University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Knowledge-Based Systems and Knowledge Engineering

Michel Schlatter

December 25, 2018

Prof. Dr. Knut Hinkelmann
Devid Montecchiari

ToBIT 2018

Abstract

This paper provides an introduction into knowledge engineering and knowledge-based systems (KBS). It lists the advantages and disadvantages of engineering and using a KBS and describes when it makes sense to invest in the development of a KBS. In addition, it shows in which application areas KBS are typically used and provides a rough description about the modules of a KBS. The paper also explains the difference between symbolic and subsymbolic systems and illustrates how these systems can be combined.

According to Akerkar and Sajja (2010), KBS belong to the field of artificial intelligence. It is a tool that offers collective knowledge of one or more experts as well as from books and other sources. As a result, a knowledge-based system can act as an expert on demand (p.18). A knowledge engineer (KE) acquires knowledge, builds and maintains the KBS (Hinkelmann, 2010, p.11). As stated by Wu (2014, p.3), another way to acquire knowledge is through machine learning.

Statement of Authenticity

I confirm that this paper research was performed autonomously by myself using only the sources, aids and assistance stated in the report, and that quotes are readily identifiable as such.

24.12.2018, Michel Schlatter

Table of Contents

1. Introduction

According to Hartmann (2002), the economy is changing from the old economy to the new economy. In the new economy, knowledge is a fundamental resource. In contrast to the old economy, which consumes its resources (labour, capital and environment) in its operations, the new economy constantly generates new knowledge through the use of existing knowledge. This ensures sustainable economic growth (p.15). Unfortunately, much of this knowledge is tied to experts. Experts are rare and it takes years for a person to become an expert in a specific field. In addition, an expert is only a human being with needs such as food, sleep and social contacts. If he cannot satisfy these needs, his work gets worse. In summary, the expert becomes even scarcer due to the limited time in which he can be productive and his knowledge is lost when he dies.

How could the problem of rare experts and the loss of knowledge be solved? One solution would be KBS, which were invented around 1970. According to Akerkar and Sajja (2010), expert knowledge can be made explicitly available by a KE via a KBS (p.60). A KBS can support non-experts in taking on some of the expert's work (p.18). To make this possible, the knowledge of the expert must be available at any time and in a constant quality.

1.1 Problem Statement and Research Goal

This paper is intended to provide a structured and clear overview about knowledge-based systems and knowledge engineering so that the reader has enough knowledge to judge whether such a system is useful for an application or not. After reading this paper the reader should be able to understand:

- The different types of knowledge
- What a knowledge-based system is
- What the function of a knowledge engineer is
- What the goals of a KBS are
- What the different modules of a KBS are and what they roughly do
- What an expert system is
- The difference between symbolic and subsymbolic KBS
- When it is appropriate to implement a KBS
- Fields of application of a KBS

2. Introduction into Knowledge Management

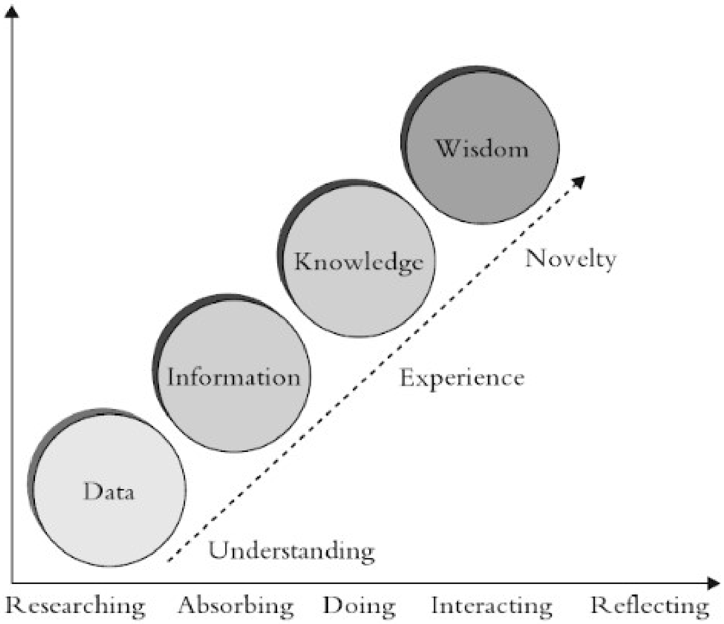


Figure 1: Knowledge process (Akerkar, Sajja, 2010, p.11)

Figure 1 shows that knowledge is created by a process. The following paragraph is inspired by the explanations given by Akerkar and Sajja (2010, p.13ff):

First there is data. Data is raw and without a meaning like for example the number 13061996. When this data is later put into context (a meaning is added), information is generated. If we have the information that the number 13061996 is a birthday, we have the information that a subject's birthday is on the 13.06.1996. After containing this information, a receiver can add his experience and connect other information to create knowledge. He has also the information from which person this data originates for example. He can link this two information and knows therefore whose birthday is on the 13.06.1996. The next step is wisdom. For this you must have a deep understanding of knowledge and be able to derive it. A person who has an understanding of the date 13.06.1996 can derive that this person will be 22 years old in 2018 (p.13ff).

Akerkar and Sajja define knowledge as follows:

> *"Knowledge is considered a human understanding of a subject matter that has been acquired through proper study and experience" (Akerkar, Sajja, 2010, p.15).*

2.1 Knowledge Management

Based on Bendel (2017), knowledge management is a specific form of dealing with knowledge, information and data. Knowledge management serves to create structures in organisations that enable the generation, spreading, preservation and utilisation of information and knowledge. Especially important is to make implicit knowledge explicit (p.16).

According to Reinmann (2009), the reasons for knowledge management are the reactions to the knowledge society, the knowledge economy and the knowledge work (p.6).

2.2 Implicit Knowledge and Tacit Knowledge

We differentiate between tacit and implicit knowledge. Both of them are internal knowledge. That means the knowledge is stored in the heads of people but with one important difference:

Tacit Knowledge

> *"Tacit knowledge is based on experiences, memories and convictions, or is shaped by personal value systems. It eludes formal expression and is difficult to communicate" (own translation) (Bendel, 2017, p.13).*

A good example of tacit knowledge is cycling. Many people know how to ride a bike, but find it incredibly hard or even impossible to describe.

Implicit / Self-Aware Knowledge

Implicit knowledge can be communicated or written down quite easily. An example would be a mother explaining her child how to use the washing machine. In this case the mother makes her implicit knowledge explicitly available.

Tacit knowledge is difficult / impossible to communicate, while implicit knowledge can be quite easily made explicitly available.

2.3 Explicit Knowledge

> *“Explicit knowledge is formally describable or articulatable knowledge - it can be expressed e.g. in words or numbers and printed in books” (own translation) (Bendel,2017, p.13).*

A user manual is a good example of explicit knowledge. This knowledge (e.g. about operating a machine) is formally describable and can be expressed in words.

According to Hinkelmann (2018), even explicit knowledge can be further distinguished. There is documented and formal knowledge. While documented knowledge is found in documents and databases, formal knowledge is described in knowledge bases or program code (p.2).

2.4 Individual Knowledge and Collective Knowledge

Based on Bendel (2017), knowledge is first tied to individuals (unless it is generated by a machine). It arises from the intellectual achievements of individuals. This knowledge is individual/ private knowledge. When this people share their explicit and implicit knowledge, then collective knowledge emerges. This knowledge can be used by several people at the same time (p.14).

This is the case when an employee writes his knowledge in the company’s internal wiki and publishes it after.

2.5 Codification

According to Bendel (2017), the codification strategy tries to extract the knowledge of the employees. This knowledge is then stored in systems so that other employees can query the knowledge without having to query the original knowledge carrier. This is also called the "person-to-document" approach (p.22).

3. Knowledge-Based Systems

> *"Building a KBS means building a computer model with the aim of realizing problem-solving capabilities comparable to a domain expert"(Studer, Benjamins, Fensel, 1998, P.3).*

According to Akerkar and Sajja (2010), a knowledge-based system is a computer program that offers collective knowledge of one or more experts to non-experts. KBS are a major branch of artificial intelligence. It uses and generates knowledge from data, information and knowledge. Since our society is becoming a knowledge-society, we rely on different experts' decisions and their knowledge, which a KBS tries to represent as good as possible. A knowledge-based system can understand the information which is being processed and therefore can make decisions. A KBS can be used at anytime and anywhere (p18).

According to Hinkelmann, the terms knowledge-based systems and expert systems are often used synonymously (2010, p.6). However, the term knowledge-based system is to understand as an umbrella term and an expert system as a specific subtype of a KBS.

3.1 Goals of a KBS

According to Akerkar and Sajja (2010), a KBS is a fifth-generation computer technology, with the following objectives:

- Assists people in discovering and developing unknown fields
- Offers a vast amount of knowledge in different areas, because a KBS is a collection of knowledge from different expert knowledge, documents, etc.
- Aids in the management of knowledge (p. 19)

The following paragraph is based on Akerkar and Sajja (2010, p.18ff):

Summarized, the main goal of a KBS is to make knowledge (from experts, printed media, internet, etc.) from a domain explicitly available to non-experts. This can facilitate the work load on experts and save costs. In addition, a KBS serves to preserve knowledge and provides it always in a constant quality. It is also interesting that a KBS can contain the knowledge of several experts, which can lead to an increase in quality: Even expert opinions can be verified with the help of the KBS.

3.2 Architecture

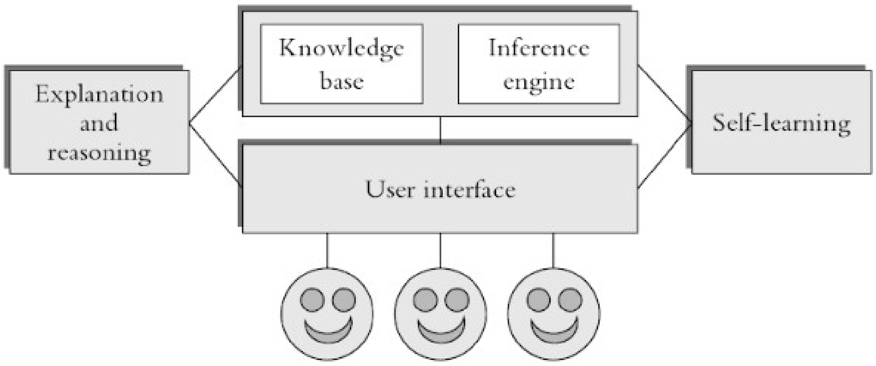


Figure 2: General structure of a KBS (Akerkar, Sajja, 2010, p.20)

Figure 2 shows the general structure of a KBS. The individual components are explained in detail in the following.

3.2.1 GUI

The graphical user interface (GUI) is what the user sees on the screen and with which he interacts. According to Tripathi (2011), the purpose of the GUI is to share the internal knowledge of the KBS in a user friendly (for the user understandable) way. The user can enter a question and gets the answer to it from the KBS. He can also request the explanation how the system came to its conclusion (p.2). However, the system is also capable of asking the user questions when it needs more information to arrive at a result.

3.2.2 Inference Engine

Based on Tripathi (2011), an inference engine (IE) is a software program that uses and manipulates the existing knowledge from the knowledge base. It uses the rule interpreter to analyse and process the rules and provides methodology for reasoning (p.2). Akerkar and Sajja (2010) additionally mention that the IE adds new facts to the knowledge base by applying certain rules (modus ponens and modus tollens) when possible (p.36).

The following paragraphs are based on Akerkar and Sajja (2010):

The aim of the inference engine is to apply the facts and rules of the knowledge base to the information entered by the user in order to reach a conclusion. For example, a medical diagnosis system has the following rules in its knowledge base:

R1 : if (fever and breathlessness) then diagnose (pneumonia) exit
R2: if (temp > 38) then assert (fever)

A user now enters via the user interface that he has a temperature of 39 degrees and is breathless. The inference engine will first apply the second rule and add fever to the symptom list. After that the first rule will be executed and the IE will conclude that the patient is suffering from pneumonia (p.40ff).

3.2.3 Knowledge Base

> *"The knowledge base is the key component of a knowledge-based system. The quality and usefulness of the system is directly related to the knowledge represented in it" (Akerkar & Sajja, 2010, p. 35).*

The sentence above reveals the importance of the knowledge-base for the knowledge-based system. The knowledge-base contains the knowledge for solving problems of a specific domain. Figure 3 shows the components of the knowledge-base.

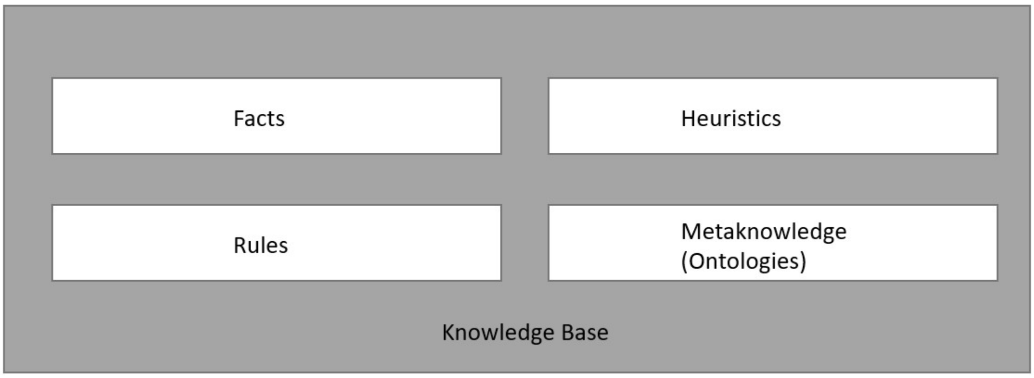


Figure 3: Components of a knowledge-base (Akerkar, Sajja, 2010, P.36) (modified)

Facts

A fact is a statement about reality (Hinkelmann, 2010, p.18). This means a fact is proved or known to be true. Examples of facts would be:

- All animals breath oxygen
- Humans are mortal

Factual knowledge is in most cases widely shared, for example in textbooks or journals (Tripathi, 2011, p.2).

Rules

Examples of rules are:

- If object is animal then needs oxygen to live
- If object is human then is mortal

We can describe rules as ‘IF […] THEN […]’ statements.

Heuristics

As stated by Gigerenzer (1999), heuristic is a method to reach practicable solutions or statements with high probability with limited knowledge and time. Tripathi (2011, p.2) describes it as the knowledge of good practice, good judgement, and plausible reasoning in a field.

According to Akertar and Sajja (2010), heuristics are, in contrast to rules and facts, generally stored in experts minds in form of tacit knowledge. Therefore, heuristics are difficult to extract for the knowledge base (p.34).

An example of heuristic is as follows:

“If there is total eclipse of the Sun, there is no daylight, even though the Sun is in the sky” (Akerkar & Sajja, 2010, p.34).

Metaknowledge

Metaknowledge is knowledge about knowledge. In the case of a KBS it generally includes knowledge about ontologies, target application and methods of use (Akerkar & Sajja, 2010, p.36).

3.2.4 Explanation Component

According to Tripathi (2011), the explanation component justifies the conclusion of the KBS. That means the KBS provides the user with an explanation of the reasoning process that leads to the final conclusion (p.2). Sasikumar, Ramani, Raman, Anjaneyulu, and Chandrasekar (2007, p.103) add, that if the KBS asks the user a question, it should be able to explain why it needs the answer.

This component is used by the user to understand how the system came to its conclusion. He can learn and understand the individual steps of the systems argumentation. If the system comes to a strange conclusion, the user can check where the error happens and possibly change the input or report a bug whereupon the system is improved.

3.2.5 Knowledge Acquisition Component

With the progress in different areas, also the knowledge changes. After the knowledge engineer has finished developing the KBS, the system has to be maintained and fed with new knowledge, respectively outdated knowledge must be changed.

The knowledge of a KBS can be updated in two different ways:

Self-Learning

According to Akerkar and Sajja (2010), a KBS can acquire new knowledge independently. For example by learning from its own cases or deriving knowledge from existing data sets. This self-learning is known as automatic learning or machine learning (p.67).

Human Driven

Based on Akerkar and Sajja (2010), the knowledge can either be updated by the knowledge engineer who has a constant eye on the trend and advancements of the specific domain or by the expert himself. To allow an expert to update the knowledge by himself, the KBS is developed in conjunction with a general editor or knowledge acquisition tool. The editor asks the expert interactive questions. The necessary knowledge is then extracted from the answers and stored in the knowledge base (p. 68). The general editor and the knowledge acquisition tool are both subsystems that help experts to update the knowledge base (Tripathi, 2011, P.2).

The knowledge acquisition process is covered in more detail in the chapter knowledge engineering.

4. Knowledge Engineering

"Knowledge Engineering is an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise" (Feigenbaum, 1983, as cited in Hinkelmann, 2010, p.11).

According to Sasikumar et al. (2007), a knowledge engineer acquires information about the domain from experts and other sources and represents it in the knowledge base. This knowledge can then be used by the KBS. The knowledge engineer tests the system with the expert and improves it afterwards. After the KBS has been put into operation, he maintains it by e.g. adding new knowledge (p.137).

Akerkar and Sajja (2010) claim that the quality of the KBS depends strongly on the skills of the knowledge engineer. His knowledge will be reflected in the KBS, not the knowledge of the expert (p.24). Therefore an experienced, reliable knowledge engineer is essential to develop a high quality KBS.

4.1 Knowledge Acquisition

"Knowledge acquisition is the accumulation, transfer and transformation of problem-solving expertise from experts and/or documented knowledge sources to a computer program for constructing or expanding the knowledge base" (Tripathi, 2011, p.2).

Based on Akerkar and Sajja (2010, p.24), there is no predefined procedure that guarantees knowledge acquisition and representation. However, the following sections should give an impression of the topic by identifying the sources of knowledge, the steps, the difficulties and the techniques for acquiring knowledge.

4.1.1 Knowledge Sources

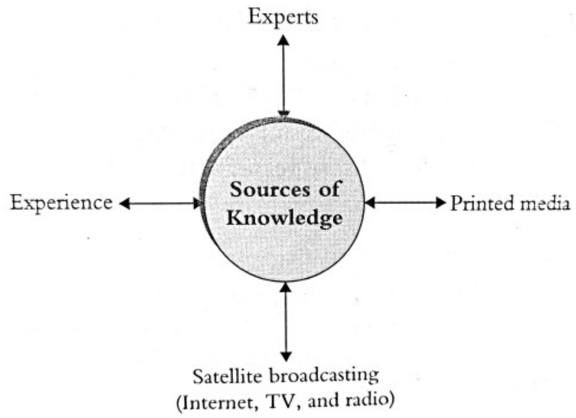


Figure 4: Sources of knowledge (Akerkar, Sajja, 2010, P. 29)

In Figure 4 you can see the knowledge sources for a KBS. A knowledge engineer tries to acquire this knowledge and represents it in the knowledge base.

4.1.2 Steps

Akerkar and Sajja (2010, p.61) describe the following steps for knowledge acquisition:

Find Suitable Experts and a Knowledge Engineer

The effectiveness of a KBS is directly related to its knowledge. As a result, it is important to find suitable experts (and convince them to share their knowledge) and reliable knowledge engineers.

Proper Homework and Planning

Before interviewing the expert, the knowledge engineer must first acquire all available knowledge of the domain in order to be optimally prepared for the acquisition process.

Interpreting and Understanding the Knowledge Provided by the Experts

The KBS represents the knowledge of the engineer and not the knowledge of the expert. As a result, it is essential that the knowledge engineer fully understands the knowledge provided by the expert before representing it in the knowledge base.

Representing the Knowledge Provided by the Experts

The understood and acquired knowledge should be immediately recorded to the knowledge base.

4.1.3 Techniques

The table below describes the techniques, the purpose of the techniques, the involved sources and which knowledge types can be extracted. The techniques, purposes and types of knowledge are described by Akerkar and Sajja (2010, p.62):

Technique	Purpose	Sources of knowledge	Types of Knowledge
Reviewing the literature	Because an expert's time is precious, the KE should only ask the expert about the knowledge not found in literature.	Printed media, internet, etc.	Explicit knowledge
Interviews	Interviews are appropriate for questions about procedural knowledge ("know-how").	Experts and other competent people in the domain field	Explicit knowledge
Surveys and Questionnaires	Surveys and Questionnaires are typically used for explicit knowledge extraction.	Experts and other competent people in the domain field	Explicit knowledge

Observation	This method may help to get a better insight into the solution strategy and helps to clear up any confusion.	Experts and other competent people in the domain field	Tacit knowledge
KBS prototype	A KBS prototype, which can be used by the experts, could help to identify improvement potentials.	Knowledge of the KBS, acquired from different sources.	Implicit knowledge

Table 1: Techniques for knowledge acquisition

4.2 Difficulties and Machine Learning

As stated by Feigenbaum (1983), knowledge is the key factor in the performance of a KBS and there exist two types of it:

- The facts of the domain (written in textbooks and journals of the field, etc.)
- The heuristic knowledge (knowledge of good practice and good judgment in a field) (p.76)

According to Feigenbaum (1983, p.76ff), the heuristic knowledge is the hardest to get, because experts do often not have the self-awareness to recognize that it is heuristic knowledge and therefore can be difficult to describe.

Akerkar and Sajja (2010) describe the same problem but in different words: The implicit and tacit knowledge of an expert is difficult to extract with typical fact-finding techniques (p. 49). Akerkar (2014, p.10) states, that also the representation of knowledge in the knowledge base is difficult because the acquired knowledge from the expert is imprecise and uncertain.

Ultsch and Korus (1995) state, that with the help of machine learning (in this case with a neural network) the above-mentioned difficulties can be mostly erased. It is even possible to learn knowledge from experience directly (p.1). This means, that also tacit knowledge can be acquired by machine learning algorithms. Another advantage is, that the knowledge does not have to be updated manually (Akerkar & Sajja, 2010, p.20). Also the representation of the knowledge is done automatically by the machine learning algorithm. However, machine learning algorithms also have risks, challenges and limitations (Pineda, 2017).

Please note that this paper only deals with machine learning in combination with neural networks. Depending on model type and machine learning algorithm there are different advantages and disadvantages.

For further information about neural networks and machine learning, please refer to chapter 6.2.

5. Advantages and Disadvantages of KBS and KE

The advantages and disadvantages of KBS and KE, based on Akerkar and Sajja (2010) are listed below:

Advantages:

- **Permanent documentation of knowledge**
    - The knowledge from multiple experts, documents, and other sources is permanently stored in the knowledge base (p.46).
    - The knowledge of individual experts became collective knowledge through codification. Several people can now access this knowledge at the same time without asking the original knowledge carrier.
- **Cheaper solution and easy availability of knowledge**
    - The development of a knowledge base is costly, but it is a one-time cost. After that, the knowledge base or the whole knowledge-based system can be cloned multiple times. KBS save cost, time and effort (p.46).
    - In my opinion, Akerkar and Sajja are a little too optimistic about this. If the knowledge base is cloned, the clones still have to be maintained. Cloning could lead to some knowledge bases not being up to date. A centralized knowledge base would surely be a better solution. This knowledge base can still be used for different purposes, but it only needs to be maintained once.
    - A KBS is always available (in contrast to the expert).
- **Dual advantages of effectiveness and efficiency**
    - Since a KBS is a computer system, and good programmed computer systems are mostly efficient, the KBS is already very fast and precise (p.47).
    - Because the KBS has a knowledge component, it could become just as effective as human experts (p.47).
    - With increasing the level and amount of knowledge in the knowledge base, the decisions of a KBS will be more reliable (p.47).
    - A KBS can contain the knowledge of several experts. This can also lead to an increase in effectiveness.

- **Justification for better understanding**
  - A KBS is able to give detailed explanations to the user to justify the decision made (p.47).
- **Self-Learning and ease of updates**
  - The knowledge-base can be updated through machine learning, manually by a knowledge engineer or via special tools (e.g. expert system shell) even by a domain expert (p.47).
  - The inference engine usually does not need to be changed (p.47).

Disadvantages:

- **High cost and effort**
  - Developing a KBS takes much time from experts, developers, knowledge engineers and testers (p.56).
  - The time of these specialists is rare and therefore very expensive.
- **Dealing with experts**
  - Since experts are rare, acquiring knowledge for the KBS can be difficult (p.56).
- **Creativity and innovation**
  - Human beings can react creatively to unusual situations while a KBS usually has trouble with it (p.48).
  - It is difficult to exactly mimic and generate humanlike reasoning and thought processes in a model of natural intelligence (p.24).
- **Partial self-learning**
  - A KBS can only learn very limitedly like a human. (p.48):
- **Knowledge acquisition**
  - It is difficult to extract the implicit and tacit knowledge of an expert with typical fact-finding techniques. (p.49)
  - The quality of the KBS depends strongly on the skills of the knowledge engineer. His knowledge will be reflected in the KBS, not the knowledge of the expert. (p.61):
  - The (symbolic) knowledge base will as a result of the tacit-knowledge-extraction-problem never be fully complete and therefore the KBS never be as competent as an expert.

- **Characteristics of knowledge**
  - Akerkar (2005) states that, even to solve a simple task, an extensive amount of knowledge is required. In addition, knowledge is constantly changing and must be updated. As a result, the development of a KBS is more difficult.
- **The high level of risk**
  - High development cost
  - Difficult to acquire the expert knowledge
  - Updating the knowledge (more costs)
  - Heuristics give the power to the KBS, but heuristics do not give a guarantee of a solution (p.56ff).

5.1 When to Invest in Creating a KBS

In these cases (or combinations of them) it makes sense to develop a KBS:

- When you work in an important, high-value problem area (Sasikumar et al., 2007, p.14)
- Where is a need for making knowledge explicitly available to the collective
- Where is a need for storing the knowledge permanently (in a formal way) (Akerkar & Sajja, 2010, p.21)
- When (expert) knowledge should be available at any time in a constant quality
- When there is a high need for efficiency and speed (Sasikumar et al., 2007, p.14)
- When there is no discussion about what constitutes a good solution (Sasikumar et al., 2007, p.14)
- Where no or very little tacit knowledge is important in the field
- Where the knowledge in the field has a medium to long lifetime
- Where creativity and innovations are not so important (mostly fixed patterns can be applied)
- When it can be accepted that good effectiveness is only possible with a large amount of knowledge in the knowledge base.

An example of Sasikumar et al. (2007):

Consider working in a hospital predicting side effects of drug combinations. The hospital gives thousands of prescriptions per day and only a small number of experts are available. The patient dossiers are already available in digital form with information about regular medication, allergies, etc. (p.14). This would be an ideal application for a KBS because training humans to do this job is very time consuming and expensive. A KBS would also be much faster and more reliable than an expert.

I claim that a KBS is very helpful in a learning process. A medical student or assistant doctor can use the KBS to review their decisions and learn from the system when no expert (e.g. the senior/ chief physician) is available. This can give the learner more confidence in his decisions. With the help of the explanation component, the learner could even see the logical intermediate steps of the system that led to the conclusion.

6. Types of KBS and Their Differences

We differentiate two main groups of KBS. The symbolic and the subsymbolic knowledge-based system, which are explained in detail in the following chapters.

6.1 Symbolic Systems

According to Oxford University Press (n.d.), a symbol is a mark or character used as a conventional representation of an object. Examples of symbols stored in the knowledge base are: numbers, letters, words, punctuation, and even complete sentences (Crankshaft Publishing, n.d).

Based on Kvasnicka (2007, p.2), symbolic systems have a symbolic processor, which accept symbolic input information and creates symbolic output information.

According to Akerkar and Sajja (2010, p.190), classical artificial intelligence deals with the symbolic representation of knowledge. This means that the knowledge is represented in a database in a symbolic way. Such a knowledge base consists of ontologies, rules, facts and heuristics. A symbolic KBS can explain and reason its steps towards the final conclusion.

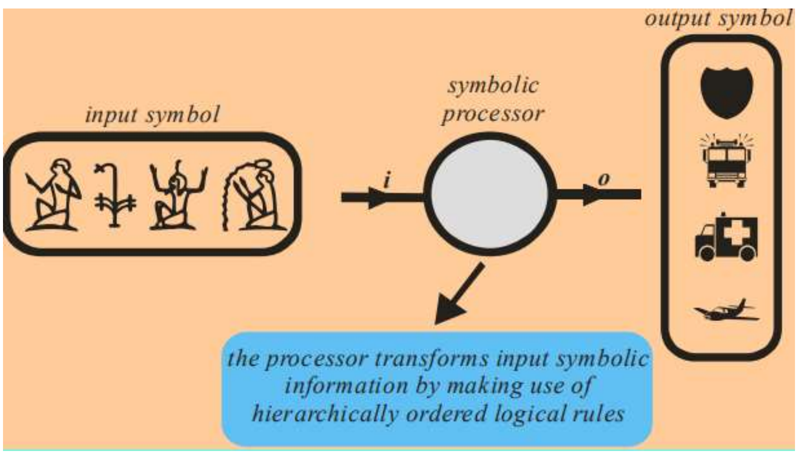


Figure 5: Symbolic system (Kvasnicka, 2007, P.3)

Figure 5 illustrates the process flow of a symbolic system.

Based on Tuthill and Levy (1991), there are five different types of (symbolic) KBS. We will take a closer look at two of them:

- Expert systems
- Case-based systems

Expert System

The expert system (ES) is the pioneer and the most popular of the knowledge-based systems (Akerkar & Sajja, 2010, p.21). Based on Haocheng (2017), an expert system emulates the decision-making ability of a human expert by reasoning knowledge. ES are an important branch of artificial intelligence (p.2).

Alberico and Micco state that, by definition, an expert system cannot be built without at least one expert (1990, p.22).

According to Akerkar and Sajja (2010, p.40ff), a rule-based ES stores rules and facts in its knowledge base acquired by a knowledge engineer from experts, books, and other sources in the domain. These rules and facts are executed by the inference machine to reach a conclusion.

A short example of Alberico and Micco (1990, p.32ff) shows how these rules and facts are applied in an expert system:

Fact pattern:
born-in(<person>, <country>)
Instantiated Fact
born-in(Susanne, Switzerland)
Rule pattern
IF (born-in(<person>, <country>)) THEN <person> is citizen of <country>
Instantiated Rule
IF (born-in(Susanne, Switzerland)) THEN Susanne is citizen of Switzerland

Case-Based System

According to Hüllermeier (2007, p.13ff), a case-based system tries to find already existing, comparable cases in its case-base for a newly entered case and transfers the solution from the existing case to the newly entered case. Akerkar and Sajja (2010, p.110) claim that researchers have shown that case-based reasoning is the more efficient approach to develop a KBS than with the rule-based approach.

6.2 Subsymbolic KBS

According to Kvasnicka (2007), subsymbolic (connectionist) systems imitate the functionality of the human brain. The functionality of artificial neural networks is inspired by theoretical concepts of neuroscience. The information entered is parallelly processed by simple calculations realized by neurons over one or multiple layers, which leads to a result (output) (p.6).

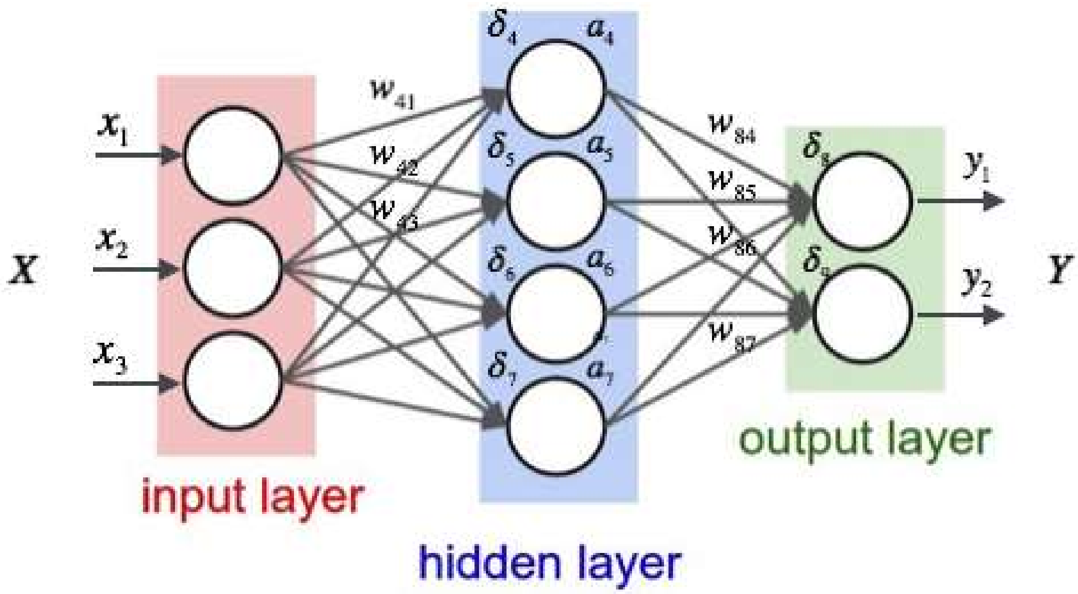


Figure 6: Neural network (Valkov, 2017)

Figure 6 visualizes a feed forward neural network. Where X is the input, W stands for the weights and Y is the result.

The difference between subsymbolic (connectionist) systems and symbolic systems lies in the fact that the knowledge is stored in the weights between the neurons and not symbolically in a knowledge-base. As a result, a subsymbolic KBS cannot explain or reason its decisions. The knowledge is automatically acquired by a learning algorithm (e.g. back propagation algorithm) by working through a dataset. The learning algorithm creates its own implicit rules, which are mapped in weights between the neurons. The weightings are continuously adjusted during the learning process until a specified minimum error is undershot.

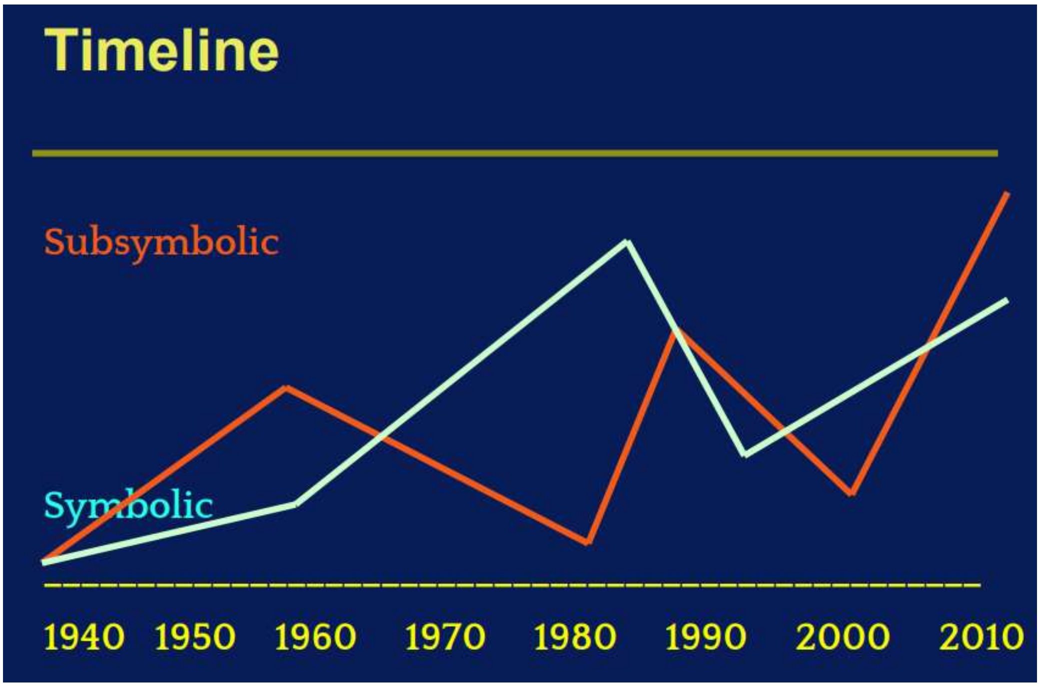


Figure 7: Timeline symbolic and subsymbolic systems (Liebermann, n.d. P.8)

Figure 7 shows the popularity of the symbolic and subsymbolic systems in a timeline.

According to Akerkar, Sajja, Pineda, Ultsch, Korus and Liebermann subsymbolic systems like a neural network have several advantages but also disadvantages compared to symbolic systems and in general:

Advantages:

- Each neuron can work independently, therefore parallel implementation is possible (Akerkar & Sajja, 2010, p.190).
    - This can lead to a better performance than with a symbolic system (Lieberman, n.d. p.3).
- A neural network can solve every problem that can be represented as a pattern, also nonlinear problems (Akerkar & Sajja, 2010, p.190).
    - Lieberman claims that subsymbolic systems are better for perceptual problems than symbolic systems (n.d, p.3). Akerkar and Sajja state the same (2010, p.191).
- Neural networks are suitable for generalization (Akerkar & Sajja, 2010, p.190).
    - Neural networks can manage unseen patterns and generalize from the training set (Akerkar & Sajja, 2010, p.191).
- A neural network can deal with noisy input data. (Akerkar & Sajja, 2010, p. 190)
    - Symbolic KBS are not as robust as subsymbolic systems against noise (Lieberman, n.d. P.3).
    - Ultsch and Korus also claim that "Knowledge-based systems often fall short in handling inconsistent and noisy data" (Ultsch & Korus, 1995, p.1).
    - An example would be speech recognition.
- A neural network can learn by itself. This means that the knowledge does not have to be updated manually. Only a training set has to be provided.

Disadvantages:

- An artificial neural network cannot fully mimic the human brain or intelligence.
  - The number of neurons in the brain is on the order of 10^11, whereas the typical artificial neural network has 10s or 100s neurons. (Akerkar & Sajja, 2010, p.190)
- A neural network cannot explain or reason its decisions, because the knowledge of a neural network is stored implicitly in the weights between the neurons (Akerkar & Sajja, 2010, p.191).
- If you have 200 input neurons and wish to extend to 201 input neurons, you have to build a new neural network. That means you have to discard the already trained neural network and completely retrain the new one. The architectural changes of a neural network, even if they are insignificant, involve much effort (Akerkar & Sajja, 2010, P.191).
  - Nevertheless, Liebermann claims that subsymbolic systems are generally easier to scale up (n.d, p.3).
- Depending on the task, training a neural network can take much time.
- In order to train a neural network, you typically need large datasets with predefined results (supervised learning) (Pineda, 2017). Creating these datasets can be very expensive.
- If a neural network exceeds a certain number of neurons, training and evaluation will be very slow (depending on the hardware).

According to Liebermann symbolic systems are also

- Easier to debug
- Easier to explain
- Easier to control
- More useful for explaining thoughts of people
- And better for abstract problems

than subsymbolic systems (n.d., p.3).

6.3 Combination of Subsymbolic and Symbolic KBS

When I got into the subject, my first thought was that a combination of symbolic and sub-symbolic systems could largely erase each other's weaknesses. This thesis is supported by Ultsch and Korus (1995). Subsymbolic KBS have trouble with explaining and reasoning their decisions. However, symbolic systems are not good at handling noisy or unstructured data such as image recognition and voice recording. Based on Ultsch and Korus (1995, p.1), it is also a known fact that the description of knowledge learned from experience is very difficult or even impossible and thus also the representation of this knowledge in the knowledge base.

In this chapter, I would like to show how symbolic and subsymbolic systems can be integrated:

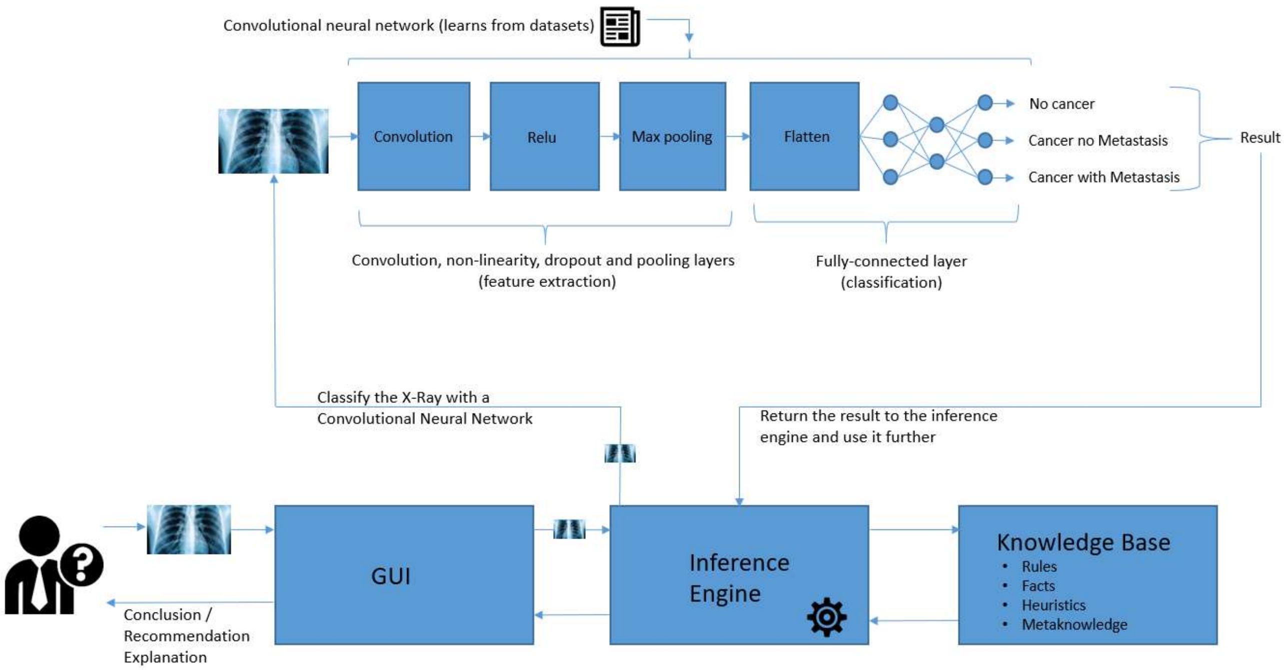


Figure 8: Combination of symbolic and subsymbolic KBS

Figure 8 shows how such an integration could look like. A user enters an x-ray image via the graphical user interface into the KBS. From the GUI the image is transferred to the inference engine, which in turn passes it on to a convolutional neural network (CNN). Convolutional neural networks are mostly used for image or speech recognition. Using a CNN is appropriate here, as according to Korus and Ultsch (1995, p.1) symbolic KBS are not good at handling noisy or unstructured data. The CNN has already been trained using a dataset that includes X-rays and the associated diagnosis. The CNN will then process the X-ray image and draw a conclusion (e.g. has cancer with metastases). This result is then passed on to the inference engine. From there it proceeds as usual. Based on rules, facts, heuristics and metaknowledge, a suitable therapy can be suggested. It is important to understand that the system cannot explain how the CNN came to the result, because the knowledge is stored subsymbolically in the weights. But how the inference engine chooses the therapy can be justified, because from here on, the knowledge is stored symbolic in the knowledge base.

Out of my own interest, I have written a project in C# based on my previous and newly acquired knowledge through this work. It is an implementation to demonstrate the integration of symbolic and subsymbolic systems by using a very simplified example.

The project contains the following components, where each component was written from scratch to guarantee a complete insight into the functioning of the system:

Neural network

The neural network is a basic feedforward neural network with backpropagation as learning algorithm. Sigmoid and Tanh are implemented as activation functions. The neural network has been tested against the XOR (non-linear problem) and OR problem (linear problem).

GUI

The GUI was implemented with Windows Forms.

Inference Engine

The inference engine uses the trained neural network. It applies rules and facts from the knowledge base.

Knowledge base (Mocked)

The mocked knowledge base stores rules and facts.

You can find the project on GitHub and the user manual in the attachment.

https://github.com/michelschlatter/Tobit_Project

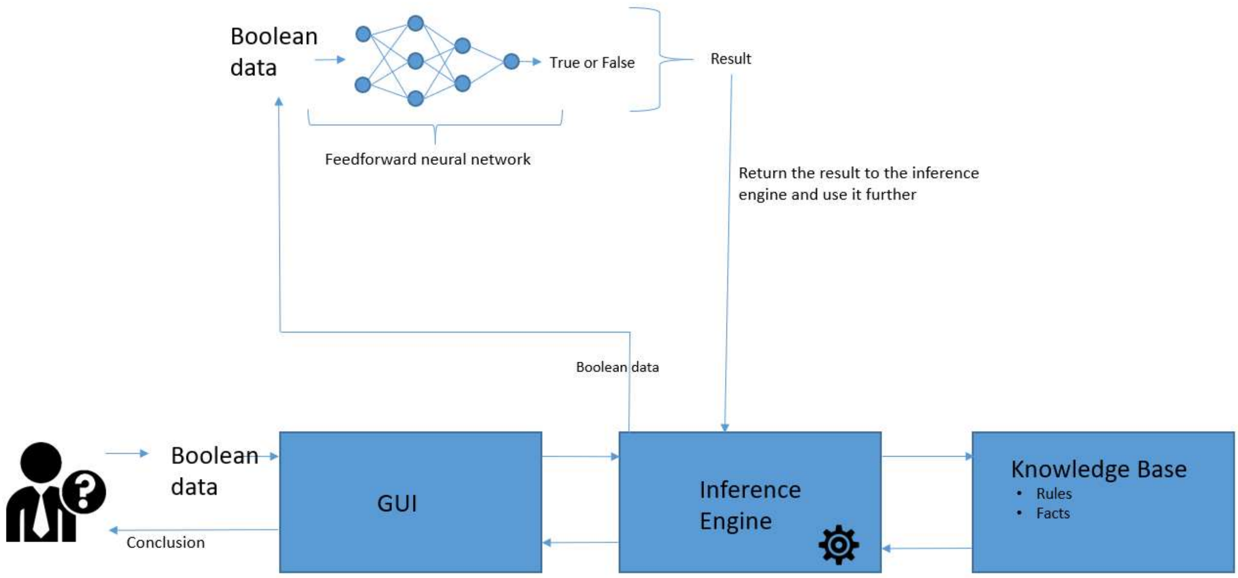


Figure 9: Combination of symbolic and subsymbolic systems in my own project

Figure 9 illustrates how the process flow of my program works.

In summary, my program is a very simplified KBS which checks if two magnets attract or repel each other. A part of the knowledge is implicitly represented in a neural network. The inference engine then uses the result of the neural network and the knowledge of the knowledge base to reach a conclusion (i.e. if the magnets attract or repel) which is after presented to the user.

7. History and Fields of Application

Based on Alberico and Micco (1990), some historical milestones in the development of knowledge-based systems and their first application fields are shown below:

First Generation (pre-1956)

Warren MCCulloch developed an artificial neural network model of the brain at the University of Illinois. This model described that the synapses and neurons behave in a binary way (i.e. fire or do not fire). Although it later turned out that neurons do not behave strictly digitally, the model still had a considerable influence (p.24).

Second Generation (1956-1970)

Around 1957 Alan Newell and Herbert Simon developed the "General Problem Solver", which would tackle any problem presented to it. This approach, however, was not successful. Subsequently, only approaches were pursued in which domain specific problems were to be solved (p.25).

Figure 10 shows the relationship between the performance of a KBS and its purpose, whereas the more general the purpose is, the less power the KBS has. It was recognized, that the operational area must be narrowed.

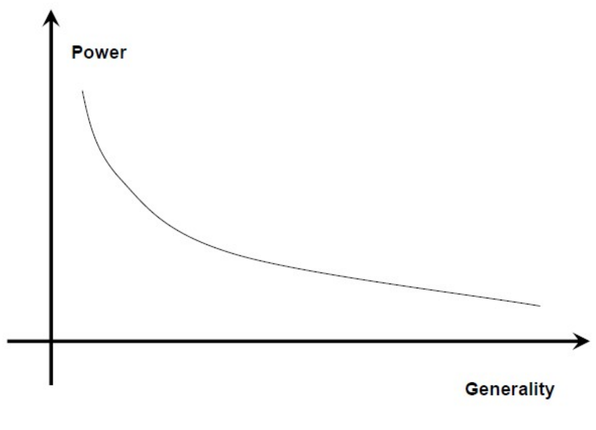


Figure 10: Power of a KBS in context with generality (Smith, 1985, P. 6)

One of the first expert systems (Dendral) was developed by Edward Feigenbaum in 1968. Feigenbaum is also called "the father of expert systems" (Enslow, 1989, p.1) and was a student of Newell. Feigenbaum was the first artificial intelligence developer which sat down with a human expert to determine the heuristics and the constraints involved in solving a complex problem. Dendral was used in chemical analysis for discovering the molecular structure of a substance and represents a major paradigm shift in AI research (p.26).

Between 1970 and 1982

Between 1970 and 1982, many large KBS were developed. A well-known KBS from the 1970s was MYCIN. MYCIN identified bacteria that caused serious diseases such as bacteraemia and meningitis and gave recommendations on the choice and dosage of antibiotics. It was also used to diagnose blood clotting disorders. However, MYCIN was never accepted in the medical community. Probably because it was restricted to a narrow domain (p.27).

KBS have developed strongly and are so widespread that most people do not even notice that they are dealing with a KBS. A good example are the shopping suggestions on online shopping sites (Zahir, 2002, p.1). These suggestions are based on the user's profile, complementary goods and what other users have bought.

According to Akerkar and Sajja (2010), knowledge-based systems can be used for the following applications:

Advisory

A KBS is an effective advisory system because it is goal-oriented and adaptive, able to deal with uncertainties and can explain its actions (p.43). An example of an advisory system would be a navigation device that suggests the best alternative route as soon as there is a traffic jam on the actual route.

Health care and Medical Diagnosis

A KBS can also be used in medicine. It can for example analyse dermatology reports, sonography reports, CAT scan reports, and so on. With the help of a medical KBS, a second opinion can be obtained. Furthermore, a medical opinion can be cross-referenced relatively cheaply. These systems can be made available at any time and place, for example via the internet (p.44).

Tutoring

The tutoring systems need to interact with the users and must provide access to the learning material in a cost-effective way. They also should understand the user's natural language. In a tutoring system, the explanation component is essential (p.44).

Control and Monitoring

An example of this would be a system that monitors a patient after surgery and intervenes appropriately when necessary (p.44).

Prediction

The stock market charts contain various indications of a future up or down. A KBS can be used for the analysis of these charts (p.44).

Searching Larger Databases and Data Warehouses

With the help of a KBS, information (also from large or multiple large databases) can be filtered and sorted according to a user profile in order to display only the information best suited to the user (p.44).

Knowledge-Based Grid and Semantic Web

A KBS can map the different ontologies on the semantic web to avoid redundant information and manage web resources in a meaningful way (p.44).

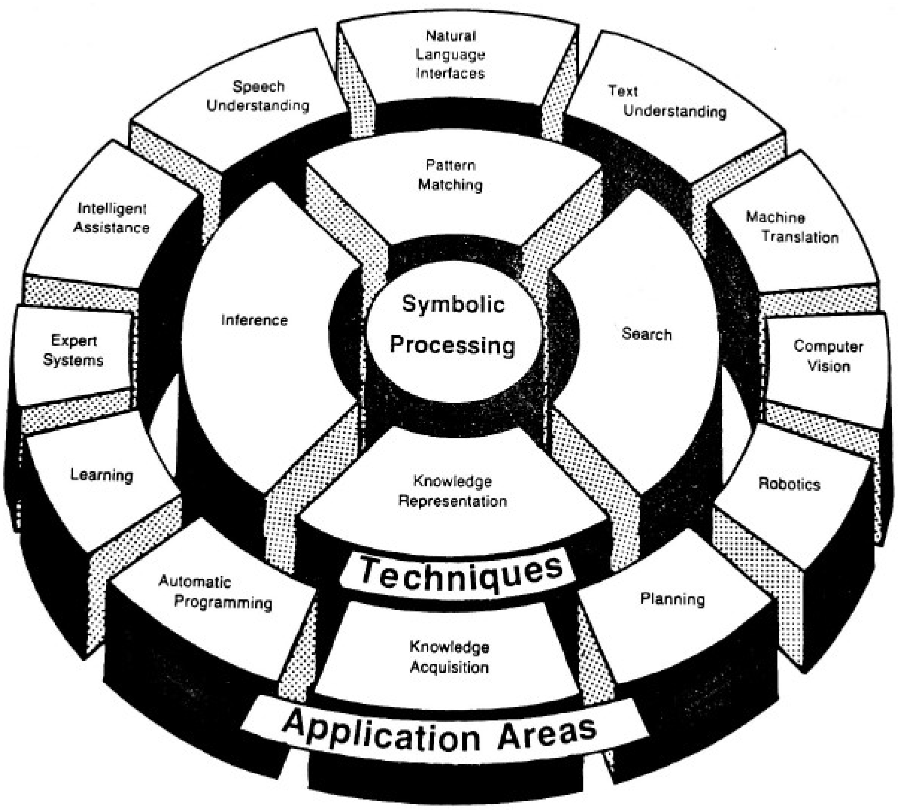


Figure 11: Application areas of symbolic KBS (Alberico & Micco., 1990, p.29)

Figure 11 shows the different techniques and application areas of symbolic KBS developed until 1990.

8. Conclusion

Knowledge-based systems and knowledge engineering are very broad topics and one could write much more about it. Nevertheless, this paper provides a good and clear insight. It shows that knowledge-based systems can support people, cushion the problem of rare experts and reduce the loss of knowledge. In order to ensure this, a good quality of the KBS is essential. The quality of a KBS strongly depends on the KE. He or the machine learning algorithm acquires the knowledge for the knowledge base, which in turn is the foundation stone for the power of the KBS. The combination of symbolic and subsymbolic systems is an advanced subject. Nevertheless, the chapter shows the important finding that a combination of both systems is useful in some applications.

Artificial intelligence still has many disadvantages and limitations that have to be eliminated in the future. In my opinion, especially subsymbolic artificial intelligence should be developed more according to biological models (e.g. spiking neural networks). Around the 15 century Leonardo Da Vinci wrote the following sentence, which could be valid until today:

> *“Human subtlety will never devise an invention more beautiful, more simple or more direct than does nature because in her inventions nothing is lacking, and nothing is superfluous” Leonardo Da Vinci (n.d).*

Bibliography

Alberico, R., & Micco, M. (1990). Expert systems for reference and information retrieval. London: Meckler.

Akerkar, R., & Sajja, P. (2010). Knowledge-Based Systems. USA, Massachusets: Jones and Bartlett Publishers

Akerkar, R. (2005). Introduction to Artificial Intelligence. Delhi: PHI Learning

Akerkar, R. (2014). Introduction to Artificial Intelligence. Delhi: PHI Learning

Bendel, O. (2017). Grundlagen des Wissensmanagements. Olten, FHNW. Retrieved from: moodle.fhnw.ch

Crankshaft Publishing. (n.d). Knowledge-Based Systems (Artificial Intelligence). Retrieval date: 15.11.2018. Retrieved from: http://what-when-how.com/artificial-intelligence/knowledge-based-systems-artificial-intelligence/

Da Vinci, L. (n.d). Famous Leonardo Da Vinci Quotes. Retrieval Date: 28.11.2018. Retrieved from: https://www.leonardodavinci.net/quotes.jsp

Enslow, B. (1989). THE PAYOFF FROM EXPERT SYSTEMS. Across the Board. 1-5. Retrieved from: https://stacks.stanford.edu/file/druid:sb599zp1950/sb599zp1950.pdf

Feigenbaum, F. (1983). The fifth generation. California USA: Addison-Wesley Publishing Company

Gigerenzer, G., & Todd, P. (1999) Simple heuristics that make us smart. New York : Oxford University Press.

Hartmann, A., & Mathieu, H. (2002). Dienstleistungen im wirtschaftlichen Wandel: Struktur, Wachstum und Beschäftigung. Bochum: Ruhr-Universität Bochum. Retrieved from: http://www.ruhr-uni-bochum.de/heinze/Downloads/DLimWandel.pdf

Haocheng, T. (2017). A brief history and technical review of the expert system research. IOP Publishing Ltd. Retrieved from: http://iopscience.iop.org/article/10.1088/1757-899X/242/1/012111/pdf

Hinkelmann, K. (2010). Knowledge in Processes Introduction to Knowledge-Based Systems. Olten, FHNW. Retrieved from: http://knut.hinkelmann.ch/lectures/KE2011/KE-1-Introduction_Knowledge_In_Processes.pdf

Hinkelmann, K. (2018). Machine Learning and Knowledge-based Systems. Olten: FHNW.

Hüllermeier, E. (2007). Case-Based Approximate Reasoning. Niederlanden Dordrecht: Springer Science & Business Media.

Kvasnicka, V. (2007). Symbolic vs. subsymbolic representation in cognitive science and artificial intelligence. FIIT STU. Retrieved from: http://www2.fiit.stuba.sk/~kvasnicka/NeuralNetworks/1.prednaska/symbolic.pdf

Lieberman, H. (n.d) Symbolic vs. Subsymbolic AI. Massachusetts: MIT CSAIL & MIT Media Lab. Retrieved from: http://futureai.media.mit.edu/wp-content/uploads/sites/40/2016/02/Symbolic-vs.-Subsymbolic.pptx_.pdf

Oxford University Press. (n.d). Symbol. Retrieval date: 03.12.2018. Retrieved from: https://en.oxforddictionaries.com/definition/symbol

Pineda, M. (09.09.2017). Benefits and limitations of machine learning. Retrieval date: 04.12.2018. Retrieved from: https://www.profolus.com/topics/benefits-limitations-of-machine-learning/

Reinmann, G. (2009), Studientext Wissensmanagement. Augsburg: Universität Augsburg. Retrieved from: https://gabi-reinmann.de/wp-content/uploads/2009/07/WM_Studientext09.pdf

Sasikumar, M., Ramani, S., Raman, S.M., Anjaneyulu, K.S.R., & Chandrasekar, R. (2007). A Practical Introduction to Rule Based Expert Systems. India New Delhi: Narosa Publishing House.

Smith, R. (1985). Knowledge-Based Systems Concepts, Techniques, Examples. Ridgefield USA: Schlumberg-Doll Research. Retrieved from:

http://www.reidgsmith.com/Knowledge-Based_Systems_-_Concepts_Techniques_Examples_08-May-1985.pdf

Studer, R., Benjamins, V., & Fensel, D. 1998. Knowledge Engineering: Principles and Methods. Karlsruhe: University of Karlsruhe. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0169023X97000566

Tripathi, K. (2011). A Review on Knowledge-based Expert System: Concept and Architecture. Kolhapura India: Bharati Vidyapeeth Deemed University. Retrieved from: https://www.ijcaonline.org/ait/number4/SPE226T.pdf

Tuthill, S., & Levy, S. (1991). Knowledge-based Systems: A Manager's Perspective. USA California: TAB Professional and Reference Books

Ultsch, A., & Korus, D. (1995). Integration of Neural Networks with Knowledge-Based Systems. Margburg Deutschland: University of Marburg. Retrieved from: https://www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/publikationen/ultsch95integration.pdf

Valkov, V. (19.05.2017). Creating a Neural Network from Scratch – TensorFlow for Hackers (Part IV). Retrieval date: 18.11.2018. Retrieved from: https://medium.com/@curiousily/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8

Wu, X. (2014). Knowledge Acquisition and Machine Learning. Vermont: University of Vermont. Retrieved from: http://www.cs.uvm.edu/~xwu/Publication/ML.pdf

Zahir, S. (2002). Designing a Knowledge-Based System for the Web to Support Consumers' Online Decisions. Canada: University of Lethbridge. Retrieved from: https://pdfs.semanticscholar.org/86f5/489ff90ce80d19563ab71b864bcf9c1aa3af.pdf

Figures

Tables

List of abbreviations

Abbreviation	Meaning
AI	Artificial intelligence
ES	Expert System(s)
GUI	Graphical User Interface
KBS	Knowledge-based System(s)
KE	Knowledge engineer / Knowledge engineering

Attachment

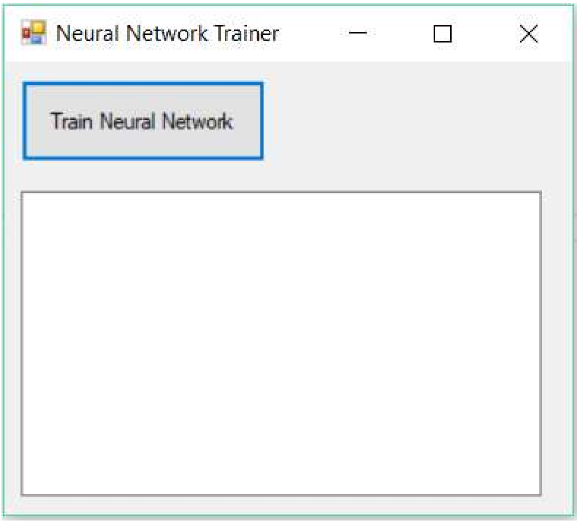
User Manual

You can download the precompiled program under the following link:

https://drive.google.com/open?id=1g6tXh1rWuGrcpq7rkn2axyb-shBaWrqK

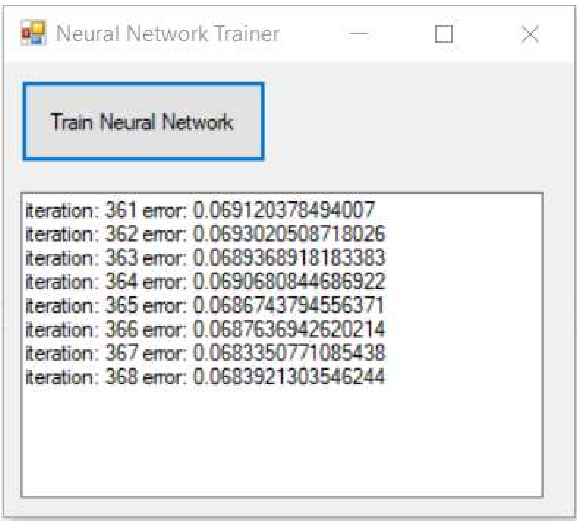
Please unzip the .rar archive and start the KS.exe. The program can only be run on a Windows environment.

First, the neural network must be trained. Please press the button "Train Neural Network" to start the process.



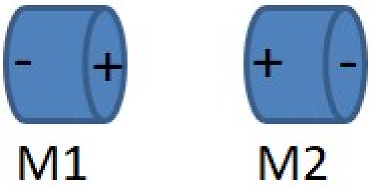
The process has now been started. In the output field you can see the current iteration and error. The error decreases with each iteration. Usually the neural network needs about 400-700 iterations to be fully trained. After the neural net has learned the pattern completely, the new window opens automatically.

Note: If after more than 1500 iterations the neural net is still not completely trained, the initial weights are badly chosen (the initial weights are chosen randomly). In this case, restart the program. However, this happens very rarely.

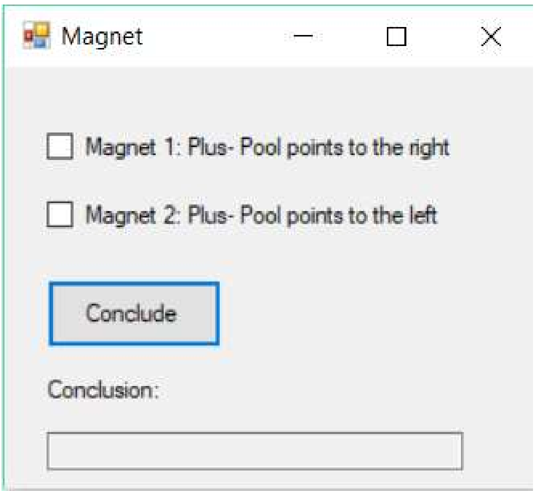


In this window you can test if two magnets attract or repel each other. You can make two entries:

Input 1: If the plus pool of the first magnet (M1) points to the right, select the checkbox.
Input 2: If the plus pool of the second magnet (M2) points to the left, select the checkbox.



When you have made your entries press "Conclude”.



The program will now make a conclusion and display it in the field "Conclusion"