Acquiring Common Sense Knowledge from Smart Environments

Rémi Barraquand and James Crowley
Grenoble Informatics Laboratory (LIG)
INRIA Grenoble Rhônes-Alpes Research Center
655 Ave de l’Europe
38330 Montbonnot, France
{remi.barraquand,james.crowley}@inrialpes.fr

Abstract
We present an approach for acquiring common sense knowledge from social interaction. We argue that social common sense should be learned from daily interactions using implicit user’s feedbacks and requires shared understanding of social situations. A service-oriented architecture, inspired from cognitive science, that foster mutual understanding between a smart environment and its inhabitants is presented. The method makes use of ConceptNet to work with common sense knowledge. We are able to successfully use and learn common sense knowledge.

Introduction
Recent technological progress has led to widespread use of digital devices that provide services for communication, commerce, entertainment and assistance. Those services have come to pervade nearly all aspect of our daily life, entering our home and office. As these services become increasingly interconnected and multifunctional, their interfaces have grown impossibly complex, often requiring an extensive learning effort to operate. The field of human-computer interaction has long been interested in providing more friendly interfaces and interaction. Recent research (Reeves and Nass 1996), backed by a series of experiments, indicates that people treat and respond to media (computers, televisions, etc.) in much the same way as they treat and respond to each other during everyday social interaction. Unfortunately, we lack a technology to allow information and communication services to interact with humans in a polite, socially compliant manner.

Ambient intelligence addresses the problem of making computer aware of human activity (current task, availability, focus of attention). Built on ubiquitous computing, ambient intelligence relates to electronic environments that are sensitive and responsive to the presence of people. Such environments are referred as intelligent environments or smart environments. However, activity understanding in smart environments is an open problem and relies on the ability to perceive and understand the social situation and affective reactions of humans. Furthermore, it requires to make sense of social roles played by human when interacting: systems are unaware of human goals and intentions.

Social common sense refers to the shared rules for polite, social interaction that implicitly rule behaviour within a social group. To a large extent, such common sense is developed using implicit feedback during interaction between individuals. Thus our goal in this research is to develop methods to endow an artificial agent with the ability to acquire social common sense using day to day interactions with people. We believe that such methods can provide a foundation for socially polite man-machine interaction, and ultimately for other forms of cognitive abilities.

In previous work (Barraquand and Crowley 2008) we have focused on a key aspect of social common sense: the ability to act appropriately in social situations. We proposed to train an association between behaviour and social situations using machine learning techniques. However, the main difficulty of this method is to ensure a shared understanding of the social situation between the user and the system.

In this paper, we propose to help the construction of a mutual cognitive environment between a smart home and its inhabitants by using common sense. First, we review existing initiatives to acquire common sense knowledge and propose an alternative. After introducing background from cognitive science we propose a framework for ubiquitous computing that foster mutual understanding between an environment and its inhabitant. The framework is then illustrated by an experiment we conducted in a smart environment. In this experiment common sense knowledge is used to automate and help the convergence toward mutual intelligibility.

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
Acquiring Common Sense Knowledge

Existing Approaches

Over the last decades there have been many attempts at collecting large common sense databases. Among the most popular initiatives let us cite the Cyc project, WordNet and Open Mind Initiative. The Cyc project was first initiated by Doug Lena in 1984 and tries to formalise common sense knowledge into a logical framework (Lenat 1995). Within Cyc, each common sense fact is written in a formal language CycL, arranged in an ontology and tagged with contextual information. One can think of Cyc as an expert system with a domain that spans all everyday objects and actions. The main drawback of this approach is that assertions are largely handcrafted by knowledge engineers, which requires an important amount of resources. The WordNet project began a year after Cyc at Princeton University in 1985. WordNet is a lexical-conceptual framework (Fellbaum 1998) that groups English words into sets of synonyms called synsets. WordNet distinguishes between nouns, verbs, adjectives and adverbs. Each word is provided with a short and general definition, and is integrated in a simple semantic network where words are linked to other words using various semantic relations. WordNet has encountered a great success among the community in the reason of its simplicity. WordNet is also available in other languages.

While WordNet and Cyc are both largely handcrafted by knowledge engineers, the Open Mind Initiative (Singh et al. 2002) goal is to build and utilise a large common sense knowledge base from the contributions of many thousands of people across the Web. The project started at MIT in 1999 with the objective to have people teach the system about everyday common sense. Unlike Cyc, all of the work is done by an automatic inference engine and natural language processing to make sense and organize the knowledge instead of the person entering the knowledge. From this database, the authors built up different projects. Among them, ConceptNet (Liu and Singh 2004) is a semantic network representation of the Open Mind Common Sense knowledge base. ConceptNet takes its inspiration in the range of common sense concepts and relations in Cyc, and in the ease-of-use of WordNet. ConceptNet contains millions of assertions such as “baseball IsA sport”. One can ask ConceptNet with different questions such as what relation connects “dog” and “bark”.

Social Approach

All this approaches focus on acquiring general common sense knowledge about what humans in common will agree. The objective of our research is to provide systems with the kind of common sense knowledge that will help them interact and engage socially with people. What kind of common sense knowledge will systems need for such interaction? Is human common sense adapted? Or do we need system’s common sense? Well, in order to provide a first answer to these questions, we propose to return to the definition of social common sense:

1 Social common sense refers to the shared rules for polite, social interaction that implicitly rule behaviour within a social group. To our knowledge there are no existing databases that provide indication of what a Roomba, an Aibo or even a cellphone should do or not do in a social group in order to be accepted by people. Each user has to configure their own devices to make them do what they expect to. Although it is possible (Lieberman et al. 2004) to adapt human common sense for the use in application, to what cost? We believe that we should capture each of these personal system’s experiences in order to create a “system’s common sense” instead; hence that system’s common sense must be forged from social interaction. In short, lets systems in common build their own common sense knowledge.

2 To a large extent, such common sense is developed using implicit feedback during interaction between individuals. Those precious pieces of knowledge have to be captured in day to day interaction, while applications or services interact with their users. There are billions of cell phones actively used everyday, most of them providing precious services to their users. Personal robots are soon going to be part of our lives. Each user’s feedbacks towards a computer or an application must be used to acquire systems’ common sense knowledge.

3 Common sense is what a community or group of people in common will agree. What do systems in common will agree? We would never know if we don’t let them tell us. For instance suppose we have knowledge of what smart phones in common will agree. We would be able of designing socially aware cell phone using their common sense knowledge. More importantly such common sense knowledge will capture their perceptions of the world, their understanding, not ours.

Like the Open Mind Initiative trying to learn common sense from the general public, we turn our approach towards the general public’s systems. Smart environments are a really interesting case to study as they regroup most of the fields in computer science: they are like robots turned inside out. Our first initiative towards this approach has been to develop methods to train an association between behaviour and social situation using machine learning techniques. The system is able to acquire its own experience from daily interaction with people. As the system interacts with its user, it is able to take more appropriate actions using its past experience. However, as stated earlier, such abilities rely on the assumption that both human and machine share a common understanding of the situation. Otherwise the system experiences will be full of misunderstanding. Mutual understanding is thus a requirement and a necessary step toward designing sociable system. In the next section, we review cognitive literature and attempt to provide a simple framework for the co-construction of mutual cognitive environments between the user and the system.
Gaining a Mutual Understanding

Salembier et al. (Salembier and Zouinar 2004) review research that points out the importance of the mutual access to contextual information in collaborative work. The better the mutual understanding, the greater the collaboration. This mutual understanding is not necessarily achieved by sharing mutual knowledge, but rather by the concept of mutual manifestness. Introduced by Sperber and Wilson, this concept is weaker but empirically more adequate than the theory of mutual knowledge which has the characteristic to produce regression at infinity.

Mutual Intelligibility

For Sperber and Wilson, “a fact is manifest to an individual at a given time, if and only if, this individual is able at this time to represent this fact mentally and to accept its representation as being true or probably true” (Sperber and Wilson 1995). Following this notion, Sperber and Wilson define the one of cognitive environment. A personal cognitive environment (PCE) is defined as whole facts which are manifest for a given individual. A shared cognitive environment (SCE) indicates all the facts which are manifest to several individuals. This simply means that they are able to perceive or deduce the same facts, and not that they share a belief, a knowledge, or a representation concerning those facts. A mutual cognitive environment (MCE) indicates a shared cognitive environment in which the identity of individuals who have access to this environment is manifest. As they share the same environment, they can establish an interaction in relation to their common perception of contextual events. In this theory, communication is defined as the process of making certain set of facts more manifest than other to another agent. This might induces a set of inferences, called cognitive effects, leading in return to the modification of the PCE of this other agent.

In his research on collaborative work, Salembier then points out that the notion of cognitive environment does not take into account the activity of individual. He proposes thus (Salembier and Zouinar 2004) the notion of shared context, which reduces mutual cognitive environment through activity filtering. Shared context is defined as a set of contextual information or events mutually manifest for a set of actors, at a given time in a certain situation, taking into account their perception and cognitive abilities, their task, and current activity.

We believe that social common sense acquisition must be a continuous collaborative process in which people and machines learn from each other. The success of this collaboration depends on the existence of a shared context, furthermore context has been identified as a key issue in human computer interaction. As a conclusion, we argue that mutual intelligibility is the key towards a social human-computer interaction.

Situation Models

In the previous section, we emphasize the importance of mutual intelligibility between both human and system. However, it is not clear what defined the facts that compose a cognitive environment. Johnson-Laird in (Johnson-Laird 1981) introduced situation model as cognitive theory for human mental models. A situation model is commonly defined as consisting of entities and of relations between those entities. While this model, as well as much of the subsequent literature in this area, has been concerned with spatial reasoning or linguistic understanding, these concepts can be adopted for the construction of software systems and services for understanding social interaction. In (Crowley, Reignier, and Barraquand 2009), the authors describe the use of situation models for observing and understanding activity in order to provide context aware services. Situations are defined as a set of relations (predicate) between entities (agents, objects or abstract concepts) where entities are sets of properties.

Using this formalism, we define a fact as an entity or a relation that can be either observed or deduced by an agent. A PCE is then defined by the set of entities and the set of relations between these entities that are manifest by an agent. In the following, the shared context between agents (human or machine) is defined by the intersection of their situation models taking into account their perception and cognitive abilities, their task, and current activity. The wider the intersection, the greater the mutual understanding.

Framework For Mutual Intelligibility

A pervasive environment is an environment where stationary and mobile devices can communicate between each other in order to provide services to the user. The term service is used here in its most general form. Generally, it will refer to assistance that informatics systems provide to people and are denoted as user-service. User-service can be designed as software agent that assists and interacts with people. This definition of service is to be distinguished with the one used in software architecture. Indeed, a classical architectural solution used to solve constraints of pervasive computing is to adopt a Service Oriented Architecture (SOA). In this context, a service is a piece of software with a well defined way to access its provided functionalities. We refer such service as software-service. A software-service then is either a perceptual service, actuator service or data processing service.

The following framework proposes to develop intermediate-services between user-services and software-services that will allow the SE to gain mutual understanding with the user. Each intermediate-service collaborates by making its information available to others. The resulting form a virtual hypergraph that we refer as factual network (network of facts). The information is not centralized but widespread among services achieving a global consciousness. In the following, u-service, i-service, s-service respectively refer to user, intermediate and software service.

The role of i-service is to transform raw information provided by s-service into shareable relevant facts: entities and relations. Extracted entities and relations are made available to any service over the network. Aggregating this information leads to a factual network allowing the SE to build its PCE. With this settings, users’s activities directly affect the environment’s PCE. Thus, the interaction between the environment and the user is seen as the process of changing each other PCE in order for the MCE to become wider.

Entities are abstract objects that can have many facets.
A facet is defined by a group of properties and is associated with a probability. The higher the probability associated with a facet, the more reliable the information. Each i-service can provide facets to any entity. For instance, a “3dTracker” i-service can provide entities with spatial facets. An “idFinder” i-service could process every entity having spatial facets and tries to apply facial detection algorithms on them. As soon as a face is detected and recognized, the i-service will provide for the given entity an identity facet. In our framework, each i-service collaborates to a better understanding of the undergoing social activity.

We distinguish a specific class of facets: interpretation facets. Interpretation facets are facets that provide an assignment of meanings to entities taking into account their relations and facets. For example, an entry with spatial and identity facets may be given an interpretation facet stating it is human. Such interpretation needs some background knowledge, that can be brought and embedded in i-services by developers; it can also be learned by the system or even taught by human. For instance, using u-services, we can let the user provides interpretation facets to existing entities. The next section details a possible use of that framework and introduce how common sense knowledge can automatically provides interpretation facets to entities. We also, demonstrate how a u-service can be used to reversely acquire common sense.

**Experiment and Results**

To illustrate our framework, we conducted an experiment in our laboratory: a smart environment. In this experiment, both the environment and the users have the opportunity to produce cognitive effects to one another. (users and inhabitants are alternatively used). The following sections review the services deployed in the environment and detail the use of ConceptNet for the design of an automatic interpretation facets provider.

**Experimental Setup**

The experiment was performed within the INRIA-Grenoble Smart Environment experimental facility. This environment (Smartroom) is an experimental laboratory equipped with furniture for simulating domestic, office and meeting environments, while observing activities with large number of cameras, microphones and other sensors. We used the OMiSCID middleware (Emonet et al. 2006) as framework for our service oriented architecture. Each device in the environment comes along with a s-service that makes accessible to other services data and controls the device. For instance each camera is related to a grabbing service that provides real time image stream. Such information is used by a “3DTracker” which is able of detecting and following moving objects in the environment. On top of these s-services we have developed a set of i-services and u-services.

**Environment’s Services**

iTracker is an entity and facets i-service provider. This i-service connects to the 3dTracker s-services and creates an entity for each target. Each entity is associated with a spatial facet (x,y,z coordinates), a shape facet (approximation of the target shape and its principal orientation) a velocity facet (current speed and orientation). Using background subtraction techniques and velocity the service is able to distinguish humans from objects. Thus it allows the iTracker to add interpretation facets such as: isA object, isA person.

iPosture is a facet provider. It analyses entities that are associated to spatial and shape facet with an isA person interpretation facet in order to provide them with posture interpretation facets: indeed, using an SVM classifier (Brabzicza, Reignier, and Crowley 2007) the service is able of recognizing if a human shape corresponds to a standing, sitting or laying person. Interpretation facets attributed to human entity are given as follows: HasProperty posture.

iColor is a facet provider. It provides color facet to any entity having a spatial and a shape facet. The extracted color corresponds to the dominant color estimated from the entity color histogram. When the color can be associated with a usual name (red,green,yellow) the service furnishes the entity with a color interpretation facet of the form "HasProperty color".

iActivitySpace is an entity, relation and facet provider. It provides entities that represent regions of interest in the environment. A region of interest is a region where an activity often occurs. Each of those regions are extracted by watching user’s activity (entities with posture facet). For instance a region in the environment where inhabitants stay often is of high interest. In order to extract such region, an algorithm creates for each posture an accumulation grid on which we accumulate mixture of gaussians representing the probability $P(p|x,y)$ of seeing the posture $p$ at the position $x,y$. For instance each time an entity has the interpretation “HasProperty sitting” we accumulate on the sitting-grid a gaussian distribution centered on the entity coordinates. Using this accumulation grid and clustering algorithms the service is able to extract regions of interest (maximuns in the gaussian mixture distribution). Each of those region is associated with an entity and relative facets (spatial, shaped). An interpretation facet is also attributed to region entities stating “usedFor posture”. The service also provides relation between entities having spatial facets. For instance if an entity is contained in another one then the relation AtLocation will be added between the containee and the container. For this experiment only two spatial relation were considered: NearBy and AtLocation.

iConceptNet is a facet provider. It provides interpretation facets to entities by using common sense knowledge. The services use the relation between entities as well as their facets to formulate simple queries. These queries allow the
service to identify concept that might characterise entity using the ConceptNet framework. The most relevant concepts are then used as interpretation facets. Such facet is of the form “isA concept” and come along with a probability. The details of how ConceptNet framework is used is detailed below.

**uTable** is a user-service that exposes a projection of the smartroom’s personal cognitive environment to the user by the mean of a user interface. In this context a projection means to filter entities and relations according to a certain criterion then to display the result using a certain layout. For our use cases the service only keeps spatial entities and spatial relations, i.e. temporal entity are removed by this projection. The layout presents the information with a spatial organisation using the spatial and shape facets of entities. Each entity is decorated with its related interpretation facets. Spatial relation (NearBy, AtLocation) are also displayed to the user. The user can at any time interact with the interface to take different actions. First the user can remove interpretation facets that he judges not correct. Secondly it can add interpretation that are missing. By interacting with the user interface the user provides the service with meaningful information: common sense knowledge. Any user intervention is captured by the user-service that updates the smartroom’s personal cognitive environment and most importantly forwards the result to the ConceptNet framework. The information is made available to ConceptNet by using methods such as the voting system or by the insertion of a new assertion.

**Using Common Sense**

Since each i-service provides information in term of facets, entities and relations, the union of this information results in a semantic network. It is then possible to query this network to provide additional information. Each relation and facet provides features that describes entities in the semantic network. Using these features we can query ConceptNet to identify concepts that provides the best meanings of entities. Let’s now take a look at Figure 1 and note as ‘e the entity corresponding the spatial region automatically extracted identified by id n°7. Using its current facets and relations one can extract the following features: “?e HasProperty blue”, “?e UsedFor sit”, “?e AtLocation to something that IsA home”, “?e LocatedNear something that IsA furniture”. Features are extracted from this virtual network of facts, dynamically updated by i-services and u-services, thus composed by humans and systems information. Each facets being associated with a probability we can use this probability to compute the probability of a features as well. Using this list of weighted features we can ask ConceptNet to get the mostly related concepts. There are then two ways of proceeding.

The first method is to create a query for each feature and combine the result to get the most likely concepts. ConceptNet for each query returns a set of assertion together with a score. Each assertion leads to a possible interpretation of ‘e that we extract and associated with the score of the assertion. After processing all the features we obtain a vector of concepts likely to represent our entity ‘e. We then take the top most ranked concepts according to their score and provide ‘e with interpretation facets of the form “?e IsA $concept”.

The second method uses AnalogySpace (Speer, Havasi, and Lieberman 2008). Using data from the Open Mind Common Sense project. AnalogySpace represents knowledge as a matrix of concepts along one axis, and features of those concepts along another, yielding a sparse matrix of very high dimension. This matrix is reduced using singular value decomposition allowing to perform mathematical operations such as dot product between concepts vectors or features vectors. Thus analogy between concepts is determined by computing the dot product between vectors representing those concepts. In our situation features extracted from our semantic network are transformed to vector of features in AnalogySpace. Each of those vectors can be combined using a weighted sum taking the probability associated to each feature as factor. The most relevant concepts can then be obtained performing a dot product between all concepts and the resulting vector. This operation is worth taking because of the low dimensionality of the reduced matrix. Again, taking the top most ranked concepts let us augment entities with interpretation facets.

**Results**

We ran different scenarios using different spatial organisations of furnitures in the environment, alternatively activating/deactivating i-services. In all trials, spatial regions of activity were correctly extracted. Table 1 present a possible interpretation from the Figure 1. In bold is presented what the user has actually provided as interpretation vs. our automatic approach using common sense. Despite being suboptimal (processing multiple queries and combining their results is really expensive in time, a simple query might lead thousand of assertions) the first method provides acceptable results but relies too much on assertions’ score. For instance the bed is characterized as an apple because it has the property of being red and the assertion “an apple HasProperty red” has a high score in the database, making that assertion take advantage on the others. However this method is a lot more interesting when trying to identify user’s activity. Indeed a user is an entity in the environment like any other, and it is possible to infer activity regarding its relations and facets. For a query such as “?e IsA Activity, ?e AtLocation Home, person CapableOf ?e, chair UsedFor ?e, computer LocatedNear ?e”, the fist technique provides acceptable interpretation even though it only has a small quality of information for guessing the activity.

The second method is a lot faster and is more reliable for guessing furniture’s interpretation as it is less sensible to high scoring assertions. Reducing dimensionality allows for the creation of robust categories and better classification. For instance the SVD method is less sensible to color and property attributes than the first approach. On the other hand it has poor capabilities regarding activity guessing: indeed it appears that during the dimensionality reduction process none of the dimensions actually relate to activity which may explain why the resulting interpretations are far from being relevant. In order to prevent such disagreement we are actually looking at reducing dimensionality by introducing cus-


In this experiment the more the user behaves and interacts in the environment the better the mutual understanding; and at any time the user has the possibility of providing more accurate interpretation or banish an existing interpretation using the uTable’s user interface. For instance, most of the time when the user was working sitting on a chair close to his desk, the system identified gaming, reading and resting as the most likely activity. The user was able then to correct this information by entering what was manifest for him at this time using his vocabulary. Similarly, in the case of the bed being interpreted as an apple by the system, the uTable has allowed the user to banish this interpretation and to vote instead for bed or sofa. All the inputs provided by users were cached in order to be send back to ConceptNet using the voting API and by adding new assertions to the database. We evaluate the contribution to approximately 56 votes and 20 new assertions for only a few sets of informal experiments.

Conclusion and Further Work

Acquiring and using common sense knowledge in still an open and challenging problem. This paper addresses the problem of gathering social common sense which we argue must be learn through everyday interaction. We present a framework inspired from cognitive science allowing users and computers to acquire mutual understanding of social situations. Through this process common sense knowledge can be both used and acquired in a transparent way. The next step is to use this mutual understanding as input to machine learning algorithms in order to provide computers with the ability to act appropriately in social situations and thus acquiring system’s social common sense.

References


