EVA – An Expert system for Vases of the Antiquity

Martina TROGNITZ

Deutsches Archäologisches Institut, Berlin

Abstract: EVA is an expert system for classification and dating of ceramics. It represents an application of natural language processing in archaeology. The program has a knowledge base that consists of a decision tree. This decision tree is traversed by taking the user's answers to a set of predefined questions. The leaf nodes contain the output, a classification of the object. By answering one question at a time the user is encouraged to look closely at the object in question for relevant characteristics. In an alternative mode of the program, the user can input a description of the object in natural language. EVA then automatically extracts the answers to the predefined questions. This is achieved by information extraction, which mainly is done by searching the text for particular patterns. The knowledge base of this expert system is independent and fully exchangeable. Its syntax is simple and thus it can easily be expanded or changed by non-programmers. First tests were done with vases of the geometric style, showing that this is a practicable approach capable of good results if the knowledge base is good enough. Further work could include a component for image processing or the use of machine learning for an automatic acquisition of the decision tree or of the information extraction patterns.

Keywords: Ceramic, Classification, Computational Linguistics, Expert System.

Motivation

Ceramic is a common find at archaeological excavations. Shape and decoration depend on multiple factors like cultural environment, production place, prevailing taste and fashion, and technological achievements. All these factors change in the course of time, thus allowing the use of ceramic finds to date archaeological deposits.

But to do so, it is required to look closely at the object in order to identify the distinctive features, only known by few experts. Tom Rasmussen and Nigel Spivey (1991: i) put this in a nutshell:

> *"An ancient Greek vase is a difficult object for the non-expert to come to terms with. Faced with rows of apparently undifferentiated black, red and buff pots, he or she is at a loss as to where to begin."*

Ceramic finds are often kept for years untouched in their deposits, because there is a lack of eligible experts to work on them.

A solution to this problem could be the use of an expert system to classify and date ceramic. The knowledge of an actual human expert is stored in the system. In this way the gap between the number of human experts and the amount of unclassified ceramic can be filled.

The expert system presented in this work, EVA, can provide a second opinion and may be used as a learning tool, because the user is encouraged to look closely at the object in question for relevant characteristics by answering one question at a time.

What is an expert system?

An expert system is a program capable of solving problems similarly as human experts would do. Knowledge and inference methods are used to solve complex problems normally requiring enormous human expertise.

Expert systems represent a special subject of artificial intelligence and the first systems were developed in the sixties. The first expert system DENDRAL was developed by Feigenbaum et al. to help chemists to analyze mass spectra of organic molecules (LINDSAY et al. 1993).

Other examples cover a wide range of subjects like medicine (MYCIN), geology (PROSPECTOR) and configuration of computers (XCON/R1).

An expert system can only achieve good results in a well-defined special field, the *knowledge domain*. This compares to a human expert who usually is specialized in only a narrow field of a specific subject.

How it works

An expert system consists of a *knowledge base* and an *inference engine* as shown in (Fig. 1). The user inputs facts and information about a problem through the user interface. These and the knowledge base are used by the inference engine to draw a deduction.

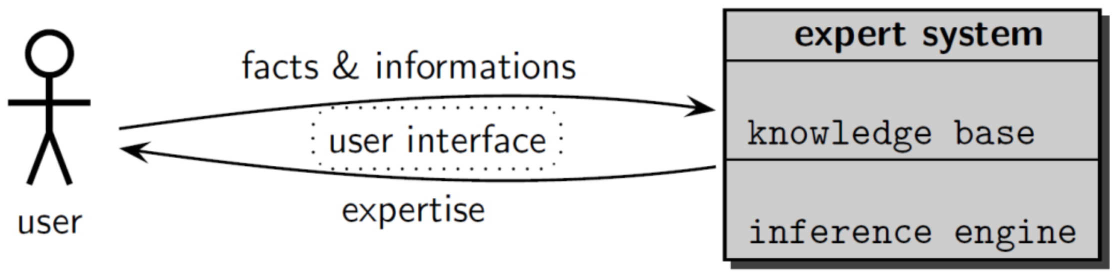

Fig. 1 – Basic functionality of an expert system.

The inference engine (Fig. 2) stores the facts given by the user in a working memory and compares them to the rules stored in the knowledge base. All applicable rules are transferred into the agenda, where they are prioritized and executed. When a rule generates a new fact, it will also traverse this process.

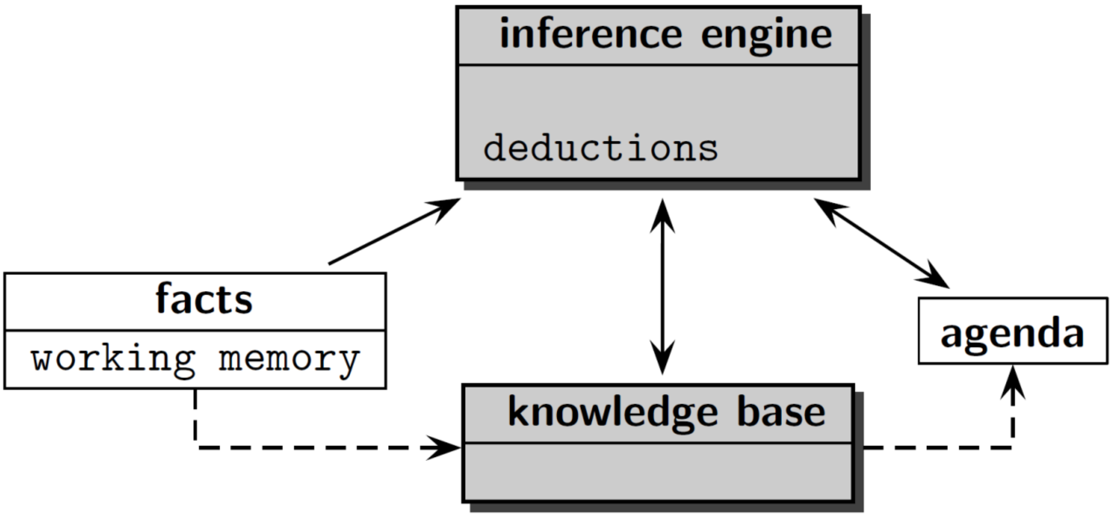

Fig. 2 – Principle of operation of the inference engine

Properties of an expert system

Expert systems have a high performance and deliver results competing with those of human experts. To be practicable the systems need to have a proper response time and be robust. Otherwise they won't be accepted by users. It is of no use, if the user has to wait several hours to get a result, or has to cope with crashes and cryptic errors.

Comprehensibility is one of the key properties of an expert system. If it can't explain how deductions were made and which rules were applied, the acceptance of the system will be little.

The knowledge of an expert system is explicitly disconnected from the processing part, thus making it very flexible. The knowledge base may be extended or exchanged at any time.

Implementation of EVA

EVA, the expert system discussed in this work, is an ***E***xpert system for ***V***ases of the ***A***ntiquity that classifies and dates ancient ceramic. The system architecture and its implications are described in the following pages.

Before implementation

As a first step the knowledge domain has to be specified. For this case I chose attic geometric ceramic, dating from 900 B.C. to 700 B.C.

The next thing to think of is the knowledge base and how to store the knowledge in it. The source of knowledge was the work from Nicholas Coldstream (2008). There is a great variety of knowledge representation formalisms, like semantic networks, logical models or rules. For EVA a rule based system based on a decision tree was chosen.

This choice emerged from the material itself. A specific vase can be described with a specific set of characteristics. For example an early geometric I and a middle geometric II amphora may be described as shown in (Fig. 3).

EG I:
figured:no; shape:amphora; handlePosition:neck; handleForm:band; body:ovoid; motifs:band of slanting lines

MG II:
figured:no; shape:amphora; handlePosition:neck; handleForm:band; body:ovoid; motifs:hatched meander, zigzag, dogtooth

Fig. 3 – Two sets of characteristics, each describing a vase

From the representation in (Fig. 3) follows that rules can be formulated, which have the form: *IF* a specific set of characteristics is given *THEN* you have specific vase.

Furthermore it is possible to represent these sets of characteristics in a decision tree, since some vases may be distinguished by only one differing attribute. The vases in (Fig. 3) for example only differ in the used motifs and could be distinguished from one another by asking a question like: *"What motifs are used as decoration?"*.

Another thing to be considered before implementation was the choice of the programming language. Although there are systems and formalisms specially designed to create expert systems, EVA was implemented with Python, because in this way the system is flexible enough to be expanded with future functionalities.

System architecture

The overall architecture of EVA is shown in (Fig. 4). A user may choose to either input a description text (B) or answer the questions directly (A). The answers to the questions lead through the decision tree stored in the knowledge base. At the end of the process the user gets back a result from the system.

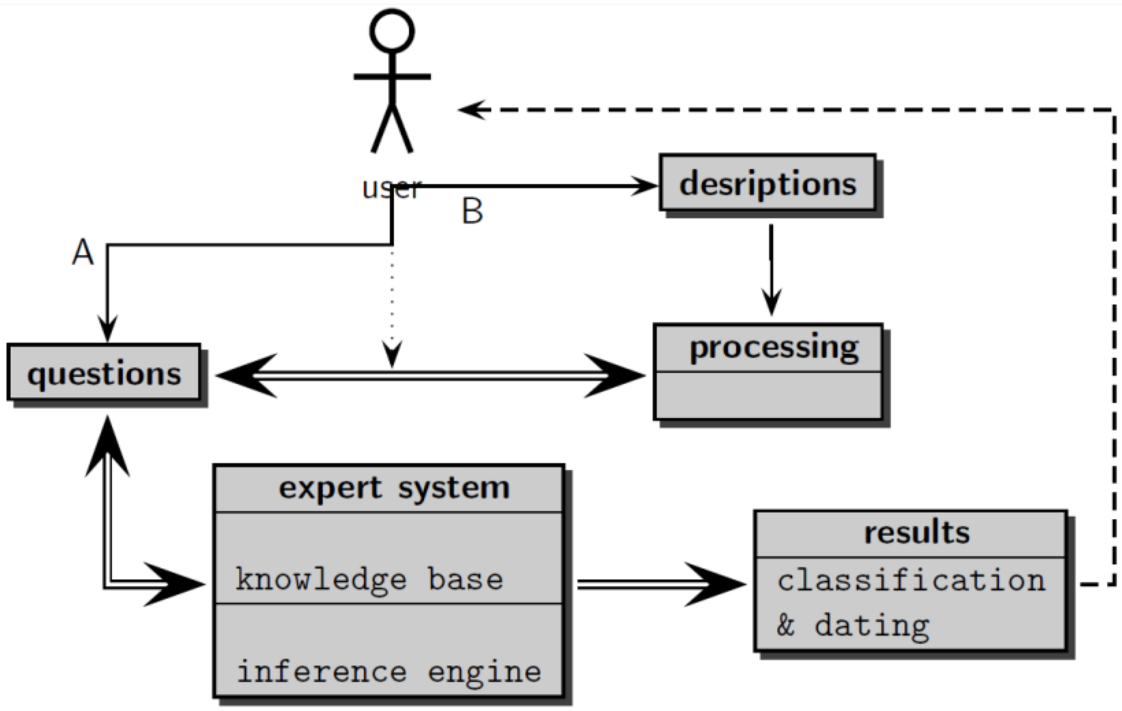

Fig. 4 – System architecture of EVA

The decision tree and the knowledge base

The decision tree is built up in the module *tree.py*, which uses the content stored in a separate text file, the actual knowledge base. Each node in the tree consists of a value, a question related to the value, a link to the parent node and a list of child nodes (Fig. 5).

The knowledge base is a simple text file where each row represents a specific vase comprised of a set of characteristics as explained earlier (Fig. 3). Each row represents a path through the decision tree from the root node to a leaf node. In (Fig. 6) some of these paths are displayed. Questions are marked with a question mark and the leaf nodes containing the answer are marked with an exclamation mark. The corresponding full texts for the questions and answers are stored in the separate file *dictionaries.py*.

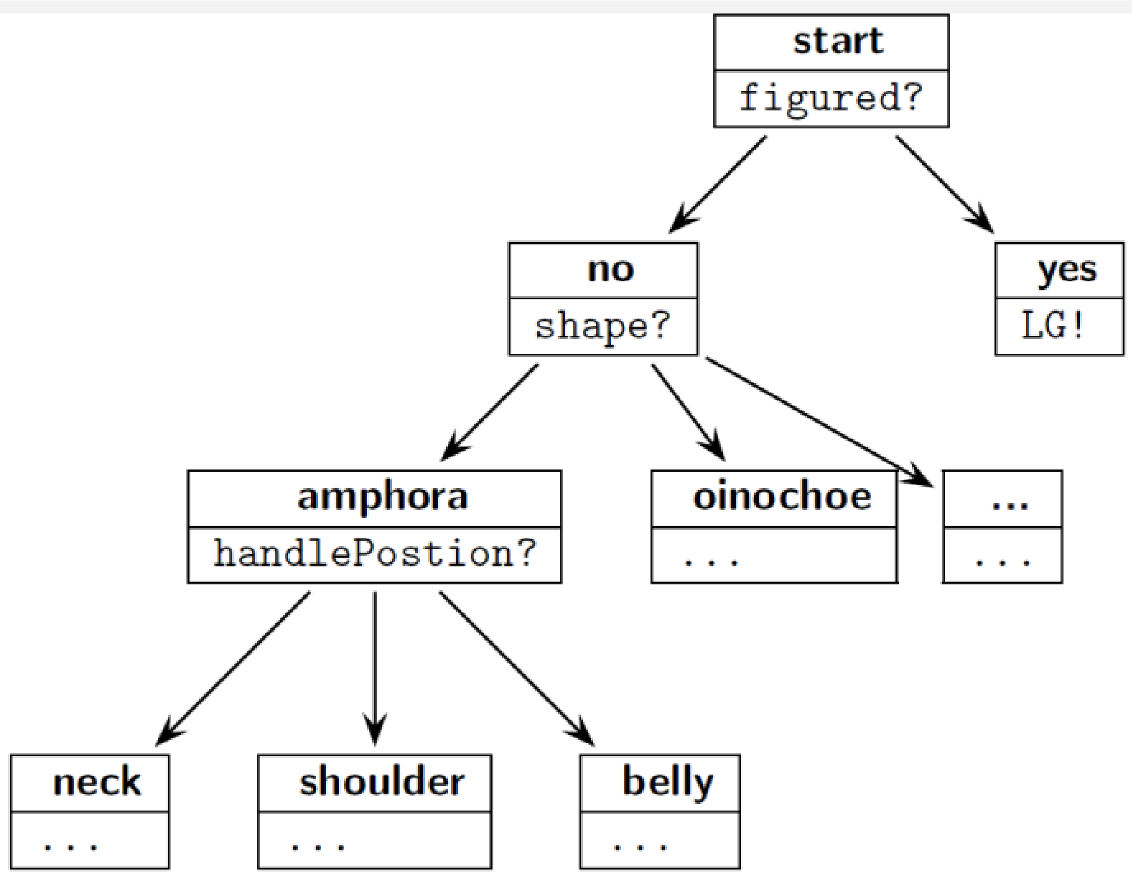

Fig. 5 – Part of the decision tree of EVA showing the components of each node

```
root - figured? : yes - LG!
root - figured? : no - shape? : amphora
                                 - handlePosition? : neck - ...
root - figured? : no - shape? : amphora
                             - handlePosition? : shoulder - ...
root - figured? : no - shape? : amphora
                                - handlePosition? : belly - ...
root - figured? : no - shape? : oinochoe - ...
root - figured? : no - ...
```

Fig. 6 – Example of the text file containing the knowledge base

Description texts

If the user decides to input a description text and wants EVA to do the rest of the work, the two blocks on the right side of (Fig. 4), reached by option B, come into play.

Let’s take a look at an example text describing a small oinochoe from the CVA Oxford 4 ((GB, 24) p. 12, plate 30 1-3)):

This is a small oinochoe. The handle is outlined and has a wavy line. The rim is decorated with three horizontal lines. On the neck a horizontal row of dots, a horizontal panel with a lozenge chain and another row of dots can be seen. The shoulder is decorated with a dotted snake and some sparse dots. On the belly are linked dots. All ornaments are interspersed by encircling bands. The lower part of the vase is covered by a thin layer of clay.

The information that tells us how to classify the oinoche is in the text and has to be extracted. This is an easy task for humans, but requires a row of algorithms and commands for computers. Also the given example text is only one way of many possible to describe the oinochoe in question.

Form and structure of natural language texts depend on personal style, language skills and the personal knowledge of the subject, amongst several other circumstances. There even may be some information missing.

Thus the given text has to be processed by EVA to extract the relevant information in order to automatically answer the questions when going through the decision tree. In the first step the text is parsed with the Stanford Parser (KLEIN & MANNING 2003). The parser analyzes the structure of a sentence and does part of speech tagging. The output of the used parser can be seen in (Fig. 7). It shows the analyzed sentence: *"This neck-handled amphora has a thick barred rim."*

```
(ROOT
  (S
    (NP (DT This) (JJ neck-handled) (NN amphora))
    (VP (VBZ has)
      (NP (DT a)
        (ADJP (JJ thick) (JJ barred))
        (NN rim)))
    (. .)))
```

Fig. 7 – Output from the Stanford Parser for the sentence "*This neck-handled amphora has a thick barred rim.*"

The next step consists of answering the questions of the decision tree with the given description text automatically. This task, *information extraction,* is done by searching for specific patterns in the text.

How this works is best explained with an example. For the question: *"At which part of the body are the handles attached?"* the answer may be encoded in different ways:

This neck-handled amphora has a thick barred rim.
The handles of this amphora are attached to the neck.

This is a neck-handled amphora with a thick barred rim.
This amphora has a thick barred rim. The handles are on the neck.

From these few sentences there already can be defined two patterns to look for:
…<string>-handled…
…handles <verbal phrase with on/to> ... <string>

Either the part of the body is found in the string directly attached to *"-handled"* or it is the string following a construction consisting of the word *handles* and a verbal phrase containing the prepositions *on* or *to*.

The patterns used by EVA are all hand-crafted and thus don't cover all possible formulations. It is possible to generate the patterns automatically with machine learning techniques, but for this, a large (digital) text corpus of vase descriptions is needed.

Another problem to deal with, when coping with natural language, are synonyms. A thesaurus could be helpful.

If EVA can't find an answer for a specific question while traversing the decision tree, the user is asked to help out. When the final result is reached and displayed, EVA also prints out how the tree was traversed and which answers were given.

Components of EVA
The main module is called *core.py* and it builds the decision tree with *tree.py* and a given text file with the knowledge base. User friendly questions and patterns for information extraction are stored in the module *dictionaries.py* and use Pythons built-in data structure dictionary.

The Stanford Parser is started from within *core.py*. The program only has a command-line interface.

Future work

EVA is in an experimental status and leaves plenty of room for improvements and functional additions. The knowledge base could be expanded by more regions and earlier and later periods. A graphical user interface with example images would improve the interaction between the program and the user. The use of example images would also help to clarify phenomena that can only poorly be described by words.

The introduction of a certainty factor may help in measuring how good the given results are. Another expansion for EVA could be the support for other languages.

The use of machine learning to learn the patters has already been mentioned, but it can also be used to build and expand the knowledge base if proper training data is provided. Last but not least image recognition could be used to classify a vase directly by processing an image, and thus skipping the part of a human describing the object; on the contrary the program may even provide an automatically generated description.

References

BARCELÓ, J. A. 1996. Heuristic Classification and Fuzzy Sets. New Tools for Archaeological typologies. *Acta Praehistorica Laidensia*, 28, 155-164.

BIRD, S., KLEIN, E. & LOPER, E. 2009. *Natural Language Processing with Python*

CHEIN, M. & MUGNIER, M.-L. 2009. *Graph based knowledge representation,* London

COLDSTREAM, J. N. 2008. *Greek Geometric Pottery,* Exeter, Devon

ERICSON, J. E. & STICKEL, E. G. 1973. A Proposed Classification System for Ceramics. *World Archaeology*, 4, 357-367

FEIGENBAUM, E. A. 1992. A personal view of expert systems: Looking back and looking ahead. *Expert Systems with Applications*, 5, 193-201

FEIGENBAUM, E. A. 2003. Some challenges and grand challenges for computational intelligence. *Journal of the ACM*, 50, 32-40

GIARRATANO, J. C. Giarratano & RILEY, G. D. 2005. *Expert Systems. Principles and Programming*

JACKSON, P. 1999. *Introduction to Expert Systems*, Harlow

KAMPEL, M., SABLATNIG, R. & MARA, H. 2002. Automated Documentation System for Pottery. *In*: MAGNENAT- THALMANN, N. & RINDEL, J. (eds.) *Proc. of 1st International Workshop On 3d Virtual Heritage, Geneva, Switzerland,* Geneva, 14–20

KLEIN, D.& MANNING, C. D. 2003. Accurate unlexicalized parsing. *In*: HINRICHS, E. W. & ROTH, D. (eds.) *ACL 2003,* 423-430.

KUNISCH, N. (ed.) 1998. *Ornamente geometrischer Vasen*, Köln

KURBEL, K. 1992. *Entwicklung und Einsatz von Expertensystemen*, Berlin, Heidelberg

LINDSAY, R. K., BUCHANAN, B. G., FEIGENBAUM, E. A., LEDERBERG, J. 1993. DENDRAL: a case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61, 209–261

MON, V. & PAIJMANS, H. 2007. Secanto: a retrieval system and classification tool for simple artefacts. *In*: POSLUSCHNY, A., LAMBERS, K. & HERZOG, I. (eds.) *Layers of Perception. Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA), Berlin, April 2–6*, Berlin

PORTER, D., WERNER, P. & UTCKE, S. 2005. *Ancient Ceramics: Computer aided Classification*

PUYOL-GRUART, J. 1999. Computer Science, Artificial Intelligence and Archaeology. *In*: BARCELÓ, J. A., VILA, A. & BRIZ, I. Briz (eds.) *New Techniques for Old Times. Computer Applications in Archaeology,* Oxford

RASMUSSEN, T. & SPIVEY, N. (eds.) 1991. *Looking at Greek Vases*, Cambridge

SHORTLIFFE, E. H., 1974. A rule-based computer program for advising physicians regarding antimicrobial therapy selection. *In*: Proceedings of the 1974 annual ACM conference - Volume 2, ACM, 74, 739–739

STUTT, A. 1987. *Second generation expert systems, explanations, arguments and archaeology, HCRL Technical report*

van den DRIES, M. H. 1998. *Archaeology and the application of artificial intelligence: case-studies on use-wear analysis of prehistoric flint tools,* Leiden

Imprint:

Proceedings of the 17th International Conference on Cultural Heritage and New Technologies 2012 (CHNT 17, 2012)

Vienna 2013

http://www.chnt.at/proceedings-chnt-17/

ISBN 978-3-200-03281-1

Editor/Publisher**:** Museen der Stadt Wien – Stadtarchäologie

Editorial Team: Wolfgang Börner, Susanne Uhlirz