Exercise Paper
Information Search and Retrieval
Graz University of Technology
WS 2009

Common Sense Knowledge

Christian Andrich
Graz University of Technology, chris_an@sbox.tugraz.at

Leo Novosel
Graz University of Technology, novosel@sbox.tugraz.at

Bojan Hrnkas
Graz University of Technology, bojan.hrnkas@student.tugraz.at

Supervisor
Dipl.-Ing. Dr.techn. Christian GÜTL
Institute for Information Systems and Computer Media (IICM), Austria
cguetl@iicm.edu and cguetl@acm.org

Abstract

This document is not a deep level resource for commonsense knowledge in computational technology, but it is an orientational overview on the subject. We will show the reasons for use of commonsense knowledge in computation science, and some methods to collect the commonsense knowledge. Furthermore, we will illustrate some practical usages of the commonsense in computational technology.

Contents

1 Introduction

Common sense knowledge is the set of all knowledge which is assumed that every ordinary human has. As soon as a person is born, his environment provides input for all his senses. This input gets processed by the brain. Thus a learning process is enforced, which leads to what we call common sense knowledge. This knowledge is one necessary pre-condition to be able to behave intelligent.

An example is, if you hold an object in your hand and then you let it out. You know that it will fall to the ground, not because you tried this with that special object before, but you know that's the way things behave due to gravity (of course, although if one doesn't know the concept of gravity, she will know that the object will fall down). That is an example for common sense knowledge about physics.

Another example would be, if you meet a person the first time, you know for sure that this person has parents without ever having talked to or about her. This is because you know a person could not exist if she had no parents.

Thus common sense knowledge is a quite natural thing for a human. But how does this apply to machines? Since the invention of computers, researchers tried to make them behave intelligently. One effort to achieve that goal is to feed computers with common sense knowledge. This is believed to be necessary due to the so called *knowledge principle*. The knowledge principle states that a system which is intended to behave intelligetly has to have knowledge about its task and especially the domain of the task [Feigenbaum, 1990].

There are several ways to get that kind of knowledge into computers. One of the obvious ways how to collect commonsense knowledge and convert it to a form that can be used by computers, is to do it ourselves - manually. This approach has been described in section 2.2. Commonsense, however, includes a huge amount of information and collectiong just a fraction of it manually takes a lots of time and energy. Therefore some attempts have been made to collect the commonsense knowledge using semi automatic 2.3 and automatic 2.4 methods of knowledge collection.

Collecting this huge ammount of data is one thing, but what can we do with it? What use can computer programs have for it, beside retrieving that information for us, which we actually do not need, since it is just common sense? Some very interesting usages of common sense knowledge like text analysis, translation and object recognition, and some particular AI topics are presented in sections 3.1 and 3.2.

It rather difficult to define what facts are a part of common sense. The extent of common sense is subject to change by social, cultural, geographical, temporal and other influences. What is considered common sense today, was not necessarily so fifty years ago. Common sense sometimes also includes "facts" that are utterly wrong.

1.1 Motivation

Since the beginnings of the artificial intelligence development, the experts have been aware that any intelligent system would have to have access to commonsense knowledge. Human beings connect "commonsense" with "good judgement", whereas "commonsense knowledge" in computational context represents the immense amount of small but well known facts that every human takes for granted. "Commomsense knowledge spans a huge portion of human experience, but is typically omitted from social communications" [Liu and Singh, 2004]. This means that since those facts are concerned as generally shared knowledge, they are absent from everyday communication. They are rarely used in written communication, and almost never in speech. Someone who writes home about a failed exam, usualy does not write that it is a bad thing, since it is commonsense. Without the commonsense knowledge, such message would not be apprehensible.

Therefore, some effort has been made to collect that commonsense knowledge and to "teach" the computers how to use it.

2 Knowledge Collection

To enable computers to make use of common sense knowledge, first of all, it has to be brought into the computers memory in some way. This raises several challenging tasks.

The first question is, where to get the data from? For this task, two main approaches exist. Firstly, the *human-based* approach. That means all the data is entered by humans into the database. Secondly the *automatic* approach. Here the knowledge is extracted from already existing data sources. There are also projects in which a mixture of both approaches, called *semi-automatic*, is used. Natually, all of these approaches have advantages and disadvantages, compared to each other, which will be discussed in the corresponding subsections below.

The next big question is, how to respresent the data in a way, that it can be handled efficiently. This means for instance, if there are statements like "A cat is a mammal" and "Mammals are animals", then we want to be able to derive new statements like "A cat is an animal".

Another thing would be to find out if things are related in some way. Let's take the following sentence as an example: "I took some yoghurt out of the fridge and ate it.". Here, for a human it is clear, that not the fridge was eaten, but the yoghurt. This is because yoghurt is something to be eaten and a fridge is not. Thus, if our system knows that "yoghurt" is stronger related to "eat" than "fridge" is, it can interpret the sentence correctly.

A similar case is, if two concepts have the same name and this name occurs in a statment our system shall interpret. If concept A is stronger related with the other concepts, which occur in the statement, than concept B, which has the same name as A, then it is more likely that concept A is the right one to choose.

To handle such issues, a graph based representation seems quite obvious and is indeed implemented in several systems (for an illustration see figure 2). However, an open and accessible standard to represent semantic data has been introudced with the Resource Description Framework (RDF). This can be, and actually is, used to share available data via RDF data dumps. To learn more about its capabilities, RDF is descriped in the next subsection.

2.1 Technical background

In this section some technics which can be used do process common sense knowledge are presented. The goal is, to give an overview of how the data can be processed but not how particular systems do that.

Resource Description Framework (RDF)[1]. RDF has been introduced in 1999 and is

[1] http://www.w3.org/RDF/ (Nov. 2009)

recommended by W3C[2]. It is a framework to describe ressources and their relations based on the XML syntax.

To identify ressources, qualified URIs are used. These are similar to ordinary URIs with an optional fragment identifier attached. Thus an URI defines a unique ressource, no matter if a fragment identifier is attached or not. An example would be "http://www.example.org/res#fr
A ressource can be described using *RDF Descriptors* which define its properties, for example:

```
<? xml version="1.0" ?>
<RDF xmlns     = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
     xmlns:ex = "http://example.org/terms/" >
  <Description about = "http://example.org/myRes" >
    <ex:myProperty> Value </ex:myProperty>
  </Description>
</RDF>
```

where the ressource "http://example.org/myRes" gets the property "Value" of type "myProperty".

Now, to connect ressources in some way, *RDF Statements* are used. Statements are triples which consist of a subject, a predicate and an object. The predicate describes the relation between the subject and the object. For instance if we want to state that "John D." is the creator of "The Document" we can define the statement:

```
<http://www.example.org/JohnD>
  <http://purl.org/dc/elements/1.1/creator>
  <http://www.example.org/TheDocument>.
```

using the so called N-Triples notation.

Of course this is just a very short description of the capabilities of RDF. Nevertheless it can easily be seen, that with a representation like that, a knowledge base can be built in a comfortable and accessible way, such that the data can be read and processed by further algorithms easily.

Ontology. Ontologies are used to represent meaning. A raw piece of data in a computer is just a string of bits. Thus, we can talk about meaning only if the data is interpreted in some way. This interpretation is necessary to be able to make use of the data. The aim of that is, to be able to define processes which serve purposes like extracting implicit knowledge, reasoning, detecting contradictions and other things.

An ontology is a formal representation of a set of concepts which are connected in some way. In practice they are often applied to certain domains and have been developed for Medicine [3], Biology [4], Linguistics [5] and many others.

To describe ontologies a number of formal languages have been developed like for example the Web Ontology Language (OWL[6]), which is based on RDF and inteted for web applications.

When designing ontologies several criteria shoud be considered. These help to increase the consistecy, interoperability between ontologies and some other aspects. In [Gruber, 1995], a study about design criteria for ontologies is presented. According to

[2] http://www.w3.org/ (Nov. 2009)
[3] http://diseaseontology.sourceforge.net/ (Nov. 2009)
[4] http://www.biopax.org/ (Nov. 2009)
[5] http://www.linguistics-ontology.org/ (Nov. 2009)
[6] http://www.w3.org/TR/owl-ref/ (Nov. 2009)

that paper the following principles should be considered: Clarity, Coherence, Extendibility, Minimal encoding bias and Minimal ontological commitment. These principles make it easier to evaluate the decistions which were made in the design process.

Due to ontology modeling is a lot of work, if done by hand, methods are being developed to automate this task. In [Pulido et al., 2007], a method is proposed to extract an ontology out of a set of unstructured documents. To achieve this, firstly the system analyses the input documents and builds a document space [Elliman and Pulido, 2003]. Then knowledge maps are built upon the extracted document space [Elliman and Pulido, 2002]. For this purpose unsupervised learning technics were used to identify the concepts of the ontology. As the results show, this approach worked out quite well and thus demonstrates the power of such automated algorithms.

2.2 Human-based approach for Knowledge Collection

As its name already states, with this approach the entire knowledge is entered by humans. The big advantage here is that the information can be controlled quite easy in terms of how it has to be entered and how it will be stored. Thus the quality of the result can more or less be guaranteed.

But the approach has the obvisual drawback that filling the data base with the amount of data, that is required to achieve an usefull system, is an enormous piece of work.

An example for such a system is WordNet [Fellbaum, 1998]. This is a database of words where the words are grouped as so called synonym sets (synsets). These synsets are again linked by lexical and semantic relations. To form this network all data was entered by members of the team which is responsible for WordNet and has therefore been a lot of work for a relatively small group.

In other projects, this effort has been distributed to a broad comunity. This has been done by offering an open interface online where everybody could enter data. Such a project will be described in more detail in the next subsection.

2.2.1 Open Mind Common Sense

The Open Mind Common Sense (OMCS) project was founded by MIT researchers in 1999. The idea was to collect information through a public, easy to use, interface. Thus everyone, who is willing to, can enter information because it is as easy as entering a phone number. A screenshot of that interface can be seen in figure 1. To imporve the quality of the input, concepts and statements from the database, can be rated.

Furthermore OMCS knowledge is available in multiple languages like English, Chinese, Portugese and some others. However, with currently about one million statements, English is by far the language with the most available data.

For storage, the entered data gets preprocessed such that the stemmed version of the words which represent the concepts is saved in a relational database. The relation of the concepts is stored corresponding to the entered type. Thus the concepts can easily be queried using SQL statements.

ConceptNet [Liu and Singh, 2004]. To make use of the information of OMCS it is added to the semantic network of ConceptNet. To achieve that, the sentences of OMCS are extracted using about fifty predefined rules such that the resulting nodes have a proper syntactic structure. After this *extraction phase* the nodes in the graph get normalised.

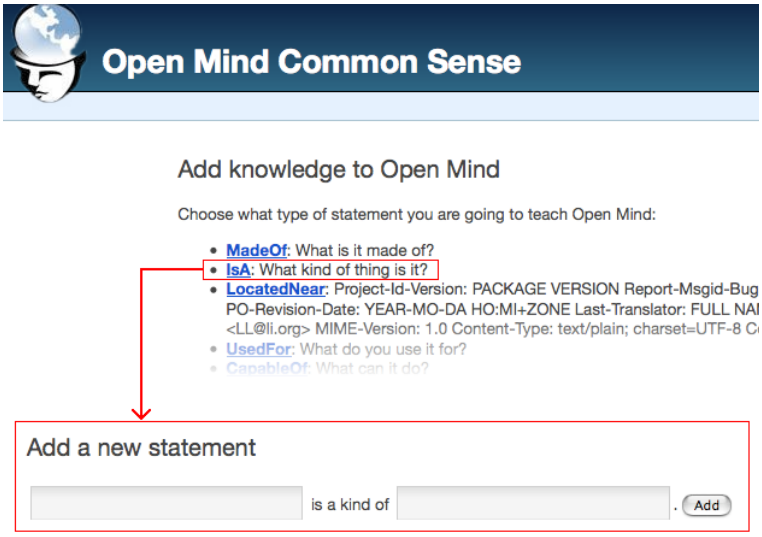

Figure 1: The OMCS interface was intented to be easy to use. After singing up, wich takes about one minute, it is possible to enter data using the shown interface. There are several relations things can have. Thus there are different predefined types of statments which chan be entered.

This means for example that nouns are converted to their base form (e.g. "apples" becomes "apple"), determiners are removed and other operations of that kind.

The last phase is the *relaxation phase*. Here the graph is improved by extracting implicit knowledge and thus filling knowledge gaps. More details of these phases can be found in [Liu and Singh, 2004].

An example for such an extracted graph is shown in figure 2.

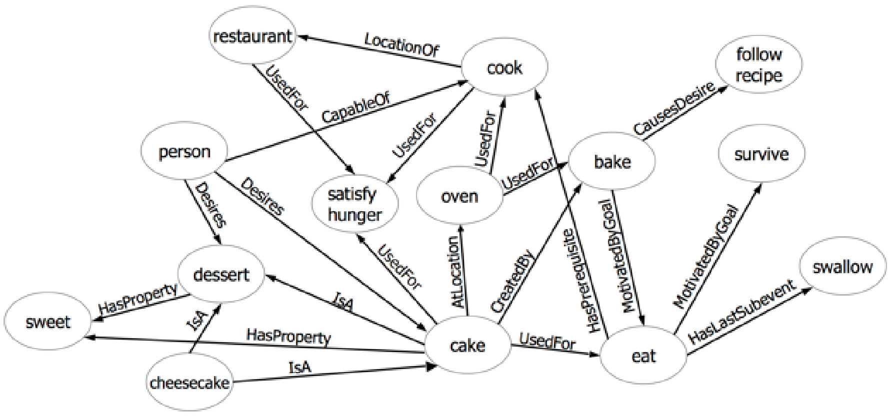

Figure 2: The data which is saved in ConceptNet can be represented as a graph. Here the concepts are the nodes and the relations between the concepts are the edges. Thus in this example one could derive that a cheesecake is for eating because of the relation "cheesecake is a cake" and "cake is used for eating". [Arnold, 2009]

Finally, to access the data, ConceptNet provides webservice functionallity, an RDF data dump and an API[7].

2.3 Semiautomatic knowledge extraction

As described in the previous section, establishing a reasonable database of common sense knowledge is a lot of work if done by humans. To alleviate that drawback, semiautomatic approaches make use of existing data, which is available in a well structured form. Thus the challange is, to access different databases which contain the according data in different representations, and then map it to an appropriate format to make use of it as common sense knowledge. In the next two subsections, tools which use such a semiautomatic approach are described.

2.3.1 Semiautomatic Method for Extracting Common Sense Knowledge

In [Storey et al., 2005] a tool is proposed, which uses data in different formats like semi-structured, entity relationship based and others to create an common sense knowledge base. The tool consists of the following components:

Knowledge Middleware and Reasoning Engine. This component is responsible for the import of data from external sources. Thus it must aware of each source, especially how the data is stored and furthermore how it shall be mapped to the internal representaion. After a database has been imported, reasoning mechanisms are appied to extract new knowledge, especially in combination with the data of other sources.

Common Sense Knowledge Repository. This component holds the data. It is similar to ConceptNet, which means concepts are stored which can have relations like *SubeventOf*, *EffectOf* or *CapableOf*.

Repository Maintenance Tools. With the maintenance tools, the user can manipulate the existing data to make sure it is consistent. It supports methods to add or erase data, to transform the data to different representations and some other tasks.

Repository APIs for Applications. This component provides methods to access the data and thus making it available for further applications.

Now the aim of this tool is to establish an intermediate component between different knowledge sources such that the set of the combined knowledge can be utilized.

2.3.2 Extracting Common Sense Knowledge from WordNet

In [Cankaya and Moldovan, 2009] the data of WordNet is used to extract commonsense axioms. For that purpose the Extended WordNet (XWN) was used. In XWN the words are semantically disambiguated and the glosses are parsed and transformed into a formal representation. After that, the glosses are transformed into semantic relations. The product is then called the XWN Knowledge Base (XWN-KB). The relations are similar to those of the previously described knowledge bases, which means of the kind of *is-a*, *part-of*, *has-property* and several others.

[7] http://conceptnet.media.mit.edu/#getting (Nov. 2009)

The extraction of commonsense axioms is performed in three main steps. Firstly, seed rules are entered manually. These are rules of the form *Common_Sense_Rule(r,p)* that commonsense rule r applies to property p.

Secondly, metarules are applied using the entered commonsense rules. There are three types of metarules: *Metarule 1: Has_Property(x,p)* & *Common_Sense_Rule(r,p)* $\rightarrow$ *Common_Sense_Axiom(r,x)*. This means, if concept x has property p, according to XWN-KB relations, and commonsense rule r applies to p, then r applies to the concept x. *Metarule 2: Has_Property(x,p)* & *Common_Sense_Rule(r,p)* & *Has_Hyponym(x,y)* $\rightarrow$ *Common_Sense_Axiom(r,y)*. The second rule satates that if concept x has property p and commonsense rule r applies to p and x has hyponym y then rule r applies to y. *Metarule 3: Has_Part(x,z)* & *Has_Property(z,p)* & *Common_Sense_Rule(r,p)* $\rightarrow$ *Common_Sense_Axiom(r,x)*. This means, if z is a part of x and z has property p and the commonsense ruel r applies to p, then r also applies to x.

The last step is, to check the constructed axioms manually to sort out incorrect axioms.

In the experiment, 27 commonsense rules were applied with which 2596 commonsense axioms were generated. The percentage of correct rules was about 98%. Although this approach has the drawback, that the commonsense rules have to be entered manually, and it strongly depends on the connectivity of the input concepts it seems quite promising in extracting commonsense knowledge.

2.4 Automatic knowledge extraction

In contrast to the semiautomatic approach, the automatic knowledge extraction is supposed to work with unstructured data sources and without further human effort. Especially Wikipedia is often used as a source of knowledge because it contains a huge amount of data of different domains. This data is stored in a way which makes it accessible for humans, but not for machines. In [Suh et al., 2006] for instance, a method is proposed to extract commonsense knwoledge from unstructured text by using natural language processing. This method was evaluated using the content of Wikipedia. And also in the Cyc project, Wikipedia was used as a source of knowledge [Sarjant et al., 2009].

In KnowItAll, which is another commonsense knowledge base, a automatic approach for knowldege extraction was implemented. This experiment is described in the next subsection.

2.4.1 KnowItAll

KnowItAll is supposed to extract its data from the web. The data extraction algorithm is initialized with an extensible ontology and some generic rule templates. Its goal is to collect as much data as possible from a wide range of domains.

The mechanism is described in [Etzioni et al., 2004]. In the first step, KnowItAll uses the so called *Extractor* to instanciate the rule templates. For instance, let "*NP1* such as *NPList2*" be such a rule template where *NP1* is a class, and *NPList2* is a noun phrase. Then the head of *NPList2* is considered to be of the class *NP1*. An example would be "cities such as Porto, Venice and Sevilla". From this phrase, KnowItAll would extract three instances of the class *city*. With these rules, not only instances of classes, but also relations between concepts can be extracted when using rules like "*NP1* is a *NP2*" for an *is-a* relation or "*NP1* is a part of *NP2*" for a *part-of* relation.

To get as much input as possible, KnowItAll formulates the rule templates as search queries. This task is performed by the *Search Engine Interface*. The query according to the example above would be "cities such as". These queries are then run in a number of search engines. The resulting pages are downloaded and searched with the according templates.

To evaluate the quality of the extracted data, the *Assessor* is used. Here the extracted instances are checked using search engines. To take again the city example, if the instance *Venice* of the class *city* was extracted, such a query could be something like "the city of Venice". The credibility is then derived from the number of search results. For that purpose a Bayesian Classifier is used to calculate the probability of the fact to be correct. With a probability of > 0.5 a fact is considered to be true. Nevertheless, the probability is also stored, to be able to trade precision for recall when queries are later entered into the system.

The big advantage of this statistical approach is the huge corpus of available information. In the experiment a constanly growing number of investigated websites and extracted facts could be achieved.

2.5 Commonsense Reasoning

Until now we have learned how different systems establish their knowledge bases. In this section Commonsense Reasoning will be described, as this contains the methods to actually make use of the knowledge. For this purpose some methods like context finding, analogy making and reasoning with fuzzy logic will be presented, while particular applications will be presented in section 3.

Context finding. Because many words can have several meanings, depending on the context, it is important to interpret such words in a statement correctly. Furthermore, some statements may be true in one context, but not in another. Thus, to be able to understand a statement, it is important to be able to interpret the context on different levels correctly.

In ConceptNet, for instance, a function called "GetContext" is implemented for that purpose [Liu and Singh, 2004]. Given a concept, this command returns all concepts which are strongly related to the passed concept, ordered by the strength of the relation. Thus the context of a statement can be derived by analysing the possible contexts of the concepts which occur in the statement.

Analogy making. Analogy making is a typical way for humans to learn and reason by comparing new situations to known ones. The basic assumption is, that if two situations A and B have similar relations, and there are relations known for A, but not yet for B, these relations may also apply for B.

In terms of Commonsense Reasoning this means that a system analyses relations, like *is-a*, *consists-of* or *has-property*, between concepts which are stored in a graph structure as described previously. Now different situations or statements may have a similar structure. Thus the system can look for similar patterns of different situations. Such a system is described in [Winston, 1980]. There a similarity meassure is proposed, which consideres the following issues: *Properties and Relations*: as described before, the accuracy of the analogy depends on the similarity of the properties and relations of the concepts. *Corresponding Comments*: in this system, relations can have describing comments. These comments are also taken into account when evaluating the analogy. *Classification Information*: the *is-a* relationship is taken into account. If two concepts are of the same kind, they are more similar. *Constraint Makes Some Relations More Important than Others*: relations that are considered to be more important are given a higher weight. *Matching Large Groups May Require Some Preliminary Classification*: because the computation can become intractable if too many concepts have to be considered, the pairings can be restricted such that matching concepts must be of the same type.

This approach has been evaluated within a framework, and it was shown, that for several tasks like finding specific or general laws, it leads to the desired results.

Also in ConceptNet a method is implemented to find analogies to a given concept. For that purpose, the sets of incoming edges of two concepts are compared. Thus, two concepts are considered to be analogous if their sets of incoming edges overlap where the analogy is stronger the more edges overlap.

Reasoning with fuzzy logic. Fuzzy logic was introduced to model things that cannot be described exactly. Typically, humans don't give exact declarations of things they describe. For instance, if a person is described, her height is usually described with terms like *small*, *tall* or *very tall*. Thus, fuzzy logic is based on fuzzy sets. This means, that an element can belong to a set to a certain degree. Therefore, if we take again our example of a persons height, if the person is about $180cm$, she may belong to the set of small people only a little, somewhat to the set of medium height, and a little bit more to the set of tall people.

Because also common sense knowledge is not always exactly defined, fuzzy logic can be applied for reasoning and inferring knowledge. In [Zadeh, 1986] such an approach is proposed. There, common sense knowledge is considered as *a collection of usually qualified propositions*. On these propostions fuzzy quantifiers, like *most*, *some* or *usually*, and fuzzy predicates, like *tall*, *small* or *high*, are applied to create fuzzy syllogisms of the form

$$\begin{array}{c} Q_1 A's \text{ are } B's \\ \underline{Q_2 C's \text{ are } D's} \\ Q_3 E's \text{ are } F's \end{array}$$

where Q_1, Q_2 and Q_3 are fuzzy quantifiers, and A, B, C, D, E and F are interrelated fuzzy predicates. Two of the most important syllogisms are the following: *(i) Intersection/product syllogism:*

$$\begin{array}{l} Q_1 A's \text{ are } B's \\ \underline{Q_2 (A \text{ and } B)'s \text{ are } C's} \\ (Q_1 \otimes Q_2) A's \text{ are } (B \text{ and } C)'s \end{array}$$

where $\otimes$ is the product in fuzzy arithmetic. An example would be: *most robbers are male* and *most male robbers are evil* implies *most*^{2} *robbers are male and evil*, where *most*^{2} ist the *most* quantifier multiplied with itself. *(ii) Chaining syllogism:*

$$\begin{array}{l} Q_1 A's \text{ are } B's \\ \underline{Q_2 B's \text{ are } C's} \\ (Q_1 \otimes Q_2) A's \text{ are } C's \end{array}$$

which is quite similar to the chaining rule in propositional logic, only with fuzzy quantifiers.

Furthermore, in [Roychowdhury and Shenoi, 1997] common sense knowledge is used as fuzzy rules to enhance databases to infer new rules and facts. Fuzzy rules are represented as tuples (X, Y), where X is the antecedent and Y the concequent. Two rules can be combined if the concequent of the first rule and the antecedent of the second rule have the same attributes. In particular three encoding techniques are described, namely: *t-conorm Encoding*, *Hypercube Encoding* and *Intermediate Implication Encoding*. Each of these techniques uses two input rules (X, Y_1) and (Y_2, Z) to create a new rule (X, Z^*), where Z^* depends on the encoding technique. An example of such an inferred rule using precise and fuzzy data can be seen in figure 3.

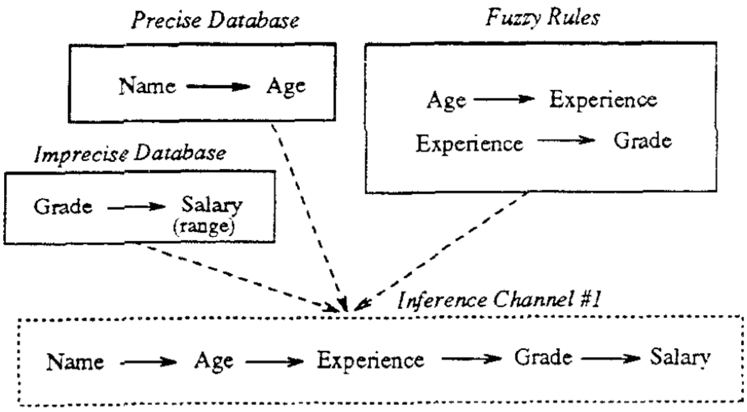

Figure 3: Here a new rule is inferred by combining precise and imprecise data with fuzzy rules. [Roychowdhury and Shenoi, 1997]

3 Usage of common sense knowledge

3.1 Common sense knowledge in artificial intelligence

In the Year 1950, Alan Turing proposed, for the first time, a test of the machine's ability to demonstrate intelligence. In his paper "Computing Machinery and Intelligence" he concerns himself with one simple but crucial question: "Can machines think?". The proposal made in the paper is known as the "Turing test". In order to find out if a machine can think like a human he proposed a simple "imitation test" involving a human interrogator that engages in a conversation with one human and one machine, with each of them trying to appear human. All participants are placed in separated locations so they can't see each other, and communication is limited to a written conversation by using natural language. The machine has passed the test if the interrogator cannot determine which one is a human and which one is a machine [Turing, 1950].

Until today there has only been several serious attempts to pass the Touring test. The first one was ELIZA, a computer program introduced by Joseph Weizenbaum in 1966. In order for his program to appear human, Weizenbaum used scripts that made ELIZA simulate a Rogerian psychotherapist. Interestingly instead of trying to tackle the problem of providing the program with a common sense knowledge data base, Weizenbaum made ELIZA only appear intelligent by rephrasing the user's statement and posing it back to him as a question [Weizenbaum, 1966]. Even the A.L.I.C.E.[8] (Artificial Linguistic Internet Computer Entity), a chatterbot which was inspired by ELIZA and developed in the late 90s by Richard Wallace, can appear intelligent by using some heuristic pattern matching and an AIML[9] knowledge base, but still fails to pass the Turing test due to the lack of a much broader common sense knowledge base.

Generally any task that requires common sense knowledge is considered to be AI-Complete. That means that the machine performing the task needs to be as intelligent as a human being i.e. it needs to be familiar with all the facts and informations that an ordinary person is supposed to know. The definition of AI-Completeness is actually a vague one if we consider the fact that many of the AI problems have not yet been formalized. This lack of formalization has motivated Dafna Shahaf and Eyal Amir to define a novel classification of the computational problems by introducing a model called Human-Assisted Turing Machine (HTM). Being that some problems are extremely difficult, or even impossible to solve by the machines, the HTM model integrates a human oracle in the computational process. The Turing Machine can utilize a human oracle to help it decide a problem at any time during the computation. This formalization includes the definition of both algorithm and problem complexity as well as reducibility in order to be able to define the complexity classes in the HTM framework. This contribution present a solid basis for the future research on complexity of AI problems[Shahaf and Amir, 2007].

In further text the application of common sense knowledge in three most common AI tasks is presented. It will be investigated to which extent can addition of such knowledge improve the results of the tasks in question.

3.1.1 Text Analysis

Text analysis has become a very important part of AI research especially in the last years. This is because due to Web 2.0 technics, data can be produced and made available on the internet, by nearly everybody. This is of course a benefit for everyone who needs knowledge about a particular topic. But the drawback is that no human can handle this amount of data and thus it has to be analyzed automatically and presented in a usable way by computers.

Here is where text analysis comes into play. Because most knowledge is represented through written text in articles, books, papers, blogs, etc. it has to be transferred into a representation which can be handled by machines.

One useful step to do this is to categorize the input text. An approach to improve text categorization using common sense knowledge was described by Majewski and Szymanski in 2008 [Majewski and Szymanski, 2008]. A common approach for categorization is to use a Vector Space Model. Here each word in the dictionary represents a dimension in the vector space. Thus a Document can be represented as a vector in that vector space by analyzing the occurring words. However, this approach doesn't consider any semantics of the text. This drawback can be eliminated by introducing relations between words [Jiang and Littman, 2000] [Wong et al., 1985].

To express such relations Majewski and Szymanski introduced a proximity matrix for the words in the dictionary. For the computation of this matrix the data of ConceptNet [Liu and Singh, 2004] was used. As described above, in ConceptNet knowledge is repre-

[8] `http://www.alicebot.org/about.html` (Nov. 2009)
[9] `http://www.alicebot.org/aiml.html` (Nov. 2009)

sented as a directed graph (as illustrated in figure 2). Thus the proximity of words was deduced using some desired characteristics which could be derived from the graph.

As the experiments had shown this approach has the potential to improve the categorization results. Although the achieved recall value cannot yet compete with that of well established methods, the precision value is very promising.

3.1.2 Object Recognition

The problem of object recognition and categorization is a classical one in the field of image processing and computer vision. The goal is to analyze a given image in order to determine if it contains a certain predefined object and to recognize and categorize the object found in the image respectively. A crucial and solid base for most of the modern object recognition algorithms has been laid by Biederman et al. in 1982. They define the five essential relational violation classes that characterize the object relations in the real-world scenes. These are Interposition (background appears to be passing trough the object), Support (an object does not appear to be resting on a surface), Probability (objects tend to be found in some contexts but not others), Position (objects position in a scene would be highly unlikely in the real world), and the familiar Size (objects have a limited set of size relations with other objects) [Biederman et al., 1982]. Rabinovich et al. argue that the classes Interposition, Support, Position and familiar Size have been mainly well utilized in the models proposed by the community, but the Probability factor hasn't been addressed enough. They propose to incorporate the semantic object context in the post-processing phase in order to reduce a possible ambiguity (Figure 4).

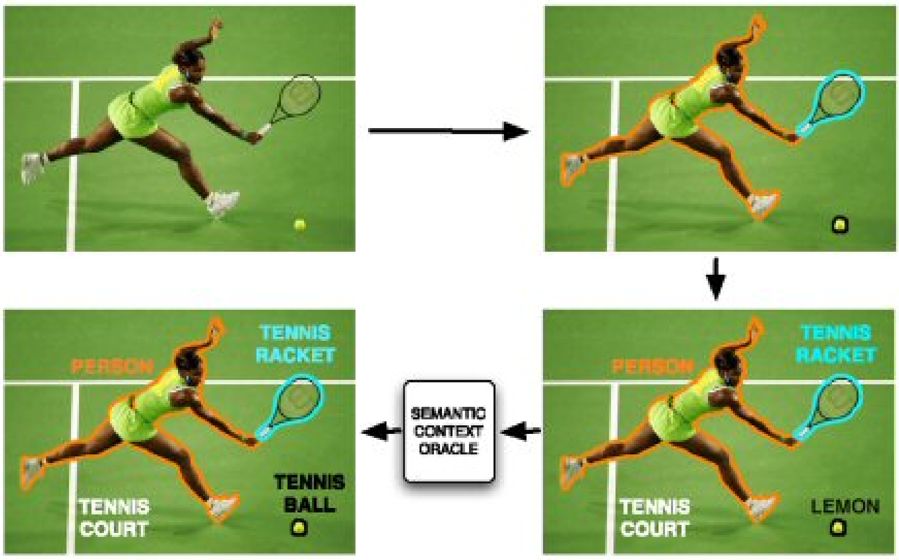

Figure 4: Idealized Context Based Object Categorization System: Image perfectly segmented into categorized objects which are labeled with respect to the global semantic context. [Rabinovich et al., 2007]

Two different sources of context information in the post-processing phase were compared. The first one was learned from the training data, and the second one was queried from the Google Sets[10] which served as a common sense knowledge base which generates

[10] http://labs.google.com/sets

a list of related items based on a few provided examples. The results have shown that by incorporating the context information into object categorization process the precision increases no matter which context information source is used [Rabinovich et al., 2007].

One further interesting example of how common sense knowledge can be used for object recognition are the so called "Games with purpose". Instead of waiting for computers to become as intelligent as humans, a group of scientists has noticed that a large number of people can willingly be integrated in the machine learning process simply by playing an online game. This way the common sense knowledge, needed for object recognition and classification, is provided by numerous sources and, as we will see, verified at the same time. There are currently two such games: the ESP Game[11] and Peekaboom[12], both developed at Carnegie Mellon University. The purpose of the ESP game is to label an image by providing the most appropriate term describing the object contained in it. The Goal is to guess what label one's random game partner would assign to the image, and as soon as both partners "agree" on a certain label a new image is provided and so on (Figure 5).

Figure 5: The ESP Game interface: The goal it to "agree" with one's partner on as many images as one can in 2.5 minutes. The thermometer-shaped indicator shows on how many images the partners have already agreed on.[von Ahn and Dabbish, 2004]

Only a few month after its deployment, the ESP game collected more than 10 million image labels, and Luis von Ahn claims that if the game would be hosted on on the major sites such as Yahoo, it would be possible to label all the images in the web in a very short period[Ahn, 2006].

The other online game - Peekaboom goes even further, and tries to locate the objects in the images. It is considered as an extension to the ESP Game, because ESP Game can only provide labels for the objects contained in the image, but it can not determine their position, and by that, the object segmentation cannot be successfully achieved. This step is very important for training of the computer vision algorithms. It is important to mention that every image-label mapping used in the Peekaboom originates from the

[11] http://www.espgame.org/
[12] http://www.peekaboom.org/

ESP Game. The Peekaboom game is played by two randomly paired players, one of them assigned with the role "Peek", and the other one with the role "Boom". Peek can only see a blank screen in the beginning, while Boom is presented with an image and a corresponding label(Figure 6).

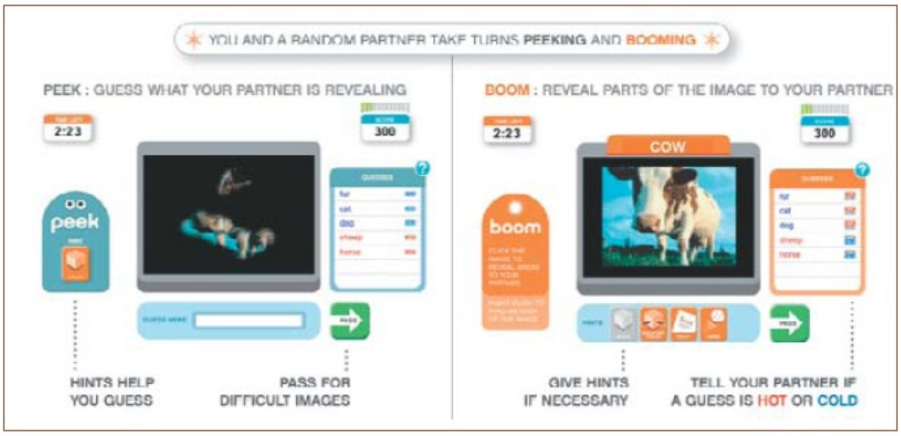

Figure 6: The Peekaboom Game interface: Peek tries to guess the associated label as Boom slowly reveals the image.[Ahn, 2006]

Peek's goal is to guess the label associated with the image, as Boom reveals a 20 pixel big circular area with each turn. The more precise a Boom is when revealing the image areas, the more points he can achieve. By means of this game the system can obtain a very precise location information for each object in a given image[Ahn, 2006].

3.2 Application for information retrieval

Due to the scope of common sense knowledge there are numerous possibilities for its utilization in the information retrieval process. For instance, in information retrieval, relevant documents are rarely identical as the search query, so one of the most common difficulties is caused by lexical and syntactic ambiguity. This means that in the most simple approach, the information retrieval is reduced to matching a set of query terms with a set of index terms, and even if some document contains the relevant information it will not be found if it contains merely the synonyms for the query terms and not the exact words. In order to tackle this problem Smeaton used a custom partitioned WordNet network and a semantic similarity estimator in order to calculate the word-word semantic distances. One problem in using the word-word semantic distance was the fact that WordNet keeps each sense of a polysemous word(a word with multiple meanings) as a separate entry, so an additional disambiguation process that chooses the most likely sense of a given noun had to be integrated in the information retrieval process. This has proven to be very critical because the obtained precision and recall results were very poor compared to term frequency - inverse document frequency term weighting strategy. Nevertheless, one further image retrieval experiment, also using word-word semantic distance, but manually disambiguated image labels, has proven to be much more efficient[Smeaton, 1999].

Previously mentioned image labeling process can also prove itself to be very useful in the image retrieval process, assuming that the image labels precisely correspond to the objects in the image of course. While resources like WordNet have been utilized to neutralize the divergence in vocabulary between queries and documents, the ConceptNet can be used to retrieve labeled images by focusing on the spatial relationships between concepts[Hsu and Chen, 2006]. The ConceptNet repository, which contains a large set of common sense concepts, assists the information retrieval system thereby by providing the concepts that usually coexist with the concepts found in the search parameter. In order to do that, ConceptNet provides the necessary natural language processing tools needed for reasoning over text. The image labels, which contain a textual information about the image content, are treated analogue to documents in the document retrieval process. Hsu and Chen describe the commonsense concept expansion as a four step process:

1. Part-of-speach tagging and lemmatizing the image labels.
2. Choosing the concepts in the image labels that will be expanded. These are called the *expanded concepts*.
3. Finding the concepts that are related to the expanded-concept by using the spatial relationship provided by ConceptNet. These related concepts are called the *projection-concepts*.
4. Part-of-speach tagging of the expanded-concept and filtering of the concepts which are not nouns.

The projection-concepts are indexed separately from the original labels, and during the information retrieval process both collections are searched and the result is linearly combined. The experiments have shown that the concept expansion approach is more suited for precision-oriented tasks and topics that are difficult for classical information retrieval systems [Hsu and Chen, 2006].

4 Lessons Learned

As we have seen the idea to equip computers with common sense knowledge has already existed for decades. For instance the knowledge base of Cyc is being developed since the 1980s. Nevertheless Cyc, as any other computer program, is far from behaving intelligently, compared to a human.

As we found, the big problem here is, that it is extremely hard to express or even to define the meaning of a concept. The shown approaches of knowledge collection enabled engineers to feed programs with millions of facts. Furthermore with using reasoning methods, several tasks can be performed and have applications in certain scenarios. But the data is still a graph with some labeled nodes and edges, and the reasoning methods are some mechanical processes. Thus no new knowledge can be found, that isn't existing implicitly in the already learned facts.

One way in the right direction seems to be, to use the immense amount of data which is available on the internet, and provide it to a system as an input channel. With that ability, the system has some kind of connection to the outside world, like humans have with their senses. Thus it may be possible to get programs, to state hypotheses and revise them empirically and maybe changing the hypotheses according to the found data, just like human scientists do.

However, the situation here is quite the same as generally in AI. There exist some applications where the methods work very good, but only if the task can be exactly defined and restricted to a narrow domain.

5 Conclusion

From the very beginning, computers have been used to do the work faster that a human beeing could ever do. They also can perform actions that are impossible for us. However there are some areas where computers can not compete with humans, such as text understanding, translation, object recognition etc. We have shown that if we want computers to perform better in such aspects, it can help to give them some understanding like that of humans. In other words, we have to give them some commonsense. In order to enter that knowledge into computers, first we have to collect it and convert it to a representation understandable by computers. We described three different approaches for commonsese knowledge collection, namely the human-based approach, semiautomatic and automatic knowledge extraction, each with its advantages and disadvantages. Furthermore, some typical applications of commonsense knowledge in computational technologies were presented, and some examples how commonsense can improve the results of several tasks were given.

References

[Ahn, 2006] Ahn, L. v. (2006). Games with a purpose. *Computer*, 39(6):92–94.

[Arnold, 2009] Arnold, K. (2009). ConceptNet. http://conceptnet.media.mit.edu/.

[Biederman et al., 1982] Biederman, I., Mezzanotte, R. J., and Rabinowitz, J. C. (1982). Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143–177.

[Cankaya and Moldovan, 2009] Cankaya, H. C. and Moldovan, D. (2009). Method for extracting commonsense knowledge. In *K-CAP '09: Proceedings of the fifth international conference on Knowledge capture*, pages 57–64, New York, NY, USA. ACM.

[Elliman and Pulido, 2002] Elliman, D. and Pulido, J. (2002). Visualizing ontology components through self-organizing maps. In Williams, D., editor, *6th International Conference on Information Visualization (IV02)*, page 434–438. IEEE Computer Soc.Press, Los Alamitos.

[Elliman and Pulido, 2003] Elliman, D. and Pulido, J. (2003). Self-organizing maps for detecting ontology components. In Arabnia, H., editor, *The 2003 Int.Conf.on Artificial Intelligence (IC-AI)*, page 434–438. CSREA Press.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M. J., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Web-scale information extraction in knowitall: (preliminary results). In Feldman, S. I., Uretsky, M., Najork, M., and Wills, C. E., editors, *WWW*, pages 100–110. ACM.

[Feigenbaum, 1990] Feigenbaum, E. (1990). Knowledge processing–from file servers to knowledge servers. In *The Age of Intelligent Machines*. MIT Press.

[Fellbaum, 1998] Fellbaum (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

[Gruber, 1995] Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. volume 43, pages 907–928.

[Hsu and Chen, 2006] Hsu, M.-H. and Chen, H.-H. (2006). Information retrieval with commonsense knowledge. In Efthimiadis, E. N., Dumais, S. T., Hawking, D., and Järvelin, K., editors, *SIGIR*, pages 651–652. ACM.

[Jiang and Littman, 2000] Jiang, F. and Littman, M. L. (2000). Approximate dimension equalization in vector-based information retrieval. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 423–430. Morgan Kaufmann.

[Liu and Singh, 2004] Liu, H. and Singh, P. (2004). Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226.

[Majewski and Szymanski, 2008] Majewski, P. and Szymanski, J. (2008). Text categorization with semantic commonsense knowledge: First results. In *Neural Information Processing*, pages 769–778. Springer Berlin / Heidelberg.

[Pulido et al., 2007] Pulido, Flores, Reyes, Diaz, and Castillo (2007). In the quest of specific-domain ontology components for the semantic web. Neuroinformatics Group, Bielefeld Germany.

[Rabinovich et al., 2007] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *ICCV*, pages 1–8. IEEE.

[Roychowdhury and Shenoi, 1997] Roychowdhury, S. and Shenoi, S. (1997). Fuzzy rule encoding techniques.

[Sarjant et al., 2009] Sarjant, S., Legg, C., Robinson, M., and Medelyan, O. (2009). "all you can eat" ontology-building: Feeding wikipedia to cyc. In *Web Intelligence*, pages 341–348. IEEE.

[Shahaf and Amir, 2007] Shahaf, D. and Amir, E. (2007). Towards a theory of ai completeness. 8th International Symposium on Logical Formalizations of Commonsense Reasoning.

[Smeaton, 1999] Smeaton, A. F. (1999). Using nlp or nlp resources for information retrieval tasks. In *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Publishers.

[Storey et al., 2005] Storey, V. C., Sugumaran, V., and Ding, Y. (2005). A semi-automatic approach to extracting common sense knowledge from knowledge sources. In Montoyo, A., Muñoz, R., and Métais, E., editors, *NLDB*, volume 3513 of *Lecture Notes in Computer Science*, pages 322–332. Springer.

[Suh et al., 2006] Suh, S., Halpin, H., and Klein, E. (2006). Extracting common sense knowledge from wikipedia.

[Turing, 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, LIX:433–460.

[von Ahn and Dabbish, 2004] von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In Dykstra-Erickson, E. and Tscheligi, M., editors, *CHI*, pages 319–326. ACM.

[Weizenbaum, 1966] Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

[Winston, 1980] Winston, P. H. (1980). Learning and reasoning by analogy. *Commun. ACM*, 23(12):689–703.

[Wong et al., 1985] Wong, S. K. M., Ziarko, W., and Wong, P. C. N. (1985). Generalized vector spaces model in information retrieval. In *SIGIR '85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25, New York, NY, USA. ACM.

[Zadeh, 1986] Zadeh, L. A. (1986). Commonsense reasoning based on fuzzy logic. In *WSC '86: Proceedings of the 18th conference on Winter simulation*, pages 445–447, New York, NY, USA. ACM.