

# **APPLICATION OF NOUN ALGORITHM TO THE WORD “KOLPER”**

**Carlos Lollett**

**April 26, 2004**

**CSE740 Contextual Vocabulary Acquisition**

## **Abstract**

In Contextual Vocabulary Acquisition (CVA), neologisms can be learnt from sentences. These sentences and their related background knowledge should provide a proper context in order to apply CVA techniques. In Daalen-Kapteins 1981[1] a set, consisting of a word and five sentences. They were studied in order to understand, how different groups of individuals could obtain the meaning of the word from the five sentences where that word was used. That word, “kolper”, was used in every sentence, each one giving a chunk of information about its meaning. This project tries to emulate Daalen-Kapteins research, constrained to only one sentence. This emulation is achieved by representing the sentence and the proper background knowledge in SNePS, whose knowledge representation structures consists of semantic networks; followed by the noun algorithm to organize information.

# 1 Introduction

## *1.1 Contextual Vocabulary Acquisition*

For Human beings, having the ability of represent objects, situations and properties by words is as natural as breath. However, the number of elements to be represented is overwhelming. Research about the number of words only for printed school English [2] shows that techniques like root learning and mnemonics are not sufficient to let a person to understand mostly of the written world. Since our speech and writing is not a series of unconnected words, they contain more information together than each separate word. CVA uses the relational information between words inside specific contexts to guess unknown parts (words). This process could be compared to the process of transmitting data over a communication link; since the data is not completely independent compression and/or error correction are possible. It is surprising how we use these techniques everyday without be conscious of that.

Contexts have been classified [3] in internal and external context. Internal context consists of all cues and clues giving by the word origin and structure, while external context consists of the words that are closest to the targeted word.

In addition to the information contained in written passage, human beings can use also previous knowledge (background) to obtain more information about the missing meanings.

## ***1.2 Semantic Network Processing System (SNePS)***

In Knowledge representation research, applications that attempt to represent knowledge usually address two main questions:

1. What are they representing?
2. How are they representing whatever they are representing?

SNePS (“Semantic Network Processing System”) developed by Stuart Shapiro and his research group represents knowledge by representing “entities” and the relation among them. The entities can be represented as so sets can be represented by extension (list of the elements belonging to a concept or entity) or by intension (qualitative description of the properties that characterize the elements of the concepts against the elements which are not belonged to the same concept). Although SNePS is capable to represent extensional entities, the developers’ goal was to represent intensional entities.

The characteristics, according Shapiro et al [4], of the intensional entities would be:

1. The equivalence relation between entities requires the same intensional context. This is the case of the morning and evening star which, even being the same astral body correspond to two different intensional entities.
2. They are not required to be exhaustive determined covering all the possible details in one entity. For example, Dr. Watson’s right arm scars or any other characteristic omitted in the description of a real or imaginary character.
3. They are not required to be physically feasible, like a round square.

4. They are not required to exist (probed existence), like a unicorn.
5. They are supposed to be distinguishable even their share the same consequences.

Shapiro et al. claim “a knowledge-representation and reasoning system must model only intensional entities”[4], is supported by two arguments:

1. Fine-Grain Representation, that’s extend the intensional context in the description of the entity, that’s matter more than the physical or imaginary object itself is the way it is thought that distinguish it from the others entities, even the entities which are physically equivalent.
2. Displacement, this principle emphasize the role of the way a concept is thought is that define the entity, giving chance of representation to imaginary and impossible entities.

Once SNePS is claimed to be designed to represent intensional entities, the next step is described the technique i.e. how are these entities represented in SNePS. SNePS primary data structure is a prepositional semantic network. In words of Shapiro and Rapaport[2]: “a semantic network is usually thought of as a labeled, directed graph, whose nodes represent entities and whose arcs represent binary relations between entities”. However, this definition bring some limitations like the idea of negate a statement. According graph rules the arcs cannot be negated and their labels are just ways to call an arc. Even thought, the entities can be negated, it brings on the table the problem of having negative objects (complements). The solution of SNePS developers was to create a propositional nodes where the nodes can be also propositions that can

be negated i.e. a proposition can be true or false, so a negation is perfectly possible. That's probably the main reason behind the propositional character of SNePS.

One of the most important principles in SNePS is the uniqueness, that means two nodes cannot represent the same entity [4].

As a software package several modules compose SNePS [4]. They are:

1. SNePS 2, a system for building and connect nodes. It is also responsible of some administrative task like save/load the network to/from a file.
2. SNIP, SNePS Inference Package is in charge to represent the reasoning.
3. SNeBR, the SNePS Belief Revision system, performs checks on the coherence of the knowledge base and tries to correct with user help the possible contradictions.
4. SNaLPS, the SNePS Natural Language Processing System, it has morphological analyzer and grammar interpreter and compiler in order to be able to interface with external sources.
5. SNePSUL, the SNePS User Language, is the programming language used to the standard SNePS input.
6. XGinseng, an X Windows-based graphical application to develop networks.

One of the goals of SNePS, as many other knowledge representation applications, is to model a mind. Answering the question: who is thinking something or in other words establishing

the context in which the knowledge represented requires a starting point. The answer is the Cognitive Agent of the SNePS System-an Intelligent Entity (CASSIE) [4]. This is the mind behind the semantic network.

### ***1.3 CVA Project***

CVA project tries to emulate human being reasoning about guessing words using context information into intelligent agents. Understanding the process of vocabulary acquisition in artificial intelligence could give us important information to develop methods to help persons to increase their vocabulary by reading. In this project CASSIE is used to emulate a person who is reading a passage and then based on the passage and its background knowledge output some information about an unknown word.

The main difficulty in this research is the lack of uniform criteria about background information and representation. Emulate real situation should be prioritized over getting the right answer. Forcing CASSIE to give the right answer would make less likely to export results to help persons.

CASSIE can store information provided by sentences and make inferences in order to obtain more knowledge. However, CASSIE by itself does not show concept information in a friendly and organized form. In order to achieve this requirement, various algorithms were developed. One of them the 'noun algorithm' is able, with the proper SNePS representation, to organized all the information relevant to one single generic noun.

## 2 One Word one Sentence

The word and sentence extracted from Van Daalen-Kapteijns[1987]:

“He virtually always studied in the library, as at home he had to use artificial light all day because of those kolpers”

The targeted word is ‘kolper’. I couldn’t find this word in any dictionary. According Van Daalen-Kapteijns[1987], “kolper is a window that transmit little light because of something in from of it”. This concept is not common in our environment, which is a factor to consider when the results are analyzed. In order to analyze this word first the sentence was separate in three parts:

1. He virtually always studied in the library
2. as at home he had to use artificial light all day
3. because of those kolpers

In order to gather background knowledge, an informal survey was done. A variety of answers were obtained defining kolper. Looking at the phrase we can say that its information is mainly functional. From only this sentence is not likely to obtain precise information about ‘kolper’ characteristics. However, a common factor among those who got a closest answer was the perception of lacking of natural light at ‘home’ and in less degree that the light wasn’t entering inside home.

An important issue about passage representation differences between humans and CASSIE is that while the programmers try to represent the exact structure of the sentences, human beings change the sentence according their requirements. For example, some individuals assumed that the study place at home was the room. This would make individuals more likely to get a good answer (closest to the real meaning), but on the other hand more likely to get a bad answer (completely misdirected from the real meaning).

Background knowledge used for this project can be summarized:

1. Libraries and homes are member of the indoor locations
2. Saying that someone (human) needs to use artificial light to work all day in an indoor place is equivalent to say that the place has bad natural illumination.
3. If a place has bad natural illumination means that the light cannot enter that place.

There are two more rules added:

1. Transitivity of cause over equivalence relation
2. Probable implication of function from cause-effect relationship.

## **3 SNePS Representation**

### **3.1 SNePS Case Frames**

The use of case frames is constrained by the noun algorithm case frame dictionary (source: [www.cse.buffalo.edu/stn2/cva/case-frames](http://www.cse.buffalo.edu/stn2/cva/case-frames)).

From Noun Algorithm Case Frame dictionary

- ◆ agent/act/act
- ◆ member/class
- ◆ superclass/subclass
- ◆ object1/rel/object2
- ◆ lex
- ◆ object/property
- ◆ object/location

Outside of Noun Algorithm

- ◆ equiv/equiv

See for syntax and semantics at appendix.

### **3.2 Representing Background Knowledge**

At the beginning, it is necessary to establish common properties of the locations used in the passage: ‘library’ and ‘home’. They are members class libraries and homes respectively, but they also are members of the superclass indoors.

```
      ; Establishing the condition that library and home are subclasses of
indoors
```

```
      (show (describe (assert subclass (build lex "library") superclass
(build lex "indoors")))))
```

```
      (show (describe (assert subclass (build lex "home") superclass (build
lex "indoors")))))
```

Representing that it is the same to require artificial light all day at home is equivalent to say that home has bad natural illumination. This assertion is not always true, but it has to do with an association with the most common situation.

```
      ; Another Rule-based inference in order to establish a equivalence
relationship
```

```
      ; 'If somebody is human and someplace is an indoor then To say
somebody(human) is in someplace and it is required to use artificial light
all-day is equivalent to say that someplace must be a bad natural iluminated
place
```

```
(show ( describe (
      assert forall ($place $person)
      &ant (build member *place class (build lex "indoors"))
      &ant (build member *person class (build lex "human"))
      cq (build
          equiv ( build
              object1 (build
                  agent *person
                  act ( build
```

```

        action ( build
                object (build
                        object (build lex "work" )
                        location *place
                )
                property ( build lex "all_day")
        )
    )
    rel (build lex "require")
    object2 (build lex "artificial light")
)
equiv (build object *place property (build lex "badlight")
)
)
)))

```

Here, there is a piece of reasoning for CASSIE. The use of transitivity in math and logic is in someway extensible to normal human being reasoning. However, this link is more associative than a logic reasoning. In this case the expression to be represented: if a cause b and b and c are equivalent then a cause c.

```

; A ruled based to extend the function relationship to equivalent
concepts

```

```

(show (describe (assert forall ($myv1 $myv2 $myk)
        &ant (build equiv *myv1 equiv *myv2)

```

```

        &ant (build object1 *myk rel (build lex "cause") object2
*myv1)

        cq (build object1 *myk rel (build lex "cause") object2
*myv2)
    )
))

```

Here, there is simple implication from having bad natural illumination and the lack of entering light in a location.

; Rule-based inference. "if a place is a bad natural illuminated place then the light is not entering in that place

```

(show (describe (assert forall $place2
        ant (build object *place2 property (build lex
"badlight")))
        cq (build object *place2 property (build lex "ligh_
not_entering_there")))
    )
)
)

```

This is another example of reasoning, not logically accurate but used by humans for simplification: if a cause b, probably a's function is b

```

; Establishing the a probable relation between causality and funtion
(show (describe (assert forall ($st1 $st2)

```

```

                                ant(build object1 *st1 rel (build lex "cause")
object2 *st2 )
                                cq (build object1 *st1 rel (build lex "function")
object2 *st2)
                                )
                                )
                                )

```

### ***3.3 Representing the passage***

In order to represent the passage, the sentence was divided in three related units.

```
;Three sections connected by cause-:
```

```
; 1) He virtually always studied in the library
```

```
; 2) at home, all day, he had to work by artificial light
```

```
; 3) those kolpers
```

```
; 1) He virtually always studied in the library
```

```
he is generic human being that I called henry
```

```
; he is human
```

```
(show (describe (add member #henry class (build lex "human"))))
```

the location is specific place named mylibrary and belongs to the class library(for extension to indoors).

```
; location is a library
(show (describe (add member #mylibrary class (build lex "library"))))
```

Here, a complicated statement, in order to simplify the sentence I joined virtually + always. Since virtually only affects always, the rest of the expression meaning does not change. Here CASSIE reads that henry studied at 'mylibrary' virtually always

```
; He virtually always studied in the library
(show (describe (add
  object (build
    object (build
      agent *henry
      act (build
        action (build
          object (build
            object (build lex study)
            location *mylibrary
          )
          property (build lex "virtually always")
        )
      )
    )
  )
))
```

```
)  
)))
```

This part implies a situation

```
; 2) at home, all day, he had to work by artificial light
```

```
myhome is of class home(by path-rule, of class `indoors`)
```

```
; myhome is a home
```

```
(describe (add member #myhome class (build lex "home")))
```

Describing the situation syntactically: henry, working at home during the day require artificial light.

```
;at home, all day, he had to work by artificial light
```

```
(show (describe (add
```

```
    object1 (build
```

```
        agent *henry
```

```
        act ( build
```

```
            action ( build
```

```
                object ( build
```

```
                    object (build lex "work" )
```

```
                    location *myhome
```

```
                )
```

```
            property (build lex "all_day")
```

```
        )
```

```
    )
```

```
)
```

```
rel (build lex "require")
```

```

        object2 (build lex "artificial light")

    )))

```

; 3) those kolpers.

Our object referred as kolper is an object of class kolper

```
(show (describe (add member #mykolper class (build lex "kolper"))))
```

### Kolper is included in the causal relation

; Causal relation from 3(cause) and 2(consequence) using function relation, to test noun algorithm(besides function there is no other relation used in noun algorithm)

```
(show (describe (add
  object1 *mykolper
  rel (build lex "cause")
  object2 (build
    object1 (build
      agent *henry
      act ( build
        action ( build
          object ( build
            object (build lex "work" )
            location *myhome
          )
          property (build lex "all_day")
        )
      )
    )
  )
))

```

```

    )
    rel (build lex "require")
    object2 (build lex "artificial light")
  )
)))

```

There is causal relation between part 2 and 1. However, this causal relation is a reduction of a great deal of possibilities.

```

; big causal relation from 1 to 2 excluded at this stage
(show (describe (add
  object1 ( build
    object1 (build
      agent *henry
      act ( build
        action ( build
          object ( build
            object (build lex "work" )
            location *myhome
          )
          property (build lex "all_day")
        )
      )
    )
  )
  rel (build lex "require")
  object2 (build lex "artificial light")
)

```



## 4 Results

After giving background knowledge and sentence representation, the noun algorithm was applied giving the following results:

1. Window as a class inclusion from information from a previous sentence.
2. 3 Cause relations
3. 3 function relations

```
; Ask Cassie what "KOLPER" means:
;; UNCOMMENT THE ONE YOU *DO* WANT
;; AND DELETE THE OTHER!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
^(
--> defineNoun "kolper")
Definition of kolper:
Class Inclusions: window,
Possible Properties: function m36, function m39, function m44, cause
m36, cause m39, cause m44,
The function and cause relations are:
    ♦ m44: kolper cause that a agent working at one place require
      artificial light all day.
(m44!
(object1
(m43
(act (m42
      (action
        (m41 (object (m40 (location b3) (object (m7 (lex work))))))
          (property (m8 (lex all_day))))))))
```

```

(agent b1)))
(object2 (m10 (lex artificial light))) (rel (m9 (lex require))))
    ♦ m39 kolper cause bad natural illumination in a place
(m39 (object b3) (property (m11 (lex badlight))))
    ♦ m36 kolper make agent to study always at libraries.
(m36!
(object
(m35
(object
(m34
(act (m33
(action
(m32 (object (m30 (location b2) (object (m29 (lex
study))))))
(property (m31 (lex virtually always))))))
(agent b1))))))
(m44! m39 m36!)

```

The possible properties were the only information provided by the sentence. However, this information was released in the form of complex nodes, that noun algorithm cannot purify.

Node m44 and m36 could be considering an example of sentence core concept show by Werner [5]. This node is part of the sentence representation. On the other hand m39 provide new information based on the background knowledge.



## **5 Further Study and Research**

In terms of the information provided by the studied sentence, more background knowledge should be added to try to extract more information. However, according the survey the information extractable from the sentence is very limited.

In terms of the word 'kolper' the inclusion the other four sentences should give us a more specific definition of 'kolper'. This task could be performed in two ways:

- ◆ Sentence by sentence, obtaining an specific incremental concept.
- ◆ All at once

Standardized procedures to collect background information should be developed. Lacking of an uniform form of collecting background knowledge for CASSIE, would make the results less general.

## 6 References

1. van Daalen-Kapteijns, M.M., & Elshout-Mohr, M. (1981), "[The Acquisition of Word Meanings as a Cognitive Learning Process](#)", *Journal of Verbal Learning and Verbal Behavior* 20: 386-399.
2. Nagy, William E., & Anderson, Richard C. (1984), "[How Many Words Are There in Printed School English?](#)", *Reading Research Quarterly* 19(3, Spring): 304-330.
3. [Sternberg, Robert J.](#), & Powell, Janet S. (1983), "[Comprehending Verbal Comprehension](#)", *American Psychologist* 38: 878-893.
4. [Shapiro, Stuart C.](#), & [Rapaport, William J.](#) (1987), "[SNePS Considered as a Fully Intensional Propositional Semantic Network](#)", in [Nick Cercone](#) & [Gordon McCalla](#) (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge* (New York: [Springer-Verlag](#)): 262-315.
5. Werner, Heinz, & Kaplan, Edith (1950), "[Development of Word Meaning through Verbal Context: An Experimental Study](#)", *Journal of Psychology* 29: 251-257.
6. • [Beck, Isabel L.](#); [McKeown, Margaret G.](#); & McCaslin, Ellen S. (1983), "[Vocabulary Development: All Contexts Are Not Created Equal](#)", *Elementary School Journal* 83(3): 177-181.