorithm consists of k codebook vectors that are initialized to random codebook vectors that represent cluster centers that will be learned from a stream of sample vectors. For each sample vector in the stream, the closest codebook

vector is found according to a given distance function. LifeNet uses a Huffman-distance function or a metropolis-distance function, $\sum_i |x_i - c_i|$, where x is the presented vector and c is the codebook vector. LifeNet makes sure that every node gets a chance to "win" an equal number of data samples. The formula for updating a codebook vector to be a weighted average of the codebook vector and the training sample is

$$c_{\text{new}} = c_{\text{old}} \frac{N}{N+1} + x \frac{1}{N+1}, \tag{5.1}$$

where N is the number of samples that have already been learned by this codebook vector. This update rule ensures that the codebook vectors are equal to the average of all samples they have learned. LifeNet uses Desieno's "conscience" factor to sometimes overrule the user-provided distance metric such that a codebook vector cannot "win" more data samples than another codebook vector, ensuring the equiprobability of the clusters in the limit of infinite training time—LifeNet uses 10 training passes over the data, which appears to generate usable symbolic categories for our purposes.

The fact that VQNN is simple to implement allows it to easily be implemented as part of more efficient algorithms, such as the hierarchical binary-decision-tree implementation within LifeNet. VQNNs can be placed into a tree structure that can be used for either training or pure categorization, both processes gain the advantages of the structure of the tree, so that to perform a categorization among $n = 2^{10}$ in a previously trained VQNN only 2×10, $O(\log n)$, distance comparisons must be made. This tree is composed of a simple a root node, which is a flat VQNN with two codebook vectors with each of these two nodes containing either another two-node VQNN or a terminating leaf state.

5.2 Measuring abstract similarity

Measuring similarity between two pieces of data can be as simple as checking equality or can be as abstract as considering the motivations of the people that may have created that data. We introduce a representation called a **perception lattice** (See Section 5.2.3) that allows us to efficiently calculate the mutual information between

two arbitrary pieces of data within the LifeNet reasoning algorithm.

5.2.1 Learning to recognize semantic relationships

Perception is the process of explaining the data: what caused the data and what process generated the data. Data are explained by their generative causal regularities, which is to say their consistent relationships with other types of data. These regularities or patterns describe ranges or modalities of perception. A **mode of perception** is a consistent relationship between elements within a subset of data. This consistent relationship defines a regularity that serves to organize a larger set of data. A mode of perception can act as a manifold in the space of percepts such that percepts lie on a specific manifold or they lie off of that manifold. Given that a set of data points lie on a mode of perception, these data points contain percepts that vary along the mode. The specific percepts that define these data points are implicitly constrained by this manifold or mode.

Jepson & Richards (1994) have developed a Bayesian formulation of the modes of the probability densities of image features, which they have described briefly in the following excerpt:

> Our framework for understanding percepts is based on recognizing that certain image structures point reliably to particular regularities of properties in the world with which we are familiar and expect in certain contexts. In other words, these regularities have high priors in that context. ... We regard these properties as special, in that their probability density functions are "modal", whereas in contrast [other] properties ... have broad density functions.

An example of a modal percept in text processing is the generative function

$$\mathcal{G}_{is\text{-}a}(x, y) = \text{“The ”} + x + \text{“ is a ”} + y + \text{“.”} \tag{5.2}$$

is a common modal textual pattern that is used in much of the natural-language-processing community. Liu & Singh (2004) have built a semantic language network

called "ConceptNet" that contains a generalization of this modal structure that includes all forms of the verb "to be" as well as other linguistic techniques (e.g. "is goal of") that allow a general type of semantic relationship between the conceptual text variables x and y in Equation 5.2. In ConceptNet this is referred to as the *is-a* semantic relationship. All semantic relationships in ConceptNet are binary relationships (only take two variable arguments [x and y in this case]). ConceptNet is limited to 20 different types of hand-programmed types of English semantic relationships, which are not limited to specific parts of speech such as verb or preposition relationships, but instead represent more abstract "Mad-lib" relations; for example,

$$\mathcal{G}_{\textit{is-used-for}}(x, y) = \text{“A ”} + x + \text{“ is used for ”} + y + \text{“.”} \tag{5.3}$$

See Appendix Section B for a complete list of the ConceptNet relations. We propose a theory of modal perception lattices for posets (partially ordered sets) in order to generalize the ConceptNet relation to take any number of arguments (an N-ary relation) and also support the automatic learning of these perception lattices from a raw-text corpus of activity stories consisting of lists of common goal-oriented actions in human-language. We demonstrate the effectiveness of our learning similar modal relationships with sensor streams that have been categorized into symbolic posets as well. In other words, these sequential modes are machine-learning tools that could be compiled into forms that are as efficient and easy to use as hand-coded ConceptNet relations.

Orderings of data representations within LifeNet (Singh & Williams 2003) inference, such as Unicode strings on an arbitrary axis (e.g. "time") is one of the primary functions that LifeNet serves in considering how data is arranged relative to one another, while also considering all contextual dimensions for knowledge in whatever axes the contextual relationships for data are provided. Axes' names are allocated dynamically by the LifeNet algorithm, so at any time a user can specify new data in a relationship along a new axis and those data will then be reasoned about in those dimensions.

5.2.2 Recognizing generative functions

A **generative function** is a computational process that takes a set of arguments and returns a set of data derived from those arguments. As computational processes, generative functions assume an algebra of data processing computation. This generative functions in this paper assume an algebra of stacked poset concatenation functions, which are easily implemented on classical digital computers, but in general, generative functions could assume biological neural networks or quantum computers as other algebras of data generative computation.

Measuring relative similarities and differences between arbitrary pieces of data depends on the generative modes that define the structure of the larger context of the pieces of data in question. Perceptual modes can be used for a number of tasks, including the detection irregular data in the context of regular data. This detection of irregular data can be used to direct the focus of a learning algorithm that is trying to develop a model of the regularities of a given set of generated data. This assumption that the data is generated is in fact a very large assumption, especially once we give a definition for generated data, but for the domain of the problem where LifeNet is applied, the assumption of generated data is argued to be a good one: different objects generate different patterns of sensor-data; different language is generated by different structural patterns of language.

5.2.3 Searching for optimal generative functions

Optimal generative functions are only optimal relative to a knowledge base and how well the function serves to reduce the overall number of bits necessary to functionally generate the knowledge base. For example, the following set of two arbitrary data, $K = \{$"The sky is blue.", "The house is green."$\}$, have the following shared generative function:

$$K_{G_0} = \Lambda(x, y) \text{"The "} + x + \text{" is "} + y + \text{"."} \tag{5.4}$$

This generative function is chosen greedily with a heuristic. The heuristic estimates the compression of the the overall knowledge base that would be accomplished by re-

membering this specific generative function. Only one variable is supported currently in the search for this generative function. It would be nice to have more than two variables in the generating functions found by this search, which could be done by searching down one level through larger patterns in the lattice and then searching for similar parent patterns within nodes in that deeper functional layer.

If we find K_{G_0} to be the highest heuristically ranked ($K_{G_{r=0}}$) generative function for the knowledge base, K. We can remember the generative function, K_{G_0}, and the following sets of argument data:

$$K_{G_0 A} = \{\{\text{“sky”}, \quad \text{“blue”}\}, \\ \{\text{“house”}, \quad \text{“green”}\}\}$$

With the generative function, K_{G_0}, and the arguments list, $K_{G_0 A}$, the initial knowledge base can be recreated or generated functionally:

$$K_{G_0}(\text{“sky”}, \text{“blue”}) = \text{“The sky is blue.”}$$
$$K_{G_0}(\text{“house”}, \text{“green”}) = \text{“The house is green.”}$$

These generative function explanations create hierarchical lattices of generative functional explanation for perception data. A **perception lattice** is a lattice data structure that represents the generative functional explanation for a given set of data. This lattice structure is used for many algorithmic operations over perceived data. For example, a perception lattice could be used to find the most likely top-down explanation of bottom-up perceptions, or alternatively, a perception lattice could be used for the projection of low-level details given high-level evidence. We use the term “perception lattice” very similarly to the *structure lattice* in Jepson & Richards (1994) except for the philosophical differences discussed in Section 5.4 regarding the objective duality between the observer and the world and how to interpret these as self-reflective computational processes.

5.2.4 Temporal modes

The process of calculating mutual information between two pieces of data requires certain assumptions of the generative processes that could have created these data. Our generative model assumes a stacked functional model similar to a generative grammar. All of the graphs that this method currently compares and abstracts into function/argument temporal forms are hierarchical trees. In calculating analogy structures, rather than only using the nodes and edges of a predefined semantic network (Gentner (1983) and Falkenhainer et al. (1989) similarity method), or just the edges (Liu & Singh (2004) ConceptNet method), the method that LifeNet uses compares mutual information between nodes by considering their generative functional structures. These generative functional structures are generalized edge types that are learned based on the given computational algebra, which is in this case a stacked functional language of simple concatenation functions. We assume that Unicode strings within ConceptNet are created by a hierarchy of generative functions, which as a whole can be structure mapped against other ConceptNet concepts resulting in analogical mappings that result in variable mappings that generalize the idea of a binary ConceptNet relation to an N-ary LifeNet relation. N-ary LifeNet relations are referred to as cliques rather than as edges, which are binary cliques. These generalized forms of ConceptNet links can, for example, recognize sentence forms, such as:

$$\mathcal{G}(A, B) = \text{“The ”} + A + \text{“ is a ”} + B + \text{“.”} \tag{5.5}$$

These generative functional structures can be used to calculate mutual information between two streams by considering the execution of a generative function to be an event that occurs in the context of given argument values. The context of the argument values provides a probability distribution over possible generative functions for each branch in the hierarchical generative function structure. Once this probability is defined in terms of specific probabilities for each structural change to a generative function structure, a mutual information distance function can be defined between each pair of data. LifeNet quickly calculates the probabilities for the generative

function execution events by using a perception lattice.

LifeNet has the ability to recognize these generative function structures for temporal sequences using a limited set of functions that involve different orderings of the string concatenation function. LifeNet cannot yet recognize generative functional structures for spatial relationships.

5.2.5 Learning perception lattices from data by greedy compression search

A **greedy compression search** is a search algorithm that begins with a list of uncompressed data, L. For all of the data in L the largest contiguous repetitive section, x, of data is found. Every datum in L containing x is removed from L and split into smaller non-contiguous pieces that do not contain x. These smaller non-contiguous pieces are appended to L, and the process of removing redundant sections of data continues until no such sections exist in L, at which point L will contain the leaves of the perception lattice structure.

Simple examples of the perception lattices resulting from this greedy compression search algorithm are shown in Figure 5-2.

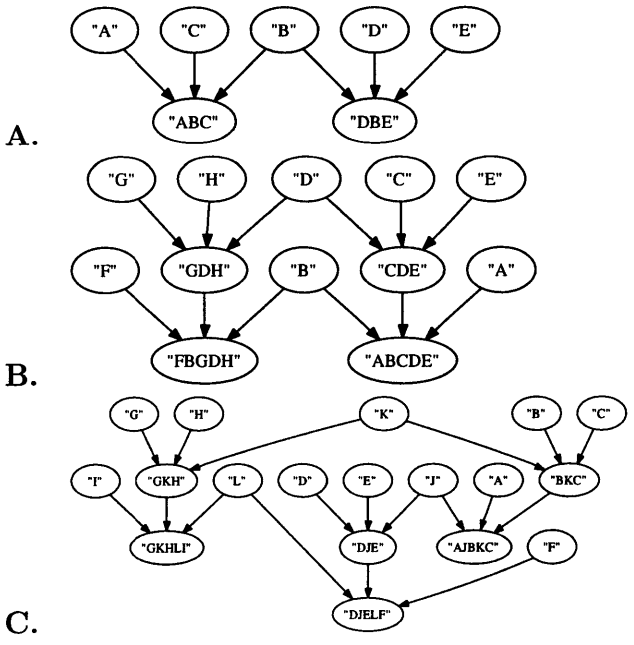


Figure 5-2: **Simple perception lattices** The lattices **A**, **B** and **C** are found by greedy compression search. The lattice in **A** is generated from the two strings "ABC" and "DBE", which contain only the character "B" as similar data. The lattice in **B** is generated from the two strings "ABCDE" and "FBGDH", which contain the characters "B" and "D" as similar data. The lattice in **C** is generated from the three strings "AJBKC", "DJELF" and "GKHLI", which share a triangular relationship in the characters "J", "L" and "K" as similar data.

Figure 5-2.A is a simple perception lattice that contains a single generative function,

$$\mathcal{G}_A(x, y) = x + \text{“B”} + y, \tag{5.6}$$

such that

$$\mathcal{G}_A(\text{“A”}, \text{“C”}) = \text{“ABC” and}$$
$$\mathcal{G}_A(\text{“D”}, \text{“E”}) = \text{“DBE”}.$$

Notice that the because the compression search used to create these perception lattices is greedy the generative functions that are found contain a maximum of one repeating phrase, which implies a maximum of two arguments. We have chosen a greedy compression search as an efficient proof of concept algorithm that operates over very large knowledge bases quickly (e.g. 3000 stories each containing approximately 1000 characters in the OMICS story knowledge base and 300,000 sentences each containing approximately 50 characters in the ConceptNet knowledge base). Figure 5-2.B is another simple perception lattice that contains two generative functions,

$$\mathcal{G}_{B_0}(x, y) = x + \text{“B”} + y, \text{ and}$$
$$\mathcal{G}_{B_1}(x, y) = x + \text{“D”} + y,$$

such that

$$\mathcal{G}_{B_0}(\text{“A”}, \mathcal{G}_{B_1}(\text{“C”}, \text{“E”})) = \text{“ABCDE” and}$$
$$\mathcal{G}_{B_0}(\text{“F”}, \mathcal{G}_{B_1}(\text{“G”}, \text{“H”})) = \text{“FBGDH”}.$$

Similarly, for the simple perception lattice in Figure 5-2.C the generative functions are

$$\mathcal{G}_{C_0}(x, y) = x + \text{“J”} + y,$$
$$\mathcal{G}_{C_1}(x, y) = x + \text{“L”} + y, \text{ and}$$
$$\mathcal{G}_{C_2}(x, y) = x + \text{“K”} + y,$$

such that

$$\mathcal{G}_{C_0}(\text{“A”}, \mathcal{G}_{C_2}(\text{“B”}, \text{“C”})) = \text{“AJBKC”},$$
$$\mathcal{G}_{C_0}(\text{“D”}, \mathcal{G}_{C_1}(\text{“E”}, \text{“F”})) = \text{“DJELF”}, \text{ and}$$
$$\mathcal{G}_{C_1}(\mathcal{G}_{C_2}(\text{“G”}, \text{“H”}), \text{“I”}) = \text{“GKHLI”}.$$

5.2.6 Efficient context-dependent mutual information calculation using generative functional grammars

LifeNet's most basic atomistic representational component is a symbol, and these symbols occur in temporal streams. A power spectrum clustering algorithm provides streams of symbols to LifeNet. Also, the story data also provides streams of textual symbols (individual Unicode characters). The greedy compression search results in a perception lattice that at its leaves contains the atomistic posets of the experience. The leaves of the perception lattice (the parents in Figures 5-2 and 5-3) are where the lattice interfaces with the external world. For example, if we would like to use the perception lattice to find an explanation for how a given poset, x, was generated, we would begin by checking which leaves in the perception lattice were used to functionally generate the poset, x. Those leaves, the few atomic posets of all experience, then form the basis for a search algorithm that flows down the generative functions of the perception lattice; note that this flow downwards in the perception lattice requires that the generative functions be reversible. Piaget (1947) emphasizes how this form

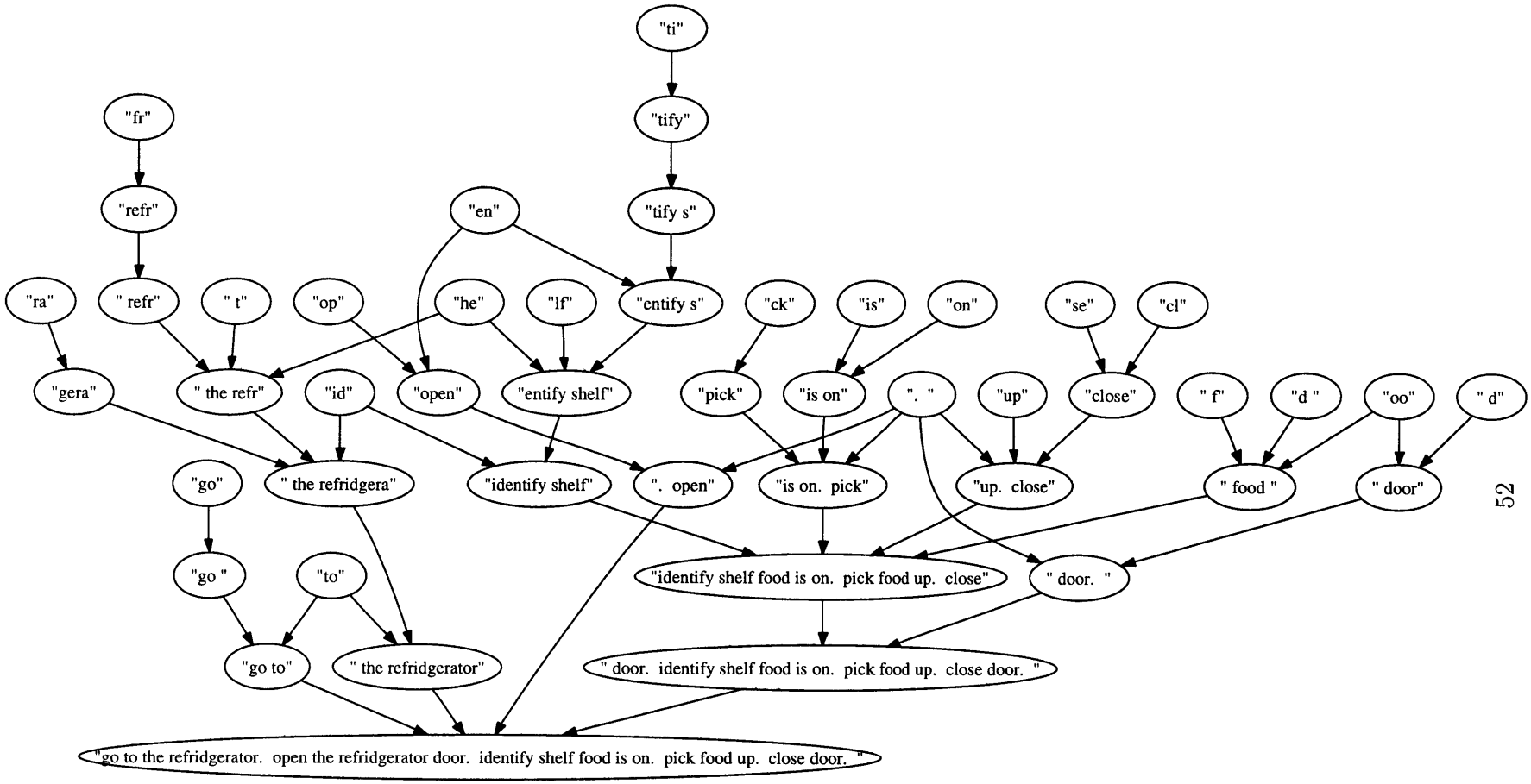


Figure 5-3: **Example of a perception lattice** This lattice was derived using just the information in this five step text story using a simplistic greedy compression search. Perception lattices like this one can form structures for very efficient data processing algorithms to be built. For example, a decision tree can be compiled that recognizes longest substring matches, or a belief network can be constructed to recognize hierarchical data patterns. Phrases of length one are omitted from this visualization.

of reversibility is a fundamental aspect of "successive adaptations of a sensori-motor and cognitive nature" of intelligence in the following excerpt:

> [Reversibility], as we shall see, is the essential property of the operations which characterize living logic in action. But we can see straight away that reversibility is the very criterion of equilibrium (as physicists have taught us). To define intelligence in terms of the progressive reversibility of the mobile structures which it forms is therefore to repeat in different words, that intelligence constitutes the state of equilibrium towards which tend all the successive adaptations of a sensori-motor and cognitive nature, as well as all assimilatory and accommodatory interactions between the organism and the environment.

LifeNet compares the symbol streams and calculates the similarity of these streams by using a form of mutual information. The calculation of mutual information measured in bits is as follows:

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}. \tag{5.7}$$

Since our primary generative function is simply the string concatenation function, we have implemented a very efficient means of calculating the mutual information between two arbitrary streams of data. Calculating extremely fast mutual information comparison functions that are experience dependent and context dependent is an extremely important and central task to building artificially intelligent computer systems that learn to perceive and act in the world.

The mutual information between two sets of posets, X and Y, depends on the modes that make up those pieces of data relative to the perception lattice that has been generated from the universal knowledge base. The mutual information between sets of posets X and Y can be calculated by searching for which posets in the perception lattice exist within X and Y separately. Let us refer to these posets that exist within the perception lattice and within X and Y as $\hat{X}$ and $\hat{Y}$ respectively. The calculation of Equation 5.7 directly with $|\hat{X}| \sim |\hat{Y}| \sim n$ takes time $O(n^2)$ and the calculation

of $P(x, y)$ is non-trivial so this is actually a very conservative estimate; however, if we take advantage of Bayes' rule and make a few reasonable assumptions, such as treating the perception lattice as a probability network, over which an efficient belief propagation algorithm can be run, the time complexity is reduced considerably. By considering the approximation $P(\hat{Y}|\hat{X})$ instead in order to cache approximate values for each conditional probability below the time complexity can be reduced to $O(n \log n)$. The assumption that makes the mutual information calculation tractable for simple queries over large knowledge bases with complex structure is explicitly

$$\forall x \in \hat{X}, y \in \hat{Y} \colon P(x|y) = P(x|\hat{Y}), \tag{5.8}$$

which is a reasonable assumption under the condition that the elements of the set of posets, $\hat{Y}$, are highly dependent variables, which will often be the case when comparing one set of dependent poset variables, $\hat{Y}$, against a second test set of poset variables, $\hat{X}$, in order to calculate the mutual information between these internally dependent clusters. Also, addressing this assumption becomes a moot point if the set, $\hat{Y}$, is a singleton set. Bayes' rule states

$$P(x, y) = P(x|y)P(y). \tag{5.9}$$

Substituting into Equation 5.7 and making the simplifying assumption that allows us to use a single application of the belief propagation algorithm, Equation 5.8, gives

$$I(\hat{X}, \hat{Y}) = \sum_{y \in \hat{Y}} \sum_{x \in \hat{X}} P(x|\hat{Y})P(y) \log_2 \frac{P(x|\hat{Y})P(y)}{P(x)P(y)} \tag{5.10}$$

$$= \sum_{y \in \hat{Y}} \sum_{x \in \hat{X}} P(x|\hat{Y})P(y) \log_2 \frac{P(x|\hat{Y})}{P(x)} \tag{5.11}$$

$$= \sum_{y \in \hat{Y}} P(y) \sum_{x \in \hat{X}} P(x|\hat{Y}) \log_2 \frac{P(x|\hat{Y})}{P(x)}. \tag{5.12}$$

We notice that the factor $log_2 \frac{P(x|\hat{Y})}{P(x)}$ tends to zero as x is independent of $\hat{Y}$, so if we consider the Markov blanket, $M_{y|\hat{X}}$, for each $y \in \hat{Y}$ with respect to $\hat{X}$ we will avoid these summations of zero. This gives

$$I(\hat{X},\hat{Y}) = \sum_{y\in\hat{Y}} P(y) \sum_{x\in M_{y|\hat{X}}} P(x|\hat{Y}) \log_2 \frac{P(x|\hat{Y})}{P(x)}. \tag{5.13}$$

Also, if we assume for the calculation of mutual information that the generative functions within the perception lattice are fully representative and exact matches, then we will have all children, C_x, of nodes, x, within the lattice to be a pure implication relationship, $\forall y \in C_x : y \rightarrow x$. If we consider specifically the situations where $y \in C_x$, Equation 5.13 expands to

$$I(\hat{X},\hat{Y}) = \sum_{y\in\hat{Y}} P(y) \left[\sum_{x\in M_{y|\hat{X}}\cap C_x} -\log_2 P(x) + \sum_{x\in M_{y|\hat{X}}\cap \overline{C_x}} P(x|\hat{Y}) \log_2 \frac{P(x|\hat{Y})}{P(x)} \right]. \tag{5.14}$$

We will refer to the second summation as the internal complexity, $\mathcal{C}_I(y)$, of node y.

$$\mathcal{C}_I(y) = \sum_{x\in M_{y|\hat{X}}\cap C_x} \log_2 P(x) \qquad \textbf{internal complexity} \tag{5.15}$$

Because internal complexity can be cached for each node, this reduces the limit of the amortized calculation of mutual information to the following equation:

$$I(\hat{X},\hat{Y}) = \sum_{y\in\hat{Y}} P(y) \left[-\mathcal{C}_I(y) + \sum_{x\in M_{y|\hat{X}}\cap \overline{C_x}} P(x|\hat{Y}) \log_2 \frac{P(x|\hat{Y})}{P(x)} \right] \tag{5.16}$$

$$= -\sum_{y\in\hat{Y}} P(y)\mathcal{C}_I(y) + \sum_{y\in\hat{Y}} \sum_{x\in M_{y|\hat{X}}\cap \overline{C_x}} P(x|\hat{Y}) \log_2 \frac{P(x|\hat{Y})}{P(x)}. \tag{5.17}$$

For reference, we can refer to $P(y)C_I(y)$ as the internal information, $I_I(y)$, of a node, y:

$$I_I(y) = P(y)C_I(y) \qquad \textbf{internal information} \tag{5.18}$$

It makes intuitive sense that the internal information within a node would be a negative quantity in the calculation of mutual information between that node and other nodes with different structures of functional generation. Therefor, to calculate the mutual information between two sets of posets, X and Y, the following optimized algorithm may be used:

$$I(\hat{X},\hat{Y}) = -\sum_{y\in\hat{Y}} I_I(y) + \sum_{y\in\hat{Y}}\sum_{x\in M_{y|\hat{X}}\cap\overline{C_x}} P(x|\hat{Y})\log_2\frac{P(x|\hat{Y})}{P(x)}. \tag{5.19}$$

Because we have restricted the poset nodes in the perception lattice that we consider due to the Markov blanket restriction, we now define the running time complexity of the general optimized version of the mutual information calculation to be in terms of

$$|\hat{Y}|\sim n \text{ and} \tag{5.20}$$

$$|M_{y|\hat{X}}\cap\overline{C_x}|\sim \log n. \tag{5.21}$$

So, after an $O(n\log n)$ belief propagation algorithm has been run with respect to an internally dependent set of posets $\hat{Y}$, Equation 5.19 can be calculated for an arbitrary set of posets $\hat{X}$ within the perception lattice in $O(n\log n)$ time.

The assumption that the perception lattice is representative of the universal knowledge base is a very strong assumption, which can be weakened if we instead assume a finite horizon of unknown data, but we leave this calculation for future work. We expect Equation 5.19 to be a helpful algorithm for recognizing small subposets within large knowledge bases that are used as different arguments within the same sets of generative functions. In text processing, this may provide a method for finding strings of language that are used in synonymous generative constructions. Within streams of sensor data posets within a perception lattice that have mutual information may represent the sensation of events that may be superficially different, but may share the same contextual causal relationships with surrounding sensor events (e.g. a metal door slamming and a glass door quietly clicking shut may both be preceded by and followed by footsteps).

5.3 Reasoning critics in different mental realms propagate beliefs to debug constraints

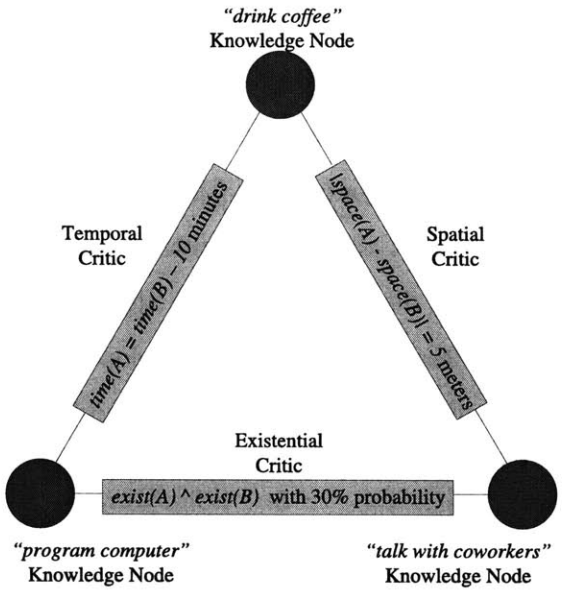


Figure 5-4: **Critics working together to solve complicated constraint problems** LifeNet uses a flat belief propagation method for solving these constraint problems. A hierarchical self-reflective method is mentioned in Minsky's (Minsky 2006) Model-6 architecture for reasoning.

The critical inference routine that is used in LifeNet is loopy belief propagation (Pearl 2000). This algorithm is used for a number of its properties: (1) scalable, (2) distributable, (3) and other equivalence-class algorithms exist. The belief propagation algorithm is scalable in the way that the algorithm functions at a fine granularity with respect to data it has to process. See Figure 5-4 for an example of how multiple critics in different reasoning domains can work together to debug inferred beliefs. Belief propagation runs in roughly linear, $O(n)$, time with the number of nodes, which is important when dealing with the millions of nodes in LifeNet; also, the memory required to implement the belief propagation algorithm is constant, $O(1)$, per node.

The locality of these finely granular data structures for each efficient calculation makes the belief propagation algorithm scalable and distributable in a heterogeneous network of many different processor types and capabilities, which applies to flat as well as heterogeneous sensor networks. Base stations may have a server class processor available with gigabytes of RAM, thus they are able to process millions of nodes, while other processors may be sleeping most of the time and are only able to process on the order of 10 or 20 nodes when they are awake, which would mainly be used for limiting radio communication between nodes. The third property of equivalence for the belief propagation algorithm refers to the fact that it belongs to a more general class of equivalent algorithms, namely Distributed Hill-Climbing algorithms. This class of algorithms includes the max-product algorithm, recurrent neural networks (or recursive sigmoidal regression networks), distributed genetic algorithms, and others. LifeNet in its present form has been designed partly as a development platform for this class of algorithm, all of which could span many processing nodes of different capabilities spanning decentralized servers to sensor network leaves in the same process.

5.3.1 Distributed-processing characteristics

Part of learning from everyday experience is our ability to categorize and segregate our knowledge into efficient domains of context-specific ways to think. We have briefly looked into ways to automatically segregate a large LifeNet into multiple domains of context-specific ways to think that can be processed independently, allowing for many independent reasoning algorithms to be run in separate processes that communicate a minimal amount of information. A hierarchical graph partitioning was calculated by iteratively applying spectral partitioning by Chaco (Hendrickson & Leland 1995).

We are experimenting with graph-partitioning algorithms on the entire LifeNet graph in order to separate very dense inference processing areas of the graph into separate processing modes. Using these techniques to divide the LifeNet processing and communication load across a heterogeneous sensor network has not been attempted, but the belief propagation algorithm has been shown implemented in a sensor network of this sort Ihler et al. (2004). Exact inference algorithms in sensor

networks such as the Junction-Tree algorithm (Paskin et al. 2005) will not scale to large belief networks such as LifeNet.

5.3.2 Logical truth inference

The LifeNet logical inference is based on a collection of truth relationships between statements about a typical person's life. The inference is used by providing LifeNet with evidence in the form of language statements associated with truth values that specify the probability of that statement. The logical model is specified as a Markov random field (Weiss & Freeman 1999), which performs roughly the same purpose as the first version of LifeNet (Singh & Williams 2003), except that the model in use now specifies explicit distances between time events rather than simply using a sliced model of time. The details of the temporal inference will be discussed with spatial inference after reasoning about logical truth.

5.3.3 Probabilistic existential critics

Each existential truth relationship between LifeNet phrases exists as a tabular probability distribution, forming a propositional Markov random field. These relationships relate the nodes within the Markov field. We will refer to these cliques as ψ_i for $i = \{1, 2, \ldots, C\}$ when C is the number of cliques within LifeNet. ψ_i is defined in terms of the probability distribution of the set of variables within that clique, ψ_{iX}. LifeNet factors, ψ_i, are tabular functions of the states of those factors, ψ_{iX}.

A	B	C	$\psi_i(A,B,C)$
F	**F**	**F**	1
F	**F**	**T**	0.75
F	**T**	**F**	0.025
F	**T**	**T**	0.05
T	**F**	**F**	0.125
T	**F**	**T**	0.05
T	**T**	**F**	0.05
T	**T**	**T**	0.1

Table 5.1: **Sample of LifeNet tabular potential factor of three nodes,** ψ_i Note that the symbols **T** and **F** are used as the *true* and *false* values for the variables A, B, and C. Also, $\sum_{\{A,B,C\}} \psi_i(A,B,C) \neq 1$.

A sample tabular potential function for a three-node potential function is shown in Table 5.1.

The potential functions, ψ, are indexed by the probabilities of their nodes, so although what is stored in each tabular potential are the probabilities of each node being 0 or 1 (*false* or *true*), these potential functions are actually linearly interpolated functions of the probabilities of these nodes, which can take on any values from 0 to 1. These potential functions are calculated as a sum weighted by the probabilities of all possible interpretations of a potential function:

Potential functions can be simplified relative to one variable, X_j, attaining a specific truth value, ν, which if the potential function is a probability table is effectively conditioning on that variable attaining that value. To calculate this potential conditioning, we sum over all possible combination of truth values within the potential function, ψ_i, that contain the condition $X_j = \nu$:

$$\psi_i(X = \nu) = \sum_{\lambda \in \Lambda_i^* \setminus (X \neq \nu)} \psi_i(\lambda), \tag{5.22}$$

where Λ_i^* is the set of all combinations of binary truth values for the set of variables, Λ_i, of the potential function ψ_i.

For each potential function $\psi_i(\lambda)$, the domain, λ is not a binary space but is

instead a bounded real space such that $\lambda \in [0-1]^{|\psi_i|}$, where $|\psi_i|$ is the dimensionality of the clique, ψ_i. This function is calculated by making a weighted sum of every tabular entry in the potential. The linear weighting is equal to the probability of that entry being true, given the domain, λ.

$$\lambda(\mu) = \prod_{X \in \Psi} P(X = \mu_X) \tag{5.23}$$

$$\psi(\lambda) = \sum_{\mu \in \Lambda^*} \lambda(\mu) \cdot \psi(\mu), \tag{5.24}$$

where λ_i is a set of probabilities for all nodes within the potential function. Potential functions need not sum to one and in general will not because they are not probabilities, but are factors that when multiplied together result in probability distributions.

LifeNet's belief propagation algorithm accepts evidence, E, for the probability of a subset of the LifeNet nodes. Given this evidence, belief propagation can efficiently estimate the probabilities of the remaining nodes. Let ξ_X^0 be the initial estimate of $P(X|E)$, which is the initial state of the iterating belief propagation algorithm. Within LifeNet, we assume $\xi_X^0 = 0.5$ for all nodes, X, such that $X \notin E$. Our purpose for using the belief propagation algorithm is that it is an efficient albeit unreliable method of iteratively calculating the following limit:

$$\lim_{k \to \infty} \xi_X^k = P(X|E). \tag{5.25}$$

Unfortunately, although the belief propagation algorithm is efficient ($O(n)$ time in the number of nodes), belief propagation is (1) not guaranteed to find the correct distribution when it converges, and (2) not guaranteed to converge. So, not only is this algorithm prone to getting stuck in local minima or garden-path interpretations of evidence but also could not converge to any solution ever. We have not yet implemented the generalized belief propagation algorithm (Yedidia et al. 2000); it has much better results for tightly interconnected constraints, such as turbo codes and probably even some of the more intricate logical reasoning capabilities of humans.

An efficient unreliable method is used in order to allow us to make the problem of probabilistically reasoning over millions of relationships tractable. For each node, X, we find a new estimate of $P(X|E)$, based on the current probability estimates, ξ_X^k, which gives us ξ_X^{k+1}. At each iteration, the probabilities for the nodes within the **Markov blanket** for each node is assumed to be equal to the most recent probability estimates for those nodes in the blanket.

The *Markov blanket*[1] for a node, X, in LifeNet, or any M.R.F., is equal to the set of cliques that contain that node. The subset of all cliques, ψ, that contain a node, X, is the Markov blanket, X_β, of that node:

$$X_\beta = \{\psi : X \in \psi \in \Psi\}. \tag{5.27}$$

The Markov blanket of X is the minimal set of nodes that when known, effectively make $P(X)$ independent from any other evidence within the network.

The belief propagation algorithm uses the potential functions by setting the domain of the potential functions at iteration, k, to be

$$\lambda^k(\mu) = P(\mu|E, \xi^k). \tag{5.28}$$

The iterative algorithm for updating the probability estimates, ξ_i, for each of the nodes is

$$\xi_X^{k+1} = \prod_{\psi \in X_\beta} \psi(\lambda^k), \text{ for all } X. \tag{5.29}$$

We allow the belief propagation algorithm to iterate for a limited number of time-steps (currently 10) in order to get an estimate of the limit.

[1] *Markov blanket*: We refer to the Markov blanket as the set of cliques that a given node belongs to because this is easier within the M.R.F. framework, but in general the Markov blanket is referred to as the set of nodes that are contained within these cliques. Or more generally, the set of nodes when whose probabilities are known fully specify the probability of a given node such that

$$P(X|X_\beta) = P(X|X_\beta, E) \tag{5.26}$$

for any evidence, E.

5.3.4 Probabilistic rendering and inference language

In order to easily render probabilistic distributions in N-dimensional space, a simple N-dimensional probabilistic rendering language was written. This language was used to render the gold-standard knowledge base in the 2-dimensional floorspace of the Media Lab, but this same language easily renders probabilistic distributions in arbitrary named-dimensions—for example, "happiness" or "risky" could be dimensions that human-language concepts could be probabilistically distributed by this rendering language. The language is very simple and only consists of three drawing primitives:

Command	*Description*
box	Boxes are volumetric distributions that have a specified size for each dimension and 90-degree right angled corners—rectangular prisms with constant probability density.
ball	Balls are volumetric distributions that have a specified radius for each dimension—ellipsoids with constant probability density.
path	Paths are volumetric distributions that have a specified radius for each dimension—cylindrical lines extending between two points with constant probability density.

Table 5.2 demonstrates the use of a programming language that was developed to represent commonsense objects of different shapes in N-dimensional space.

```
(or (at (box  1 x 43.75 -0.75      y 18.75  0.75)      "coke vending machine")
    (at (ball 1 x 34.75  0.25      y 19.75  0.25)      "trashcan")
    (at (path 1 x  6.75  6.75 0.03 y  9.25 18    0.03) "glass wall"))
```

Table 5.2: **Example of the LifeNet programming language** This representation for commonsense objects in N-dimensional space uses three object shapes to create probability distributions within the two-dimensional space, including the dimensions "x" and "y."

Arbitrary strings can be used to name arbitrary numbers of dimensions in this language, so that objects can be placed in 4-vector space-time or any other space of conceptual (or phenomenological) dimensions, such as "utility", "risk", "crime", or "happiness" easily. The specifications for this language are included in Section A.

Using these rendering commands as an abstract definition language for shapes of probability distributions in N-dimensions is a very useful way to not only create probability distributions as it is being used for in this thesis but may also be useful as a representation for learning and recognizing concise descriptions from arbitrary probability distributions, but this is left for future research.

The three shapes that can be rendered in this simple representation are interpreted by LifeNet to create mixtures of Gaussians that are limited in resolution, so that any single command only allocates a predetermined number of Gaussians for that distribution. This number is set to a relatively low number for current LifeNet computations—typical 16 or 32 Gaussians per phenomenon. Figure 5-5 shows how changing this resolution affects the resulting probabilistic distributions in two-dimensions (graphed in three-dimensions with height representing instantaneous probability density).

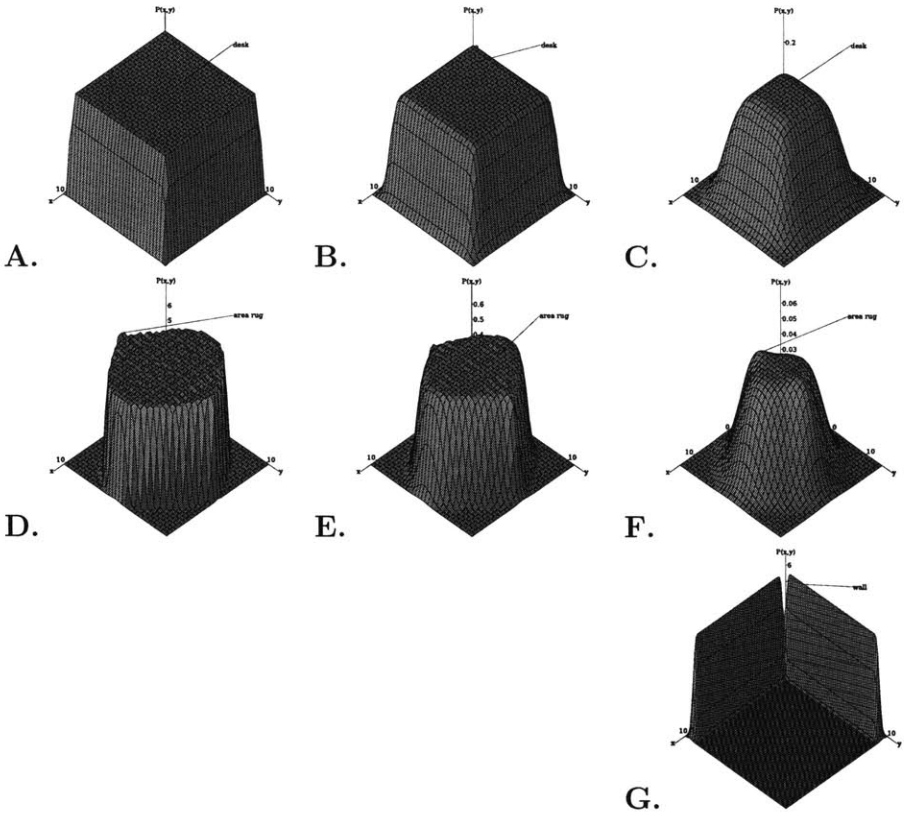


Figure 5-5: **Example of compositional LifeNet probability distributions** The LifeNet language is used to compose these probability distributions into existential relationships between phenomena. These two-dimensional boxes (A-C), balls (D-F), and paths (G) are graphed in three dimensions with height representing instantaneous probability density. These figures are each represented by either 3200 (A, D), 320 (B, E), or 32 (C, F, G) Gaussians, which gives them high to low resolution forms. The resolution that LifeNet uses by default for large-scale N-dimensional inference problems is 32 Gaussians.

5.3.5 Inferring past, present, and future.

The temporal (see Figure 5-6) and spatial (see Figure 5-7) reasoning within LifeNet are now handled as part of a mixtures of Gaussians belief propagation algorithm that uses mixtures of Gaussians to represent distributions in real-number spaces. This will be the technology that allows us to incorporate the 9-dimensional sensor space of the Plug network with LifeNet's commonsense spatial and temporal inference.

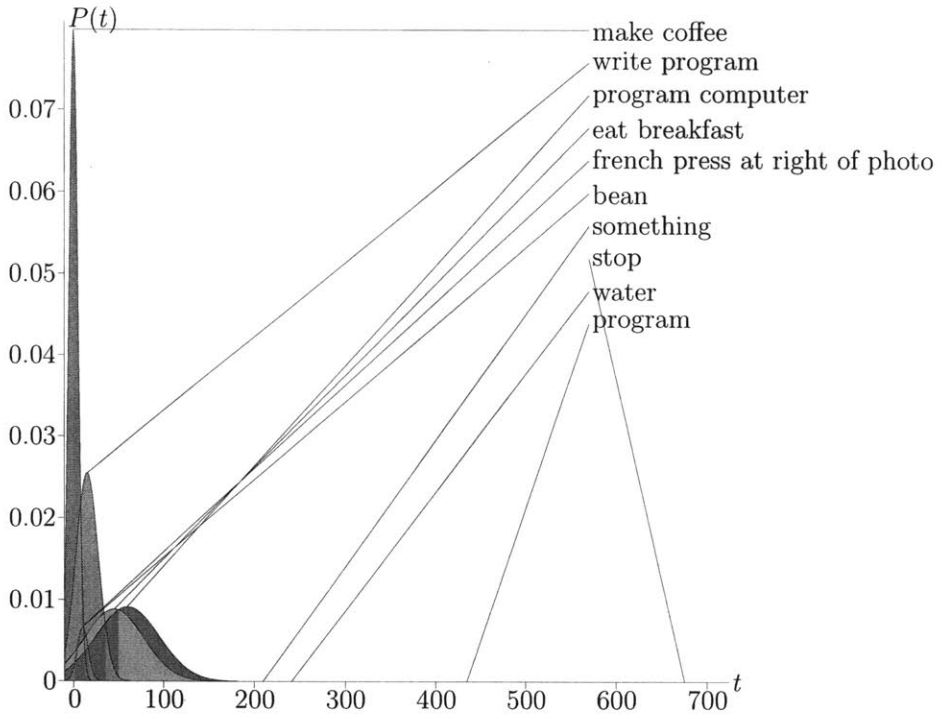


Figure 5-6: **Temporal probability distribution of LifeNet phenomena** The "make coffee" node is set to have a probability distribution in time that is a simple Gaussian with $\mu = 0$ minutes and $\sigma = 5$ minutes. Time, t, is in minutes. The subsequent distributions were generated by LifeNet's commonsense knowledge and assumptions (to be corrected by sensor data) of temporal distance.

5.3.6 Inferring nearby phenomena

The spatial representation uses a three-dimensional Gaussian subspace to represent spatial relationships between propositions that can be true or false relative to a position in latitude, longitude, and altitude dimensions measured in meters. For example, if the system were given a list of objects that are known to exist together, LifeNet can provide a probability distribution over all possible arrangements of those objects.

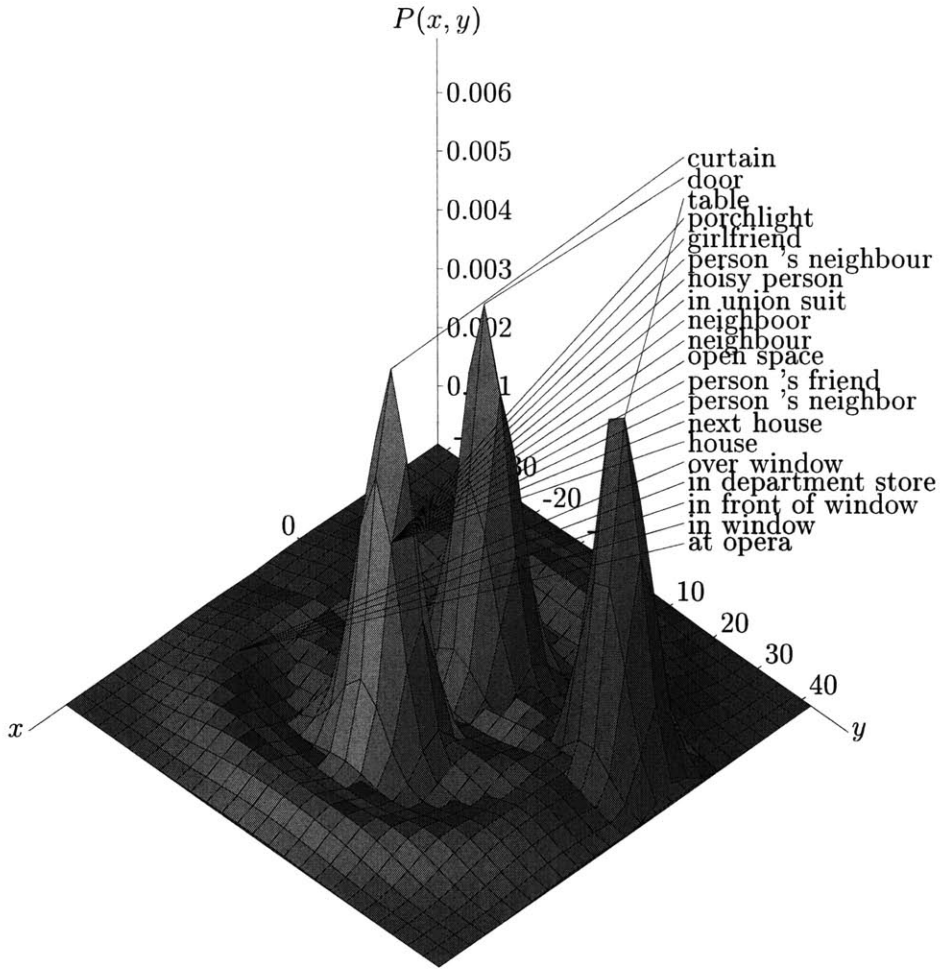


Figure 5-7: **Spatial probability distribution of LifeNet phenomena** The "window", "curtain", and "door" nodes are set to have probability distributions that are simple Gaussians in two-dimensional floor space measured in meters. The inferred probability distributions for the other concept nodes are shown as mixtures of Gaussians that are in this case circular, but can be in general an approximation of any distribution.

The LifeNet graph is a heterogeneous network of different types of phenomenon data nodes and modal phenomena pattern edges, such that it is a more general type of artificial intelligence processing architecture. LifeNet is a graph with nodes and cliques (N-member edges). LifeNet edges can be considered as the following different types:

- concept phrases in Unicode text

- symbolic sensor phenomena
- sensor modal stream patterns
- commonsense modal story patterns
- analogous modal patterns
- metaphorical modal patterns

The nodes of LifeNet are things that can be reasoned about in space and time—they can each have a position—and they can also be reasoned about existentially—they can each have a probability value of existing. See Figure 5-8 for a visualization of the types of knowledge and critics within LifeNet.

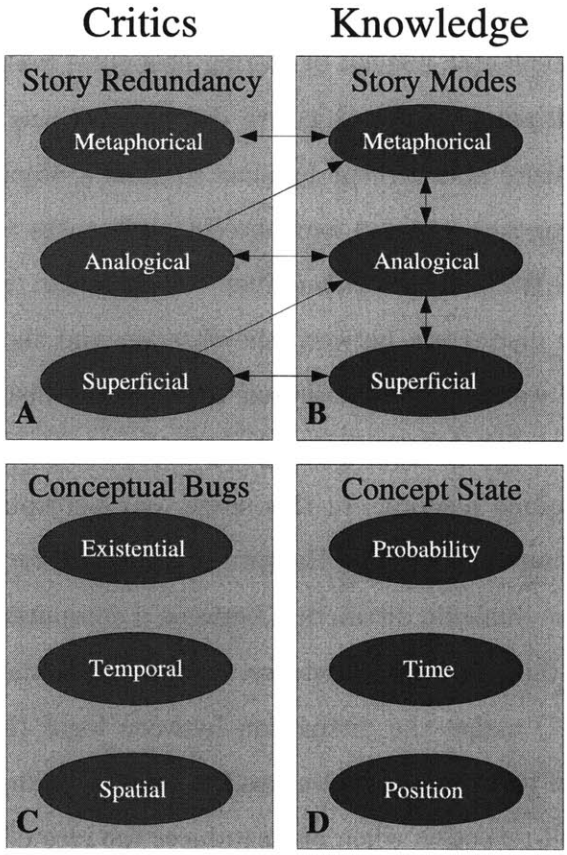


Figure 5-8: **The LifeNet graph knowledge nodes and critics** Critics operate over knowledge nodes making sure that constraints are maintained between knowledge nodes. *(A)* Story Redundancy Critics operate over all knowledge nodes that are sequential streams, which includes superficial knowledge in the form of commonsense text stories or symbolic sensor streams. Superficial streams are scanned for symmetries (redundancies) and these are abstracted to analogical modal representations; this process of abstraction repeats in order to gain metaphorical abstraction levels. *(B)* Story Modes are the knowledge layers that the Story Redundancy Critics operate over. A repetition recognized in the Superficial Story Mode layer is stored in the Analogical Story Mode layer and the repeating superficial stream is reduced to a single copy in the Superficial Story Mode layer. Repetitions in the Analogical Story Mode layer are similarly abstracted to the Metaphorical Story Mode layer. *(C,D)* Phenomenological Bugs exist in the constraints between the Phenomenon States, which are the states that are associated with the phenomenon knowledge nodes, such as the position or probability of a phenomenon.

5.4 Toward self-reflection by blurring the objective duality between algorithms and data

We use the term "perception lattice" very similarly to the structure lattice in Jepson & Richards (1994) except that instead of storing elemental preference relations in a separate lattice, which relies on the objective distinction being placed between the two types of data, feature and percept (similar to Kant's objective duality of neumenon and phenomenon respectively), our percept preferences are inherently part of the same structure as the percepts themselves. To be clear, we have not avoided the objective dualistic distinction between the observer and the world but by using the perception lattice we have merely chosen to place the distinction between the algebra of computation and the perceptual data that is to be explained by itself as structured by the assumed algebra. In this sense, our perception lattice explicitly includes a model of computation. The Harvard architecture for memory access and storage makes the same dualistic distinction between a computational algorithm and the data, over which this algorithm operates. Similarly, in the philosophical literature, Heidegger (1962) makes the distinction between logos (letting something be seen) and phenomenon (that which shows itself in itself). Heidegger also introduces the powerful idea of self-reflection when he introduces the idea of considering logos as phenomenon, which in the present analogy maps to considering algorithms as data, bypassing the assumptions of the traditional objectively dualistic Harvard architecture. Some programming languages, such as Lisp and Python, allow algorithms that dynamically process algorithms as data, which blur the objective duality between algorithms and data, making a rich and under-explored area of self-reflective computational research. We feel that the perception lattice is more appropriate to future research in self-reflective computational perception than the structure lattice, which places the objective duality between two different types of data. For two proof-of-concept examples of social robots with multiple layers of self-reflective debugging algorithms that are processed as data for perception and action please see Singh (2005).

5.5 Toward abstraction using explanation-based similarity

Using generative functions as modes of perception that provide explanations for what it means for data to exist provides a means for considering how similar two pieces of data are based on whether or not their "means of existence" are similar or, in other words, how the processes that generated the data are similar. Explanation representations were used in a planning environment (Bergmann et al. 1994) in order to judge similarity in a multiple-layered graph structure that makes the distinction between rule nodes and fact nodes and takes advantage of fact abstraction and rule abstraction to map between different layers. The data nodes within the perception lattice could be considered fact nodes, while the generative functions that form the edges of the perception lattice could be considered rule nodes. Considering this mapping of terminology, perhaps a similar method of fact abstraction could be employed where multiple data nodes could be mapped to an abstract data node, and similarly, a sublattice of generative functions and data nodes can be compiled to abstract generative functions and data nodes. See Figure 5-9 for a visualization of the LifeNet process of abstraction by recognizing similar generative functional explanations for data.

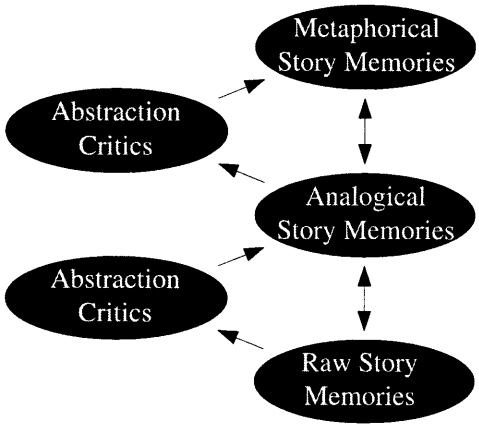


Figure 5-9: **LifeNet analogies by functional abstraction** Perception lattice abstractions result from using a greedy compression search to find functional generative structures in partially ordered text stories and partially ordered sensor streams. The generative function structures that take similar arguments could be considered as analogies because they provide a measure of similarity between data, and the arguments that are analogous share a large amount of mutual information. If the process of abstraction was generalized to work over the generative function structures and arguments themselves then a second level of metaphorical abstraction and similarity could be developed for more abstract cross-modal mappings.

Chapter 6

Performance evaluation

We evaluate the LifeNet critical reasoning algorithm on two very different knowledge bases:

1. **Commonsense objects in a research office environment**

2. **Ubiquitous Plug sensor-network platform audio streams**

We demonstrate that using commonsense knowledge bases and inference tools, such as LifeNet, improves traditional bottom-up machine learning performance at understanding the human patterns inherent in these knowledge base.

6.1 Commonsense English language spatial position learning and inference

Our first task in evaluating LifeNet is to test how well the commonsense English language phrases that were gathered from volunteers on the web reflect real-world spatial relationships between commonsense objects.

6.1.1 Gold-standard knowledge base for spatial inference evaluation

We use a gold-standard testing knowledge base that represents an example of a typical office environment that contains desks, chairs, computers, and other common objects. This knowledge base was collected by hand by a single researcher by printing a 4-foot-by-4-foot poster of the architectural layout of the Third Floor of the Media Lab office and research space and subsequently drawing and labeling all of the common objects in a few select public spaces of this environment. This knowledge base consists of 14 public architectural spaces that contain a minimum of 15 and a maximum of 94 common objects. These spaces included a kitchen, a bathroom, private office spaces, and public research and presentation spaces. Figure 6-1.A shows a visualization of the boundaries of these lab spaces.

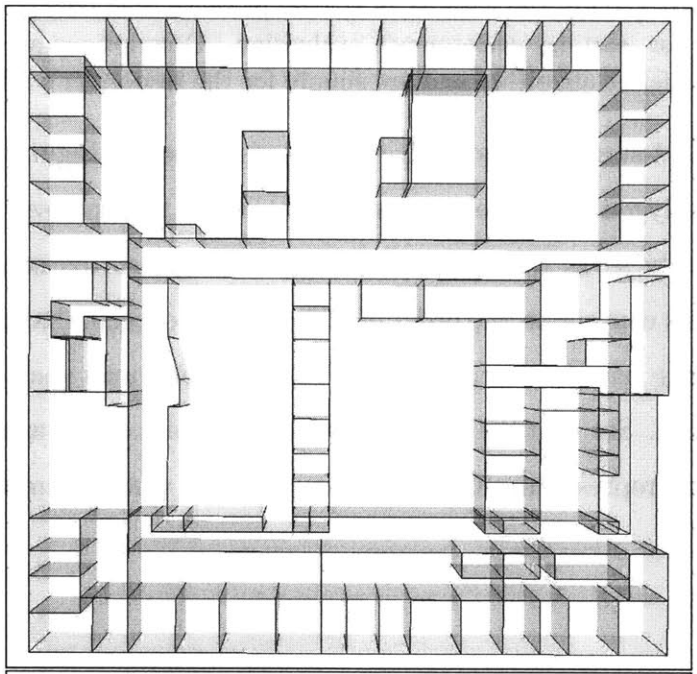
Description	**Code**	**Object count,** n	**Binary relations,** $\binom{n}{2}$
vending machines	E15-300CB	11	110
kitchen	E15-342	14	182
kitchen hallway	E15-300CC	16	240
men's bathroom	E15-399	21	420
private office space	E15-311	15	210
private office space	E15-319	24	552
electronics lab space	E15-344	36	1260
public research space	E15-301	94	8742
public research space	E15-305	20	380
public research space	E15-310	16	240
public research space	E15-318	31	930
public research space	E15-368	86	7310
public research space	E15-383	59	3422
public research space	E15-384	63	3906
sum		506	27904
mean		36	1993

Table 6.1: **Commonsense object architectural space data summary** These data were gathered from the Third Floor of the Media Lab at MIT. Commonsense objects were enumerated with their simple shapes, sizes, orientations, and positions. This knowledge base is an example of an average arrangement of commonsense objects in a research lab environment. Positions of some objects in this knowledge base, such as oscilloscopes and signal generators are atypical of common office environments, but most objects such as

tables, chairs, bookshelves, filing cabinets, staplers, etc. are assumed to be relatively representative of common office environment configurations of these more universally common objects. Binary relations, $\binom{n}{2}$, between objects refers to the number of pairs of objects in each space and subsequently the number of spatial pairwise relationships between objects that LifeNet uses for learning performance evaluation. Descriptions are not used in the evaluation of the LifeNet algorithm and are simply for the researcher's reference.

The spatial gold-standard knowledge base is visualized in Figure 6-1.B. The map is composed of polygonal architectural spaces. LifeNet reasons over each polygonal space independently so as to confine reasoning to single rooms at any given time. These single rooms contain on the order of 100 common objects that comprise a gold-standard knowledge base for testing how well LifeNet performs spatial inference in a real-world situation. See Figure 6-2 for a visualization of a single architectural space with colored areas representing Gaussian distributions where commonsense objects exist.

A.



B.

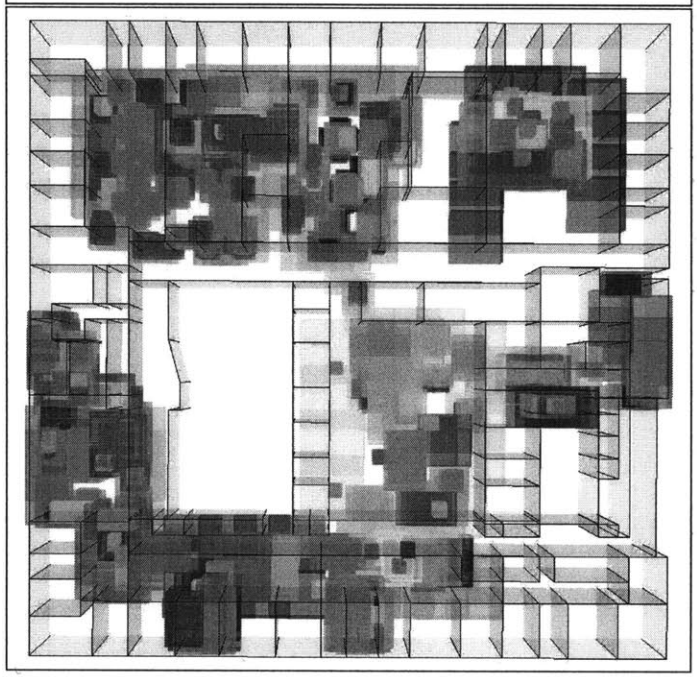
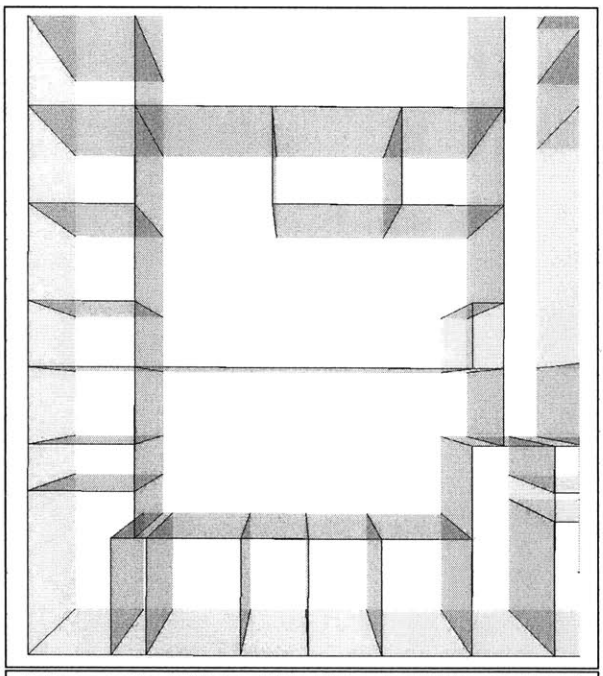


Figure 6-1: **Visualization of the commonsense objects in the Third Floor of the Media Lab** (A) The architectural spaces of the Third Floor of the MIT Media Lab consists of both public and private research and office environments. This knowledge base is composed of the positions, shapes, sizes, and rough orientations of 506 commonsense office environment objects (B) that were collected by hand. These 506 objects exist in 14 architectural spaces, which (considered independently) contain 27904 pairwise spatial

relationships for evaluating the LifeNet inference algorithm on real-world data.

A.



B.

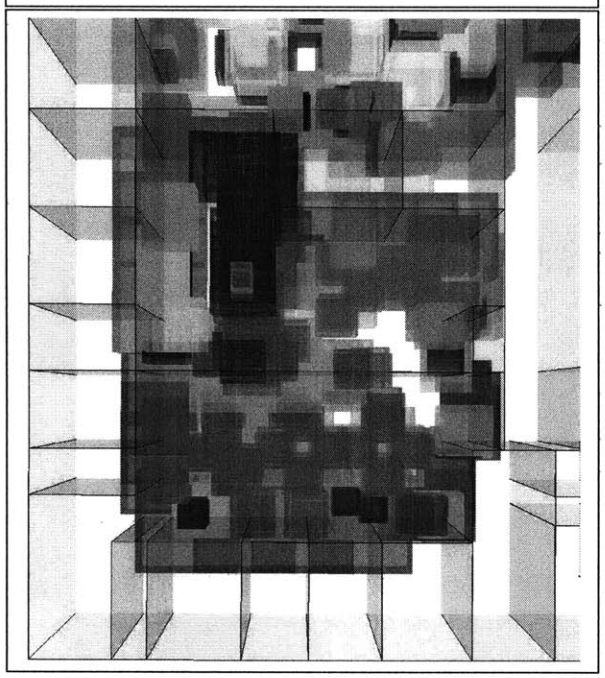


Figure 6-2: **Visualization of the commonsense objects within a single public research area in the Third Floor of the Media Lab** The architectural layout of a single public architectural space of the (A) Northwest corner of the Third Floor of the MIT Media Lab is shown (B) filled with different common physical objects. Figure 6-1 shows the complete overview of the lab space, while this figure shows only one space within the lab.

6.1.2 Spatial evaluation tests and results

The method of evaluation for the spatial gold-standard knowledge base was to divide the knowledge into separate sets in order to train the algorithm on some of the data, while leaving some of the data for testing the performance of the inference algorithm. The data was divided into 10 sets. One of these sets was used for testing, while the remaining nine sets were used for training the LifeNet inference algorithm. The LifeNet algorithm was evaluated in its ability to infer the existence of the commonsense object phenomena within architectural spaces, given 90% of the other objects in the space.

LifeNet's spatial inference accepts a set of objects at locations (mixture of Gaussians probability distribution of locations) as evidence. LifeNet's belief propagation algorithm was limited to return 1000 phenomena partitionings of the 350-thousand English language phenomena. LifeNet is able to expand the context of a given architectural space within the training data set. Of the inferences made by LifeNet, 13% of these accounted for 85% of the gold-standard testing sets, while 87% were false-positive inferences. If we limit the perception lattice to be only trained on the language in the OMICS (Kochenderfer & Gupta 2004) knowledge base, then the performance drops to 17% of the returned inferences accounting for only 55% of the gold-standard testing sets with 83% of inferences being false-positives.

6.2 The Plug sensor-network

The sensor network that we are using for both learning commonsense and for recognizing and predicting human behavior is the Plug sensor network (Lifton et al. 2005). This network is a heterogeneous network consisting of base-station power-strips that contain 9 sensor modalities: sound, vibration, brightness, current (through 4 separate plugs), and wall voltage. the Plug sensor network is augmented by small low-power nodes with accelerometers that can be used to roughly position and track individual common objects around our lab space, which has the base-station plugs scattered throughout. Using this sensor network to monitor how individuals interact with their

physical environment by moving specific objects or simply by their sensor impression on the environment provides a stream of data that can be correlated with simple conceptual human-language descriptions of the same events so as to define a supervised probabilistic learning problem. the Plug sensor network is a useful device that theoretically could be readily deployed in both home and office settings.

We have supervised the data collection for the activities in Table 6.2. The data collection with the Plug sensor-network took place in the four-hour period from 2:00 A.M. to 6:00 A.M. on Wednesday, July 12, 2006. This was a period of time when there were very few people working in the Media Lab, providing a relatively quiet environment for collecting the audio power spectra for our proof-of-concept evaluation. There was a group of people watching television in the background of some of the data collection, and approximately 4–5 other researchers in the lab at the time. The Plug sensor network was distributed across the length of the lab from the kitchen area down a long hallway and into the men's bathroom, passing through three sets of doors, one glass and the other two metal.

ID	Description	Plug	Duration (s)
1	*drinking fountain used*	1E	43
2	*drinking fountain cooler turned on*	1E	110
3	*flushing urinal*	10	155
4	*walking in bathroom*	10	140
5	*closing bathroom stall door*	10	165
6	*flushing bathroom toilet*	10	140
7	*washing hands in bathroom*	10	90
8	*pulling out chair, sitting, pushing chair in (by drinking fountain)*	1E	165
9	*opening and closing latched metal door*	15	60
10	*laptop music*	1D	155
11	*opening and closing microwave door (with TV in background)*	01	17
12	*conversation in next room (with TV in background)*	01	45
13	*coffee machine beep (with TV in background)*	01	60
14	*washing dishes (with TV in background)*	01	125
15	*opening and closing toaster oven (with TV in background)*	01	110
16	*using urinal*	10	23
17	*walking by drinking fountain*	1E	85
18	*try to open locked glass door*	1E	20
19	*walking by printer in hall*	17	150
20	*walking by corner of TMG area*	1D	120
21	*pressing microwave buttons (with TV in background)*	01	60
22	*microwave on (with TV in background)*	01	180
23	*typing on a computer keyboard*	17	110
24	*opening and closing microwave door*	01	100
25	*pressing buttons on coffee machine*	01	80
26	*using vending machine*	15	30
27	*washing dishes*	01	90
28	*opening and closing toaster oven door*	01	35
29	*walking by outside of TMG door near vending machine*	15	120
30	*microwave on*	01	180

Table 6.2: **Plug sensor-network activities** The Plug sensor-network was used to collect data while these activities were being performed within audible range of one of the sensor-network nodes. The **ID** column lists the identification numbers of the activities, while the **Plug** column lists the identification code of the nearest plug that is also within audible range of the activity. Note the free use of human-language in the descriptions.

In order to evaluate whether or not the commonsense data within LifeNet helps in order to predict these activities, we have told LifeNet the commonsense contextual information about the six Plug sensor nodes in Table 6.3.

Plug	Commonsense Context
01	*kitchen*
10	*men's bathroom*
15	*hallway*
17	*hallway*
1D	*hallway*
1E	*hallway*

Table 6.3: **Plug commonsense context** This commonsense context is used to attempt to show that when we use a simple piece of commonsense context to describe the surroundings of a sensor node the inferences about that node become more accurate.

The commonsense context for each plug in the sensor-network in Table 6.3 is purposefully very simple in order to demonstrate that only a small amount of commonsense context can refine an inference algorithm's prior probability distribution so that the posterior is limited to a much smaller range of inferences. This sort of top-down effect on the posterior distribution allows the massive state spaces of LifeNet's 350,000 language phenomena (10^{90309} existential states) to be quickly narrowed down to a few thousand phenomena (10^{301} existential states). For example, one of the commonsense stories that LifeNet knows about kitchens is as follows: "go to the kitchen", "find the coffeemaker", "put ground coffee", "fill water in coffeemaker", "turn coffeemaker on", "wait for coffee to stop dripping", "pour coffee from pot into cup." In this story, we see that LifeNet has learned existential and temporal relationships between "kitchen" phenomena and "coffee" phenomena.

Without the commonsense context for each sensor-network plug, LifeNet infers the existence of 22% of the correct activity language phenomena in 17% of the inferences with 83% being false-positives. With the commonsense context for each sensor-network plug, LifeNet infers the existence of 13% of the correct activity language phenomena in 20% of the inferences with 80% being false-positives.

6.3 Performance discussion

Although the large semantic networks that we have used to provide context to the LifeNet inference algorithm did not increase the accuracy of the inferences on our relatively small evaluation knowledge base, we do see promise in the semantic depth of the inferences that were made.

6.3.1 Evaluation of human-scale state spaces

For example, in the context of the phrase "kitchen" LifeNet correctly associates the commonsense knowledge that "dishes" are found in a "kitchen" with the activity of "washing dishes"; however, LifeNet also associates "dishes in one stack", "dishes on the table", "meal in dishes", and "silverware" with this same context. The introduction of these contextual phrases into the LifeNet inference algorithm reduced the accuracy of the LifeNet algorithm in predicting the specific activity of "washing dishes," but if we look through the high number of false-positive inferences that the LifeNet inference algorithm made we notice that some of these inferences did correctly represent "silverware," "dishes in one stack," and "dishes on table," which did exist in the kitchen that we supervised but were not labeled within the small knowledge base. We hypothesize that these contextual inferences were evaluated as false-positive inferences because of the small size of our evaluation knowledge base. The collection of a data set that is large enough to contain all of the commonsense descriptions of an activity recognition environment, such as the one we used in this evaluation, would likely prove to be a task at least as difficult as collecting the semantic commonsense knowledge base itself. We propose that the formal machine-learning evaluation that we have performed in this thesis is not appropriate for such a large state space, where a human psychological evaluation may be more appropriate.

6.3.2 Context-expansion

These evaluations show the ability of LifeNet to perform context-expansion by limiting a top-down activity recognition to detect objects in a room given the contextual

knowledge of other objects within the room. However, because of the high false-positive percentage of returned inferences, the LifeNet algorithm is not good for accurately predicting exactly what objects are in a room when given only the other objects in the same room. The limitation of the number of nodes that could be included in the belief propagation algorithm (1000 phenomena) was due to limitations in the processing requirements necessary. This limitation could be overcome in the future by optimizing the matrix operations necessary for processing the Gaussian mixtures. Also, optimizations using hash spaces for calculating Gaussian intersections in mixtures would reduce many of the multiplication bottlenecks in high resolution (100 Gaussians) mixtures of Gaussians.

Chapter 7

Future directions

Sensor data cannot be simply understood by machine learning algorithms that do not have a human-level language description of what is going on. In order to predict and understand human behavior in a sensor-rich environment, sensor-networks will need to incorporate models like LifeNet that contain first-person commonsense conceptual models of human behavior. We also hypothesize that once human behavior can be explained by these top-down commonsense constraints, more specific commonsense patterns can be bootstrapped from this initial mapping of sensor data to human behaviors, leading to typical patterns of human behaviors, which would not be possible without the initial top-down commonsense language constraints.

7.1 Learning metrical commonsense

We expect that gathering the spatial and temporal arrangements of commonsense phenomena from web volunteers, which is a slightly different task from the original OpenMind Commonsense knowledge acquisition project that mainly focused on gathering commonsense semantic and language patterns, will be a fruitful path of future research. The problem of gathering supervised descriptions of human activities still remains the highest hurdle for automatic activity recognition algorithms such as LifeNet. But we have shown that the processes of gathering commonsense from the web community at large and the supervision of specific sensor data tasks can be used

to bootstrap one another, such that commonsense language phenomena can be used to improve human activity recognition, while human activity recognition can provide new commonsense stories of human activities.

7.2 Learning commonsense privacy

The cell phone diary application touches directly on the complex issue of privacy and learning trends from large numbers of user records. The issues of privacy and security and how to share these personalized common senses between users within this system are key issues that have not received enough focus, but here is a simple breakdown of one possible axis with which to consider this issue of privacy and security:

Privacy

Complete	No information is shared between individuals.
Commonsense	Common patterns are shared, but personal information is not shared.
None	All information is shared and used across diary accounts, so any activity pattern for one person will be used to find patterns in all other users.

The optimal privacy strategy will lie somewhere between the complete isolationism of complete privacy and the complete insecurity of having all information shared between people. Perhaps there is a common sense threshold for information that specifies that if a particular pattern of activity and description are associated among a large enough percentage of the population then that piece of information is considered to be commonsense within that population of people.

7.3 Goal-oriented people and objects

LifeNet learns and uses generative modes to explain the sensor data that it is trying to relate to humans that are trying to reflect on themselves and their environment in order to plan their future behavior. The future of LifeNet is the ability to not only recognize and learn single activities, but to keep track of more complex models of humans that involve keeping track of different people by trying to keep track of their current goals. For example, if we know that someone is "trying to buy something

to eat" maybe he would "drive a car" or "take a walk", while someone who has the different goal of "getting ready for work" would "take a shower" or "brush teeth."

Possible avenues to realize the probabilistic models necessary to infer these more complex goal states may include object-oriented representations (Daphne & Pfeffer 1997) (Minsky 1974) to try to answer more difficult questions, such as the following:

- "Which one of these types of objects generated these data?"
- "Who generated these data?"

Also, social and intentional object-oriented models (El Kaliouby & Robinson 2005) (Baron-Cohen 1995) of generative processes might provide hints toward how to answer the following questions:

- "What goals was this person pursuing while generating these data?"
- "Why would a person generate these data?"
- "Is this person the type of person that would generate these data?"
- "What was this person thinking about that person when these data were generated?"

Social and intentional models of goals and other aspects of human mental thought processes will be necessary for the artificially intelligent robots, computers, and sensor-networks of the future.

We have shown that the commonsense semantic language data, such as the Open-Mind Commonsense knowledge base, can be used to bootstrap quicker learning algorithms for classifying and recognizing sensor phenomena and in turn common human activities. This ability for LifeNet to symbolically and numerically reason over multiple different mental realms of commonsense phenomena will allow people to interact with sensors more easily in their own language or in a novel sensor environment. Also, we hypothesize that extending this architecture into a system that learns object-oriented generative explanations for sensor data may lead to systems that can

learn to objectify sensor data and use these objectifications to project more intelligent perceptions onto the world.

Appendix A

LifeNet Programming Language (CriticsLang) Commands

A.1 System Commands

- quit()
- load(F)

  $F \equiv$ Filename of file containing of CriticsLang LifeNet programming language commands
- print(E)

  $E \equiv$ LifeNet phenomenological evidence that should be printed to standard output
- graph(E)

  $E \equiv$ LifeNet phenomenological evidence that should be graphed on x and y dimension axes (vertical axis representing probability of phenomena).

A.2 Logical Operators

- not(E)

$E \equiv$ LifeNet phenomenological evidence that is inverted by the *not* operator. The *not* operator currently only works for Boolean existential truth evidence and does not currently work for mixtures of Gaussian distributions.

- and$(E_0, E_1, ..., E_n)$

  $E_i \equiv$ LifeNet phenomenological evidences that are pointwise-multiplied together.

- or$(E_0, E_1, ..., E_n)$

  $E_i \equiv$ LifeNet phenomenological evidences that are pointwise-added together.

- xor$(E_0, E_1, ..., E_n)$

  $E_i \equiv$ LifeNet phenomenological evidences that undergo the operation $\bigvee_{i\in\{1,2,...,n\}} \left(E_i \wedge \bigwedge_{j\in\{1,2,...,n\}\setminus i} \overline{E_j}\right)$.

- implies(E_0, E_1)

  $E_i \equiv$ LifeNet phenomenological evidences that undergo the operation $\overline{(E_0 \wedge \overline{E_1})} = E_0 \rightarrow E_1$.

A.3 Probability Distribution Rendering Commands

- point$(d_0, v_0, d_1, v_1, ..., d_n, v_n)$

  Creates a point-like (Gaussian with minimal variance) probabilistic distribution with probability of 1.0

  $d_i \equiv$ A phenomenological mode specifying the symbolic reference for one dimension of the n-dimensional point

  $v_i \equiv$ A real-number specifying the position of the point along the dimension d_i

- ball$(p, d_0, c_0, r_0, d_1, c_1, r_1, ..., d_n, c_n, r_n)$

  Creates a mixture of Gaussians probabilistic distribution that approximate the shape of an n-dimensional ball

$p \equiv$ Total probability of the mixture

$d_i \equiv$ A phenomenological mode specifying the symbolic reference for one dimension of the n-dimensional ball

$c_i \equiv$ A real-number specifying the position of the center in the dimension d_i of the n-dimensional ball

$r_i \equiv$ A real-number specifying the standard deviation along the d_i of the n-dimensional ball

- box$(p, d_0, c_0, l_0, d_1, c_1, l_1, ..., d_n, c_n, l_n)$

  Creates a mixture of Gaussians probabilistic distribution that approximate the shape of an n-dimensional box

  $p \equiv$ Total probability of the mixture

  $d_i \equiv$ A phenomenological mode specifying the symbolic reference for one dimension of the N-dimensional box

  $c_i \equiv$ A real-number specifying the position of the corner in the dimension d_i of the N-dimensional box

  $l_i \equiv$ A real-number specifying the length (can be negative) along the d_i of the N-dimensional box

- path$(p, d_0, a_0, b_0, r_0, d_1, a_1, b_1, r_1, ..., d_n, a_n, b_n, r_n)$

  Creates a mixture of Gaussians probabilistic distribution that approximate the shape of an n-dimensional path

  $p \equiv$ Total probability of the mixture

  $d_i \equiv$ A phenomenological mode specifying the symbolic reference for one dimension of the N-dimensional path

  $a_i \equiv$ A real-number specifying the position of one end of the path in the dimension d_i

  $b_i \equiv$ A real-number specifying the position of the other end of the path in the dimension d_i

  $r_i \equiv$ A real-number specifying the standard deviation along the d_i of the N-dimensional path

A.4 Phenomenological Relationship Commands

- at(D, m)

  Creates LifeNet evidence that places a phenomenological mode, m, and a probability distribution, D. Within the LifeNet algorithm this datatype is referred to as a *reaction*, which is a reference to the *reactive* layer of the *Model-6* architecture mentioned in Minsky (2006). This LifeNet reaction forms the universal representation that all of the diverse reasoning algorithms within LifeNet share. As LifeNet grows beyond simply a *reactive* algorithm, other types of data will be necessary that will be used for *ways of thinking* beyond simply probabilistic inference.

  $D \equiv$ Probability distribution (either mixture of Gaussians or simply existential)

  $m \equiv$ A phenomenological mode specifying the symbolic reference for the phenomena that is distributed as D

- related(D, E_0, E_1)

  Creates a binary relationship within LifeNet that relates two pieces of evidence, E_0 and E_1, by the distribution D. LifeNet uses these relationships to infer posterior probability distributions from prior evidences.

  $D \equiv$ Probability distribution (either mixture of Gaussians or simply existential)

  $E_i \equiv$ LifeNet evidences that are related by the distribution D.

- north(d, E_0, E_1); south(d, E_0, E_1); east(d, E_0, E_1); west(d, E_0, E_1); above(d, E_0, E_1); below(d, E_0, E_1); before(d, E_0, E_1); after(d, E_0, E_1)

  All of these functions are similar in that they are wrappers for the *related* function above. These functions create a binary relationship within LifeNet that relates two pieces of evidence, E_0 and E_1, by a precompiled distribution. The positive dimensions of "x", "y", "z", and "time" are used to specify the relationships of *east*, *south*, *above*, and *after* respectively.

$d \equiv$ Real-numbered distance between the two LifeNet evidences, E_0 and E_1, in the specific dimension of the relationship

$E_i \equiv$ LifeNet evidences that are related by the distance d

- around(d, E_0, E_1)

  Creates a relationship within LifeNet that is used mainly for relating modal phenomena in the *north-south-east-west* plane. The assumed mixture of Gaussians for this distribution is a circle with density at distance, d, in the plane of the ground, which are the LifeNet dimensions of x and y internally.

  $d \equiv$ Real-numbered distance between the two LifeNet evidences, E_0 and E_1

  $E_i \equiv$ LifeNet evidences that are related by the distribution D

A.5 Inference Commands

- infer(E)

  Creates a new set of LifeNet evidence that is the posterior distribution that results from the prior evidence, E, which includes all of the concepts that have been related to these concepts through the LifeNet network. This function invokes all of the critical belief propagation agents (spatial, temporal, existential) within the LifeNet reasoning algorithm.

  $E \equiv$ LifeNet evidence that functions as the prior distribution over all modal phenomena that are used by the critical belief propagation agents that reproduce and spread in parallel over the LifeNet network, computing the posterior distribution when considering all of the relationships that have previously been programmed into LifeNet.

A.6 Data Structure Commands

- set(m, E)

  Binds the LifeNet evidence, E, to the LifeNet phenomenological mode, m, which can later be used within the LifeNet programming language to quickly refer to

the evidence, E. This command is the beginning of a self-reflective programming language that allows first-person Commonsense probabilistic inference to become a part of a full programming language with symbolic variable references as part of the probabilistic inference process itself—getting closer to a self-reflective programming language based on the current probabilistic context of the reasoning algorithm. This function marks the beginning of the development of the *deliberative* layer that will function above the *reactive* layer of the LifeNet reasoning algorithm.

$m \equiv$ Phenomenological mode specifying the symbolic reference for the phenomena that is to serve as a *k-line* (Minsky 1985) reference to the evidence, E.

$E \equiv$ LifeNet evidence that subsequently (in the current context) can be referenced simply by the symbolic mode, m.

Appendix B

ConceptNet Semantic Relationships

1. *conceptually-related-to*
2. *superthematic-k-line*
3. *thematic-k-line*
4. *capable-of*
5. *is-a*
6. *effect-of*
7. *location-of*
8. *capable-of-receiving-action*
9. *motivation-of*
10. *desire-of*
11. *property-of*
12. *used-for*
13. *last-subevent-of*

14. *part-of*

15. *subevent-of*

16. *defined-as*

17. *desirous-effect-of*

18. *made-of*

19. *prerequisite-event-of*

20. *first-subevent-of*

Glossary

commonsense computing	Vision of computation where computers have the set of general knowledge and ways of reasoning that a given community shares, so that computers can have a deeper understanding of humans and become a more integral component of daily life, 13
commonsense object	Physical object that a human might use commonly to solve everyday problems, such as a "stapler" solving the common problem of keeping papers together in a document. Like all commonsense, these objects are specific to the social cultures, groups and clubs to which the individual belongs. In general, a commonsense object is an object that all people within a context would consider to be a common everyday object, 21

commonsense phenomenon
: Mental state that a given "club" or group of people share; for example, a specific sensory experience that one might be able to express in conceptual human-language terms. Any given group of people will most likely share language capabilities that provide the ability to recall large sets of shared commonsense phenomena that are not necessary human-language concepts themselves, 16

concept
: Human-language Unicode string representing a human-language phrase, which functions as the primary mode of indexing the ConceptNet reasoning algorithm, 16

critic-selector model
: Theory of how humans perceive, reason, and act. Minsky (2006) introduced the critic-selector model as a hierarchical implementation of the agents within the society of mind where critics and selectors are two specific types of agents. The model fits within a six-layered model of human intelligence, which has increasing levels of abstraction from the peripheral aspects of intelligence that interface directly with the physical world and the human sensations and motor-control, 37

generative function — Computational process that takes a set of arguments and returns a set of data derived from those arguments. As computational processes, generative functions assume an algebra of data processing computation. This generative functions in this paper assume an algebra of stacked poset concatenation functions, which are easily implemented on classical digital computers, but in general, generative functions could assume biological neural networks or quantum computers as other algebras of data generative computation, 46

greedy compression search — Search algorithm that begins with a list of uncompressed data, L. For all of the data in L the largest contiguous repetitive section, x, of data is found. Every datum in L containing x is removed from L and split into smaller non-contiguous pieces that do not contain x. These smaller non-contiguous pieces are appended to L, and the process of removing redundant sections of data continues until no such sections exist in L, at which point L will contain the leaves of the perception lattice structure, 49

LifeNet	Model that functions as a computational model of human life and attempts to anticipate and predict what humans do in the world from a first-person point of view. LifeNet utilizes a commonsense knowledge base (Singh et al. 2002) gathered from assertions about the world input by the web community at large. In this work, we extend this commonsense knowledge with sensor data gathered *in vivo*, 14
LifeNet story	Partially ordered sequence of events expressed as conceptual human-language phrases (Unicode strings), 18
mode of perception	Consistent relationship between elements within a subset of data. This consistent relationship defines a regularity that serves to organize a larger set of data. A mode of perception can act as a manifold in the space of percepts such that percepts lie on a specific manifold or they lie off of that manifold. Given that a set of data points lie on a mode of perception, these data points contain percepts that vary along the mode. The specific percepts that define these data points are implicitly constrained by this manifold or mode, 44

perception lattice
: Lattice data structure that represents the generative functional explanation for a given set of data. This lattice structure is used for many algorithmic operations over perceived data. For example, a perception lattice could be used to find the most likely top-down explanation of bottom-up perceptions, or alternatively, a perception lattice could be used for the projection of low-level details given high-level evidence, 47

phenomenon
: More general sense of the ConceptNet "text phrase" type of knowledge and forms the basic index to the LifeNet reasoning algorithm. The set of LifeNet phenomena includes all ConceptNet concepts as well as groups of sensor data. A recognized mode of text or sensor datum is a phenomenon functioning as a percept, while a contextual mode of text or sensor datum functions as a top-down projection phenomenon, 16

Index

Bibliography

Baron-Cohen, S. (1995), *Mindblindness: An Essay on Autism and Theory of Mind*, MIT Press.

Bergmann, R., Pews, G. & Wilke, W. (1994), 'Explanation-based Similarity: A Unifying Approach for Integrating Domain Knowledge into Case-based Reasoning for Diagnosis and Planning Tasks', *Topics in Case-Based Reasoning, Springer Verlag* pp. 182–196.

Chung, H., Chung, J., Eslick, I., Kim, W., Myaeng, S. H. & Bender, W. (2006), Multi-lingual ConceptNet.

Daphne, K. & Pfeffer, A. (1997), 'Object-oriented Bayesian Networks', *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97)* **313**, 302–313.

Dong, W. & Pentland, A. (2006), 'Multi-sensor Data Fusion Using the Influence Model', *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks.*

Eagle, N. & Pentland, A. (2005), 'Reality Mining: Sensing Complex Social Systems', *J. of Personal and Ubiquitous Computing* **10**(4), 255–268.

Eden, U. T. (2004), 'Dyanamic Analysis of Neural Encoding by Point Process Adaptive Filtering', *Neural Computation* **16**, 971–998.

Edmison, J., Lehn, D., Jones, M. & Martin, T. (2006), 'E-Textile Based Automatic

Activity Diary for Medical Annotation and Analysis', *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks.*

El Kaliouby, R. & Robinson, P. (2005), Mind Reading Machines: Automated Inference of Cognitive Mental States from Video, PhD. Dissertation, PhD thesis, University of Cambridge UK.

Falkenhainer, B., Forbus, K. & Gentner, D. (1989), 'The Structure Mapping Engine: Algorithm and Examples', *Artificial Intelligence* **41**(1), 1–63.

Fulford-Jones, T., Wei, G. & Welsh, M. (2004), 'A Portable, Low-Power, Wireless Two-Lead EKG System', *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.*

Gentner, D. (1983), 'Structure Mapping: A Theoretical Framework for Analogy', *Cognitive Science* **7**(2), 155–170.

Hecht-Nielsen, R. (1990), *Neurocomputing*, Reading, MA: Addison-Wesley.

Heidegger, M. (1962), *Being and Time*, Harper San Francisco.

Hendrickson, B. & Leland, S. (1995), The Chaco User's Guide: Version 2.0, Technical report, Sandia.

I.E.E.E. (2003), 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), Technical report, IEEE Computer Society.

Ihler, A., Fisher, J., Moses, R. & Willsky, A. (2004), 'Nonparametric Belief Propagation for Self-Calibration in Sensor Networks', *Proceedings of the Third International Symposium on Information Processing in Sensor Networks* pp. 225–233.

Jepson, A. & Richards, W. (1994), 'What makes a good feature?', *Proceedings of the 1991 York Conference on Spatial Vision in Humans and Robots* pp. 89–125.

Kahn, R., ed. (1978), *Distributed Sensor Networks*, Carnegie-Mellon University.

Kochenderfer, M. & Gupta, R. (2004), 'Common Sense Data Acquisition for Indoor Mobile Robots', *AAAI.*

Kohonen, T. (1995), *Self-Organizing Maps*, Berlin: Springer-Verlag.

Liao, L., Fox, D. & Kautz, H. (2004), 'Learning and Inferring Transportation Routines', *Proc. of the National Conference on Artificial Intelligence (AAAI-04).*

Liao, L., Fox, D. & Kautz, H. (2005), 'Hierarchical Conditional Random Fields for GPS-based Activity Recognition', *Proc. of the International Symposium of Robots Research.*

Lifton, J., Feldmeier, M. & Paradiso, J. (2005), *The plug sensor network.*

Liu, H. & Singh, P. (2004), 'ConceptNet: A Practical Commonsense Reasoning Toolkit', *B.T. Technology Journal* **22**(4), 211–226.

Luprano, J., Sola, J., Dasen, S., Koller, J. & Chetelat, O. (2006), 'Combination of Body Sensor Networks and On-Body Signal Processing Algorithms: the practical case of MyHeart project', *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks.*

Madabhushi, A. & Aggarwal, J. (1999), 'A Bayesian Approach to Human Activity Recognition', *Second IEEE Workshop on Visual Surveillance* pp. 25–30.

McFarland, D., McCane, L. & Wolpaw, J. (1998), 'EEG-Based Communication and Control: Short-term Role of Feedback', *IEEE Transactions on Rehabilitation Engineering.*

Merleau-Ponty, M. (1962), 'Phenomenology of Perception and an introduction: the spatiality of one's own body and motility'.

Minsky, M. (1974), A Framework for Representing Knowledge, Technical Report 306, MIT-AI Laboratory.

Minsky, M. (1985), *Society of Mind*, Simon and Schuster, Inc.

Minsky, M. (2006), *The Emotion Machine*, http://web.media.mit.edu/~minsky/.

Mohri, M., Riley, M., Hindle, D., Ljolje, A. & Pereira, F. (1998), 'Full expansion of context-dependent networks in large vocabulary speech recognition', *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.*

Moore, M. & Kennedy, P. (2000), 'Human Factors Issues in the Neural Signals Direct Brain-Computer Interface', *ASSETS.*

Oliver, N., Horvitz, E. & Garg, A. (2002), 'Layered Representations for Human Activity Recognition', *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces* pp. 3–8.

Paskin, M., Guestrin, C. & McFadden, J. (2005), 'A robust architecture for distributed inference in sensor networks', *Fourth International Conference on Information Processing in Sensor Networks* pp. 55–62.

Pearl, J. (2000), *Causality: Models, Reasoning, and Inference*, Cambridge University Press.

Piaget, J. (1947), *The Psychology of Intelligence*, Littlefield, Adams and Co.

Rosenweig, M., Leiman, A. & Breedlove, S. (1999), *Biological Psychology: An Introduction to Behavioral, Cognitive, and Clinical Neuroscience*, Sunderland, MA: Sinauer Associates, Inc.

SenSys-151 (2006), 'BehaviorScope: An Architecture for Studying and Interpretting Behaviors with Distributed Sensor Networks', *SenSys.*

Singh, P. (2005), EM-ONE: An Architecture for Reflective Commonsense Thinking, PhD thesis, Massachusetts Institute of Technology.

Singh, P. & Williams, W. (2003), 'LifeNet: a propositional model of ordinary human activity', *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture.*

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T. & Zhu, W. L. (2002), 'Open Mind Common Sense: Knowledge acquisition from the general public', *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems.*

Subramanya, A., Raj, A., Bilmes, J., & Fox, D. (2006), 'Recognizing Activities and Spatial Context Using Wearable Sensors', *Proc. of Conference on Uncertainty in AI (UAI).*

Tapia, E. M., Intille, S. & Larson, K. (2004), 'Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors', *Proc. Pervasive.*

Thiemjarus, S., Lo, B. & Yang, G. (2006), 'A Spatio-Temporal Architecture for Context Aware Sensing', *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks.*

Warren, S. (2004), 'AI FAQ Neural Nets, Part I: How many kinds of Kohonen networks exist?'.

Weiser, M. (1991), 'The Computer for the 21st Century', *Scientific American* **265**, 94–104.

Weiser, M., Gold, R. & Brown, J. (1999), 'The origins of ubiquitous computing research at PARC in the late 1980's', *IBM Systems Journal.*

Weiss, Y. & Freeman, W. (1999), Correctness of belief propagation in Gaussian graphical models of arbitrary topology, Technical Report Report No. UCB/CSD-99-1046, University of Berkeley California.

Wyatt, D., Philipose, M. & Choudhury, T. (2005), 'Unsupervised Activity Recognition Using Automatically Mined Common Sense', *AAAI* pp. 21–27.

Yedidia, J., Freeman, W. & Weiss, Y. (2000), 'Generalized Belief Propagation', *Advances in Neural Information Processing Systems (NIPS)* **13**, 689–695.