State University of New York at Buffalo
Computer Science Department

CSE-663
Advanced Knowledge Representation & Reasoning
Instructor: Dr. William J. Rapaport

Contextual Vocabulary Acquisition Programming Project

Pondokkie

Jesper Dybdahl Hede
Person♯: 35964041

Fall 2008

Abstract

The Contextual Vocabulary Acquisition project is an attempt to computerize the everyday human task of interpreting an unknown word and find a meaning of the word. When a human person come across an unknown word, he or she will try to use information from the word's context along with useful prior knowledge, to find the meaning of the word. In the CVA project this task is given to a computer, by the use of the knowledge representation and reasoning tool SNePS. SNePS is an application capable of representing information in a network, and then do inference reasoning on that network.

My work in this project has been to find the meaning of the word *pondokkie*, from the sentence: *She had a **pondokkie** down in the Low Veld, on the Crocodile River, where she went sometimes in the winter to stay for a week or two on her own.* My work was based on a number of human interviews, where the interviewees where given the same task as I later would give to SNePS' cognitive agent *Cassie*. The thought process lead to the formation of a number of "prior knowledge rules" which was represented in SNePS along with the sentence itself. The result was a correct, although simplified, meaning of the word *pondokkie*. Future work would include creating a representation in greater detail as well as using other sentences where *pondokkie* is used. More initial information could lead to a more detailed result.

Contents

1 Introduction to CVA project

Contextual Vocabulary Acquisition is the science of learning the meaning of unknown words or phrases by examining the context where the word or sentence appear. A person reading, or hearing, an unknown word will try to interpret the context surrounding the word and combine it with any useful prior knowledge that the person might have, in order to learn the meaning of the word. It is believed that humans learn most of their vocabulary in this manner, which makes it an interesting field for Artificial Intelligence scholars to duplicate in computers.

This project is centered around a simple implementation of an artificial knowledge base; finding the meaning of a single unknown word is the goal. The knowledge base is implemented in SNePS[1] and the basic knowledge and inference rules are based on human interviews done prior to the implementation phase. The unknown word is given in a context. In this project the word as part of a single sentence found in a short story.

The basic knowledge is based on the human interviews. The interviews were conducted by a single interviewer and a single interviewee. The interviewee were given the sentence on a piece of paper. The unknown word were clearly marked and all other uncertainties were explained to the interviewee[2]; only the unknown word could be *unknown*. The interviewee was then asked to interpret the sentence and try to give a definition of the word's meaning. It is important to note that the exact meaning of the word was not the goal, but a description of what the word meant was.

As an example, given the word **cottage** in a context (assuming the word "cottage" is unknown), the interviewee might respond "place to stay and sleep" as a good response; finding the exact meaning "house" is not important.

The interviewee was asked to verbalize his/her thought process which the interviewer noted down during the interview. The interviews were meant to examine what background information and logical rules the interviewees used in order to find the meaning of the unknown word. Background information might be things like "everyone has a mother" and a logical rule might be something like "my mother's mother is my grandmother". These are the things that every ordinary human knows, but a computer does not. The goal of this project is to implement knowledge and rules into a computer, ie. using SNePS, and make the computer do the same reasoning on the knowledge as a human, ie. the interviewees, would do in order to find the meaning of the unknown word.

[1] SNePS is explained in section 1.1.

[2] In the sentence used in this case there were few uncertainties. Any explanation was given verbally in a neutral manner, so not to infer any meaning/belief to the interviewee.

1.1 SNePS

SNePS (Semantic Network Processing System) is a program used to represent and reason about knowledge representation networks. SNePS was originally developed by Stuart C. Shapiro at the State University of New York at Buffalo. SNePS can be used in different ways, with different components; in this project the SNePS representation was created using the SNePS User Language, SNePSUL, and reasoning was done using the standard cognitive agent known as *"Cassie"*.

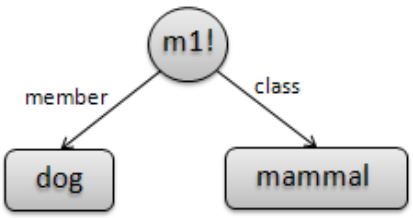
On a more practical level, the basic knowledge and rules are written using SNePSUL in a single text file. The file should contain the basic knowledge and rules, which is to be evaluated by *Cassie*. Running the file should then lead to *Cassie* giving a useful definition of the unknown word.

When entering SNePSUL commands in SNePS and semantic network gets build. The network contains the basic information as well as any inference rules. A graphical component in SNePS is capable of giving a graphical representation to the programmer. Information is entered into SNePS using various constructs called "case frames". Case frames is a way to connect information, in a network-type way. Using the standard case frames enables *Cassie* to do inference, ie. interpret, on the information and rules in the network.

A basic example is the representation of the knowledge

a dog is a mammal

which in SNePS would be represented in a member-class case frame, ie. dog is a member of the class "mammal"[3]:

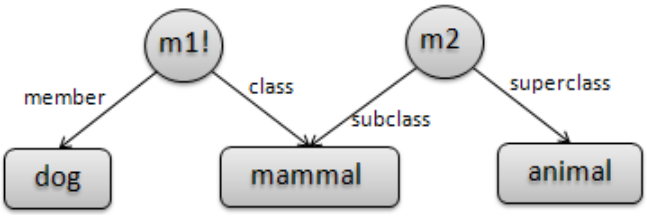


Background knowledge could be the knowledge that

a mammal is an animal

which in SNePS could be represented as a subclass-superclass case frame, stating that *mammal* is a subclass of the superclass *animal*. In SNePS this would give the network:

[3]The actual SNePS constructs are a bit more complicated, but the simplified version is shown here. The understanding of SNePS' inner network should become clear from these examples.



By inference a human would know that these two bits of information leads to a dog also being a member of the class *animal*. In the figure one would follow the arrows, and the connection between the classes are obvious to see. In SNePS we would need a antecedent-consequence rule, which in pseudo code would be:

```
IF X is-member-of Y AND Y is-subclass-of Z
THEN X is-member-of Z
```

The relations `is-member-of` and `is-subclass-of` are meant to be intuitively understood relations that will require not further explanation.

2 Problem in hand: ”Pondokkie”

For this project I chose a single sentence with an unknown word. The sentence I chose is: *She had a **pondokkie** down in the Low Veld, on the Crocodile River, where she went sometimes in the winter to stay for a week or two on her own.*
The unknown word is ”pondokkie”, written in bold face. The rest of the sentence is plain English, besides the two specific names ”the Low Veld” and ”the Crocodile River”. Explaining that these are names of specific places has been sufficient information for the interviewees to read the sentence.

The first step in getting the sentence entered into SNePS was to split the sentence up into smaller parts that would fit better in the standard case frames. I split the sentence into the following four smaller sentences:

- She has a pondokkie.
- The pondokkie is at ”Crocodile River”.
- The pondokkie is at ”Low Veld”.
- She stays at pondokkie in winter for a week or two.

The implementation of these sentences in SNePS will be discussed in section 3.2.

3 Results

3.1 Human Protocols

In my initial research I interviewed 3 people to get information about how humans would approach the problem of finding the meaning of the word *pondokkie*. The interviews was conducted individually by giving the interviewee a piece of paper with the sentence written on it. Any uncertainties about the sentence, not concerning the unknown word, was answered by me verbally. The only thing that the interviewees had a problem with was the two place names "Low Veld" and "Crocodile River", which I explained was names of specific places. I made a point out of not telling where the places where geographically or any specifics of that nature. I only stated that "Crocodile River" was a river and that "Low Veld" was a flat grassy area. Revealing details about climate, continent, country, or similar could have lead the interviewees which was to be avoided.

Having understood the rest of the sentence, but not the unknown word, the interviewees was asked to try and find a meaning for the word. While doing this they was asked to verbalize their thoughts and explain how their thoughts lead them in a given direction and which conclusions they drew on the way. For each statement the interviewee gave he/she was asked to explain why he/she thought of it, and which part of the sentence lead to that point.

The three interviewees gave quite similar interpretations and they all succeeded in finding a somewhat correct meaning of the word *pondokkie*. The notes I wrote during the interviews were short sentences or single words, that I later interpreted on. The notes are:

Ryan
Some place to live or stay. Dwelling. Stay overnight. Stay for some time. Tent. By a river.

Kirsten
It sounds Indian, maybe a boat. But no, because it is winter (the river would be frozen). But it does not have to be a place where there is snow.

Tipi/housing.

Go on water/house. Because of winter, something there is on the ice.

Cottage, she can live there (because of "a week or two").

She uses it. Music instrument, because she does not necessarily have to live *in* it. Something for an activity.

Jimmy
It is in a specific place. It is a thing. It is warm, since she can stay there during the winter. So indoors.

Does the stay in the pondokkie or near the pondokkie? "where she went" could be the pondokkie or it could be the Low Veld.

Thinks it is something she can sleep in. A small house.

From these interviews I extracted the most important points that could be used in the SNePS implementation.

- The location seems to be in nature. This was not explicitly noted down, but I deduced it from the statements. There is a clear sense that this is outside of a normal populated area. The interviewees based this on the two names (Crocodile River and Low Veld) which are specific names, but nature objects (river and grassy plain) are parts of the names.
- It is a place that would be cold. This is mainly based on the "in the winter" part of the sentence. The interviewees did not seem to consider that the place could be in a warm part of the world, where the winter is not as cold as we have in our part of the world.
- She stays overnight. This is based on "for a week or two" part of the sentence.
- Since she stays overnight in nature, then she would need some kind of shelter to stay in.
- Since the outside is cold this shelter must be warm or keeping her warm. This tells us that a pondokkie is something man-made or contains something man-made.

This gives a pretty good description of the word pondokkie. From the interviewees' statements we can say that a pondokkie is *a warm man-made shelter in which a person can stay overnight protected from nature.* This meaning of pondokkie is derived from their explanations, but the interviewees also gave more concrete guesses for the meaning of the word. They were not asked to do this, but it seemed to be natural for them to try a give a precise definition of the word, ie. trying to translate the unknown word pondokkie into a well-known word. Ryan guessed "dwelling" or "tent", Kirsten guessed "boat", "tipi" or "house"[4], and finally Jimmy guessed "a small house". These are all man-made objects and all of them can be said to be places where humans go into as being protected from the outside. Kirsten's guess of a boat is the only object not land based, but she quickly left the idea because of the sentence stating it was winter.

[4]Kirsten ended up going in a totally different direction with her guesses and the meaning of the word. The meaning of a music instrument is strange and Kirsten could not directly explain what part of the sentence gave her the idea. I write it off as "over thinking" the sentence. I had a sense that she tried to satisfy my questions, since I did not give any positive response to her first, and correct, interpretation. I think she tried to "get it right" and changed her guess since I did not acknowledge her first guess as being correct.

The guesses are fairly close to the correct meaning of pondokkie being a summer house/temporary house where one can stay for a given amount of time, without the pondokkie being a permanent home.

3.2 SNePS Implementation

My sentence along with the rules I derived from the human interviews were entered into SNePS using a `.demo` file. The `.demo` file makes it easy to run the rules in a single operation, and makes it possible to save the work in a convenient way. In the `.demo` file the rules come before the sentence itself, but for simplicity the sentence will be discussed first.

3.2.1 The Sentence

To recap, the sentence was: *She had a **pondokkie** down in the Low Veld, on the Crocodile River, where she went sometimes in the winter to stay for a week or two on her own.* The sentence was split up into smaller sentences, each of which were separately entered into the `.demo` file using SNePSUL. The first sentence was:

- She has a pondokkie.

In SNePSUL this took 3 statements:

```
1  ; First: Create object PONDOKKIE
2  (describe (add member #pondokkie class (build lex "PONDOKKIE")))
3
4  ; Second: Create object SHE
5  (describe (add object #she propername "SHE"))
6
7  ; Third: SHE owns this PONDOKKIE
8  (describe (add possessor *she
9                 object *pondokkie
10                rel (find lex "pondokkie")))
```

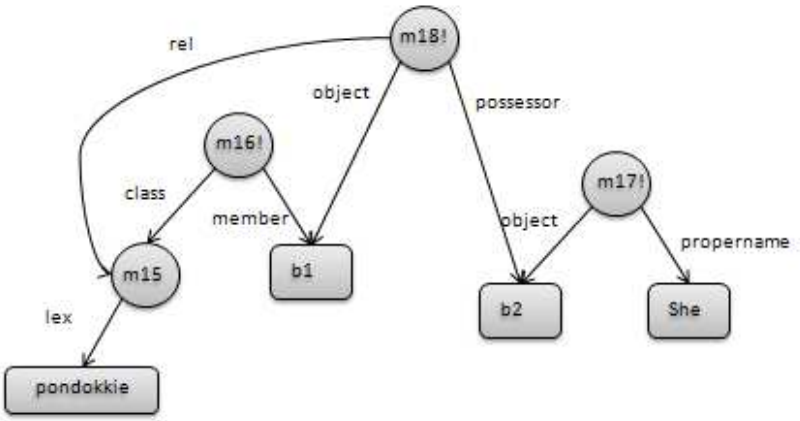
Lines starting with a semicolon (`;`) are comments.

In line 2 an object is created by using the ?? character (`#`), and it is set to be the member of the class with the lexical name "pondokkie".

In line 5 an object representing "she" is created. Since "she" is a specific object, the member-class case frame is not used. In stead the object is created having a propername, here "SHE". This is done since we assume "She" represent a female person, but we do not know anymore about her.

Finally in the third statement, lines 8-10, the sentence is put together, telling SNePS that the object SHE own the object PONDOKKIE. This is done using the object-possessor-rel case frame. The case frame states that the object is the possessor's rel. In this case is states that the object previously created for the specific pondokkie is the object "SHE"'s

pondokkie. In other words, the object she owns the object pondokkie in the sense that it has the type pondokkie. The SNePS network representing the first sentence is:



Second sentence:

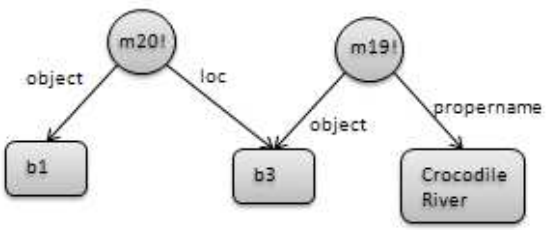
- The pondokkie is at Crocodile River.

SNePSUL:

```
1 ; There is a place called CROCODILE RIVER
2 (describe (add object #criver propername "CROCODILE\ RIVER"))
3
4 ; PONDOKKIE is at CROCODILE RIVER
5 (describe (add object *pondokkie loc *criver))
```

In line 2 an object representing the Crocodile River is created. Again, since it is a specific thing in the real world the object-propername case frame is used instead of the member-class case frame.

In line 4 a spacial case frame is used to state that the pondokkie is located by the Crocodile River. There are a number of standard spacial case frames, but since the sentence does not state any specifics about the location this basic one is used. The sentence states that the pondokkie is at Crocodile River, which can have many different meanings. However, the interviewees did not seem to question this further, so the knowledge that the pondokkie is *at* Crocodile River seems sufficient.



The network representation of this sentence is simple, and is equivalent for the two location sentences (this plus the next one).

Third sentence:

- The pondokkie is at "Low Veld".

SNePSUL:

```
1 ; There is a place called LOW VELD
2 (describe (add object #lveld propername "LOW\ VELD"))
3
4 (describe (add object *pondokkie loc *lveld))
```

This is equivalent to the second sentence. An object representing the Low Veld is created, and the pondokkie is stated to be *at* it. One could argue that having both location statements is not useful, since adding the second one does not give any additional information. But since the two places are members of different entities in nature, there can be said to be more information in the sentences than just the names and the fact that they are part of nature[5]. The interviewees reacted to the names, as they used their background knowledge to derive information from the names. They had a knowledge about how place names are usually linked to some specific characteristic about the place. Therefore both sentences are used.

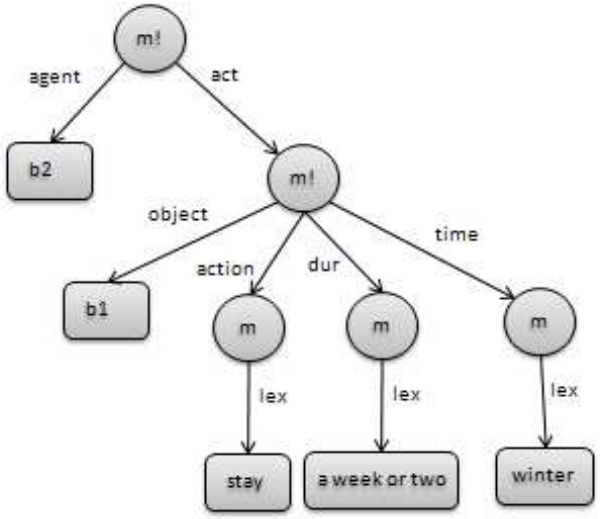
Fourth sentence:

- She stays at pondokkie in winter for a week or two.

SNePSUL:

```
1 ; SENTENSE 4: She stays at pondokkie in winter for a week or two.
2
3 (describe (add agent *she
4                act (build action (build lex "stay")
5                          object *pondokkie
6                          dur (build lex "a\ week\ or\ two")
7                          time (build lex "winter"))))
```

[5]Rules for these sentences are discussed in section 3.2.2.



The sentence is represented using two case frames; one wrapped into the other. The outer case frame is the standard agent-act case frame. This represents that an agent (here it is "SHE") performs an action (ie. does an "act"). The act is to stay at the pondokkie for a week or two in winter. An action-dur-time case frame was chosen and used with a little alteration. First, in line 5 the action is described by its lexical string "stay", ie. the act is to stay. "dur" is the duration of the action and it is simply the lexical string "a week or two". One could argue that a numerical value would be better, but as the interviewees did not interpret it into a specific number of days, neither will SNePS. The time (line 7) is understood to be the time of year and it is set to be winter. The case frame can be used in many other ways, where time could be something else than a season[6].

Finally the alteration I did to the case frame was to introduce the `object` line. The `object` is set to be the object that the action is performed on. In line 5 the object of "SHE's" action is the pondokkie. Not just any pondokkie, but the same pondokkie that was created when representing the first part-sentence.

So the full representation of sentence four is: "SHE" performs an act, which is a action of staying performed on the pondokkie, with a duration of a week or two in the time of winter. It is not quite a simple to read as the *human* sentence, but the meanings can be said to be equivalent.

3.2.2 Background Knowledge

Background knowledge used to interpret the sentence is entered into SNePS as rules. The rules are based on the interviews. After all interviews were completed, a common interpretation of the interviewees' thought process was noted down as rules that SNePS should be able to use. This lead to the following list of rules:

[6]It could for example be "evening" or "the 1960s" when used in other contexts.

1. "Crocodile River" is a part of nature, ie. not a populated area.
2. "Low Veld" is a part of nature, ie. not a populated area.
3. "A week or two" means it is overnight, ie. more than one day.
4. If one stays more than one day in a place in nature, one would need to stay in a shelter of some kind.
5. If one stays more than one day in a place during winter, one would need to stay in a warm place.
6. A warm shelter is some kind of house.

The actual implementation does not take the rules individually. Some of them are gathered and some are used in several rules. I will go through each of them and explain the purpose.

3.2.3 Rule: Low Veld and Crocodile River are places in nature.

```
1 ; Low Veld and Crocodile River are places in nature.
2 (describe (assert forall ($x)
3    ant (build object *x loc *lveld)
4    ant (build object *x loc *criver)
5    cq (build object *x loc (build lex "nature"))))
```

The SNePSUL rule is a conjunction of rules 1 and 2. Using two `ant` has an "OR" meaning, so if one of the antecedents are true, then the consequence is true. The rule simple states that if an object x has the location either "Low Veld" or "Crocodile River", then x also has the location "nature". The objects referenced for "Low Veld" and "Crocodile River" are the same objects created when the sentence was represented.

3.2.4 Rule: If an action has duration "a week or two" then it also has the duration "overnight"

```
1 ; If an action has duration "A WEEK OR TWO"
2 ; then it also has the duration "OVERNIGHT"
3 (describe (assert forall ($a $v $w $u)
4    ant (build agent *a act (build action *v object *w
5          dur (build lex "a\ week\ or\ two") time *u))
6    cq   (build agent *a act (build action *v object *w
7          dur (build lex "overnight") time *u))))
```

This SNePSUL rule is equivalent to rule 3. What is basically says is that because "she" stays for a week or two, she would have to stay overnight. What I wanted to state was that any action that takes the duration of a week or two it would also take the duration of overnight. Unfortunately, SNePS' inference engine is not able to generalize like needed in this case. Cassie is able to minimize the sentence, and cut away unused parts.

Here the consequence case frame would consist only of the parts that are explicitly stated in the rule. So if the rule only contains the duration, then it will not be matched with a case frame containing other parts, like the object and the time.

Therefore I had to create the rule containing all the parts that the original case frame contains in order to get Cassie to infer correctly as I wanted.

3.2.5 Rule: If x stays at place y overnight, and y is in nature, then y is a shelter of some kind.

```
1  ; If x stays at place y overnight, and y is in nature,
2  ; then y is a shelter of some kind.
3  (describe (assert forall ($x $y)
4     &ant
5     (
6           build agent *x act (build action (build lex "stay") object *y
7           dur (build lex "overnight"))
8     )
9     &ant (build object *y loc (build lex "nature"))
10    cq (build member *y class (build lex "shelter"))))
```

The is the representation of rule 4. The SNePSUL rule consists of two antecedents that both have to be true for the consequence to be true. Therefore two `&and` were used, where the &-sign tells Cassie that the antecedents should be interpreted as a conjunction.

The first antecedent states that an agent-act case frame must consist of the action of staying overnight in a given place y. The second one states that this y must have a location in nature. The consequence is that y is states to be a member of the class of shelters.

3.2.6 Rule: If x stays at place y overnight in winter, then y is a warm place.

```
1  ; If x stays at place y overnight in winter,
2  ; then y is a warm place.
3  (describe (assert forall ($x $y)
4     ant
5     (
6           build agent *x act (build action (build lex "stay") object *y
7           dur (build lex "overnight")
8           time (build lex "winter"))
9     )
10    cq (build object *y property (build lex "warm"))))
```

This SNePSUL rule is very similar to the previous one, and it is the representation of rule 5 from the human protocol. The antecedent is searching for the agent-act case frame that states that an agent performs an act of staying at a place y, for the duration "overnight" and in time "winter". The consequence is that this place y then is said to

have the property of being warm. This is really a simplification of the notion that the place where the agent stays has the property of being able to keep the agent from freezing. To say that an object is "warm" is a bit ambiguous, since being warm can have many different meanings. However, in this very limited world we examine here, being warm had a specific meaning for the interviewees, thus I kept the notion of "warm" in the representation without further explanation.

3.2.7 Rule: A warm shelter is some kind of house.

```
1 ; A warm shelter is some kind of house.
2 (describe (assert forall ($x)
3                 &ant (build member *x class (build lex "shelter"))
4                 &ant(build object *x property (build lex "warm"))
5                 cq (build member *x class (build lex "house"))))
```

This is rule 6, which joins the results of two of the previous rules. It is the final step in the process, stating that if a member x of the class of shelters has the property of being warm, then this x also a member of the class of houses. As stated earlier, a house is a pretty good explanation of the word "pondokkie".

Once again two antecedents are used in conjunction, where both have to be true for the consequence to be true.

3.3 Running the SNePS Demo File

The demo file "pondokkie.demo" can be seen in full in appendix A. The output from SNePS when running the demo file can be seen in appendix B.

As we can see from the output, Casse gives the following definition of the word *pondokkie*:

```
1 ; Ask Cassie what "pondokkie" means:
2 ^(
3 --> defineNoun "pondokkie")
4  Definition of pondokkie:
5  Possible Class Inclusions: shelter, house,
6  Actions performed on a pondokkie: b2 stay,
7  Possible Properties: warm,
8 nil
```

In other words, Cassie defines a pondokkie as a shelter or a house; an agent can perform the action of staying at the pondokkie, and the pondokkie can have the property of being warm. The end result can be seen as being the "house" definition. "Shelter" is a generalization of house, and gives less information to the reader. The ability to stay at the house, and the property of being warm is less interesting and only used by Cassie during the inference process. To a human reader these would be logical conclusions of having a house.

Comparing Cassie's definition to the SNePSUL rules there is a clear correlation. Each rule gives a partly conclusion, and Cassie states the ones related to the pondokkie. Intermediate conclusions about the winter being cold, the river being in nature, etc. are not stated when Cassie is asked about the pondokkie, for obvious reasons.

4 Further Steps

A few immediate next steps could be taken on my work. The first one is that a property of the pondokkie, the house, seems to be that it is only used once in a while, on a temporary basis. "She" only stays there for a week or two, which suggests that the house is a small cottage or summer house[7].

Another step to take is to get a better definition of the location inserted into SNePS. "On the river" and "in the Low Veld" gives the information of being in the nature/countryside, but the SNePSUL rule stating that these places simply are part of nature is a bit vague. "On the river" could suggest that we are talking about a boat, and it could give information about the use of the pondokkie. From this single sentence there are few indications that it would matter much, but it might still lead to new results or more detailed results.

5 Future Work

This things I feel is missing in the current work is in the details. At the moment we get the result that the pondokkie is a house, a shelter in nature, and a place where a person can go live for an extended amount of time. The information I miss is information about the use of the house, any special architectural features, the surroundings, etc.

The obvious way to get more information is to find more sentences where the work "pondokkie" is used. More sentences should lead to new human interviews and finally an expansion on the SNePS implementation. Additional sentences of interest can be found in appendix C.

[7] In a lack of a better word, since we know she goes there in winter time.

References

[1] Gordimer, Nadine , *The Life of the Imagination,* in Charles Bohmer & Lyman Grant (eds.), *Short Fiction: Classic and Contemporary:* 519, Sixth Edition, Pearson Education, 2006.

[2] Brachman, Ronald J. & Levesque, Hector J., *Knowledge Representation and Reasoning,* Morgan Kaufmann Publishers, 2004.

[3] Rapaport, William J. & Ehrlich, Karen, *A Computational Theory of Vocabulary Acquisition,* Department of Computer Science and Center for Cognitive Science, The State University of New York at Buffalo, 2000.

[4] Shapiro, Stuart C. & Rapaport, William J., *SNePS Considered as a Fully Intentional Propositional Semantic Network,* in Nick Cercone & Gordon McCella (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge:* 262-315, Springer-Verlag, 1987.

[5] Shapiro, Stuart C. & Rapaport, William J., *An Introduction To a Computational Reader of Narratives,* in Judith F. Duchan, Gail A. Bruder & Lynne E. Hewitt (eds.), *Deixis In Narrative: A Cognitive Science Perspective:* 79-105, The State University of New York at Buffalo, 1995.

[6] Almeida, Michael J., *Time in Narratives,* in Judith F. Duchan, Gail A. Bruder & Lynne E. Hewitt (eds.), *Deixis In Narrative: A Cognitive Science Perspective:* 159-189, The State University of New York at Buffalo, 1995.

[7] Martins, João Pavão, *Knowledge Representation,* Departamento de Engenharia Informática, Universidade Técnica de Lisboa, 2001.

[8] Shapiro, Stuart C. & The SNePS Implementation Group, *SNePS 2.7 USER'S MANUAL,* Department of Computer Science and Engineering, The State University of New York, 2008.

[9] Shapiro, Stuart C. et al., *A Dictionary of SNePS Case Frames,* Department of Computer Science, The State University of New York, 1996.

APPENDIX

A Demo File

```
1 ; ==========================================================================
2 ; FILENAME:       pondokkie.demo
3 ; DATE:     11/01/08
4 ; PROGRAMMER:     Jesper Dybdahl Hede
5
6 ;; NOTE TO PROGRAMMER:  GLOBALLY REPLACE "WORD" BY YOUR WORD,
7 ;;        "DATE" BY TODAY'S DATE,
8 ;;        "YOUR_NAME" BY YOUR NAME,
9 ;;        AND THEN DELETE THIS COMMENT (TO MAKE IT LOOK GOOD!)
10
11 ;; this template version:        snepsul-template.demo-20061005.txt
12
13 ; Lines beginning with a semi-colon are comments.
14 ; Lines beginning with "^" are Lisp commands.
15 ; All other lines are SNePSUL commands.
16 ;
17 ; To use this file: run SNePS; at the SNePS prompt (*), type:
18 ;
19 ;        (demo "pondokkie.demo" :av)
20 ;
21 ; Make sure all necessary files are in the current working directory
22 ; or else use full path names.
23 ; ==========================================================================
24
25 ; Turn off inference tracing.
26 ; This is optional; if tracing is desired, then delete this.
27 ^(setq snip:*infertrace* nil)
28
29 ; Load the appropriate definition algorithm:
30 ^(load "/projects/rapaport/CVA/STN2/defun_noun.cl")
31
32 ; Clear the SNePS network:
33 (resetnet)
34
35 ; load all pre-defined relations:
36 (intext "/projects/rapaport/CVA/STN2/demos/rels")
37
38 ; Definitions:
39 (define
40         loc propername dur
41 )
42
43 ; load all pre-defined path definitions:
44 (intext "/projects/rapaport/CVA/mkb3.CVA/paths/paths")
45
46 ; BACKGROUND KNOWLEDGE:
47 ; ========================
48
49 ; Low Veld and Crocodile River are places in nature.
50 (describe (assert forall ($x)
51    ant (build object *x loc *lveld)
```

```
  ant (build object *x loc *criver)
  cq (build object *x loc (build lex "nature"))))

; If an action has duration "A WEEK OR TWO"
; then it also has the duration "OVERNIGHT"
(describe (assert forall ($a $v $w $u)
  ant (build agent *a act (build action *v object *w dur (build lex "a\ week\ or\ two") time *u))
  cq  (build agent *a act (build action *v object *w dur (build lex "overnight") time *u))))

; If x stays at place y overnight, and y is in nature,
; then y is a shelter of some kind.
(describe (assert forall ($x $y)
  &ant
  (
      build agent *x act (build action (build lex "stay") object *y
      dur (build lex "overnight"))
  )
  &ant (build object *y loc (build lex "nature"))
  cq (build member *y class (build lex "shelter"))))

; If x stays at place y overnight in winter,
; then y is a warm place.
(describe (assert forall ($x $y)
  ant
  (
      build agent *x act (build action (build lex "stay") object *y
      dur (build lex "overnight")
      time (build lex "winter"))
  )
  cq (build object *y property (build lex "warm"))))

; A warm shelter is some kind of house.
(describe (assert forall ($x)
  &ant (build member *x class (build lex "shelter"))
  &ant(build object *x property (build lex "warm"))
  cq (build member *x class (build lex "house"))))

; CASSIE READS THE PASSAGE:
; ==========================

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; REPRESENTATION OF THE SENTENSE
;
; She had a pondokkie down in the Low Veld, on
; the Crocodile River, where she went sometimes
; in the winter to stay for a week or two.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; SENTENSE 1: She has a pondokkie

; First: Something is a pondokkie
(describe (add member #pondokkie class (build lex "pondokkie")))

; Second: Create this "She"
(describe (add object #she propername "She"))
```

```
; Third: She owns this pondokkie
(describe (add possessor *she
            object *pondokkie
            rel (find lex "pondokkie")))

; SENTENSE 2: The pondokkie is at "Crocodile River"

; There is a place called "Crocodile River"
(describe (add object #criver propername "Crocodile\ River"))

(describe (add object *pondokkie loc *criver))

; SENTENSE 3: The pondokkie is at "Low Veld"

; There is a place called "Low Veld"
(describe (add object #lveld propername "Low\ Veld"))

(describe (add object *pondokkie loc *lveld))

; SENTENSE 4: She stays at pondokkie in winter for a week or two.
(describe (add agent *she
    act (build action (build lex "stay") object *pondokkie
    dur (build lex "a\ week\ or\ two")
    time (build lex "winter"))))

; Ask Cassie what "pondokkie" means:
^(defineNoun "pondokkie")
```

B Transcript of Demo

```

=============================================================
Starting image '/util/acl/composer'
  with no arguments
  in directory '/home/unmgrad/jdhede/'
  on machine 'localhost'.

;;; Installing locale patch, version 1.
International Allegro CL Enterprise Edition
8.1 [Linux (x86)] (Oct 27, 2008 11:52)
Copyright (C) 1985-2007, Franz Inc., Oakland, CA, USA.  All Rights Reserved.

This development copy of Allegro CL is licensed to:
   [4549] University at Buffalo

;; Optimization settings: safety 1, space 1, speed 1, debug 2.
;; For a complete description of all compiler switches given the current
;; optimization settings evaluate (explain-compiler-settings).
;;---
;; Current reader case mode: :case-sensitive-lower
cl-user(1): ;; Setting (stream-external-format *terminal-io*) to :utf-8.
cl-user(2): :ld /projects/snwiz/bin/sneps
; Loading /projects/snwiz/bin/sneps.lisp
;;; Installing jlinker patch, version 1.
;;; Installing regexp2-s patch, version 1.
Loading system SNePS...10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
```

```
SNePS-2.7 [PL:1 2008/02/12 17:19:45] loaded.
Type '(sneps)' or '(snepslog)' to get started.
cl-user(3): (sneps)

   Welcome to SNePS-2.7 [PL:1 2008/02/12 17:19:45]

Copyright (C) 1984--2007 by Research Foundation of
State University of New York. SNePS comes with ABSOLUTELY NO WARRANTY!
Type '(copyright)' for detailed copyright information.
Type '(demo)' for a list of example applications.

   12/8/2008 11:20:54

* (demo "pondokkie.demo")

File /home/unmgrad/jdhede/pondokkie.demo is now the source of input.

 CPU time : 0.01

* ; ============================================================================
; FILENAME:      pondokkie.demo
; DATE:    11/01/08
; PROGRAMMER:    Jesper Dybdahl Hede

;; NOTE TO PROGRAMMER:  GLOBALLY REPLACE "WORD" BY YOUR WORD,
;;       "DATE" BY TODAY'S DATE,
;;       "YOUR_NAME" BY YOUR NAME,
;;       AND THEN DELETE THIS COMMENT (TO MAKE IT LOOK GOOD!)

;; this template version:       snepsul-template.demo-20061005.txt

; Lines beginning with a semi-colon are comments.
; Lines beginning with "^" are Lisp commands.
; All other lines are SNePSUL commands.
;
; To use this file: run SNePS; at the SNePS prompt (*), type:
;
;       (demo "pondokkie.demo" :av)
;
; Make sure all necessary files are in the current working directory
; or else use full path names.
; ============================================================================

; Turn off inference tracing.
; This is optional; if tracing is desired, then delete this.
^(
--> setq snip:*infertrace* nil)
nil

 CPU time : 0.00

*
; Load the appropriate definition algorithm:
^(
--> load "/projects/rapaport/CVA/STN2/defun_noun.cl")
; Loading /projects/rapaport/CVA/STN2/defun_noun.cl
```

```
t

 CPU time : 0.02

*
; Clear the SNePS network:
(resetnet)

Net reset - Relations and paths are still defined

 CPU time : 0.00

*
; load all pre-defined relations:
(intext "/projects/rapaport/CVA/STN2/demos/rels")
Loading file /projects/rapaport/CVA/STN2/demos/rels.

 CPU time : 0.00

*
; Definitions:
(define
        loc propername dur
)

(loc propername dur)

 CPU time : 0.00

*
; load all pre-defined path definitions:
(intext "/projects/rapaport/CVA/mkb3.CVA/paths/paths")
Loading file /projects/rapaport/CVA/mkb3.CVA/paths/paths.
before implied by the path (compose before (kstar (compose after- ! before)))
before- implied by the path (compose (kstar (compose before- ! after)) before-)
after implied by the path (compose after (kstar (compose before- ! after)))
after- implied by the path (compose (kstar (compose after- ! before)) after-)
sub1 implied by the path (compose object1- superclass- ! subclass superclass-
                          ! subclass)
sub1- implied by the path (compose subclass- ! superclass subclass- !
                           superclass object1)
super1 implied by the path (compose superclass subclass- ! superclass object1-
                            ! object2)
super1- implied by the path (compose object2- ! object1 superclass- ! subclass
                             superclass-)
superclass implied by the path (or superclass super1)
superclass- implied by the path (or superclass- super1-)

 CPU time : 0.00

*
; BACKGROUND KNOWLEDGE:
; ====================

; Low Veld and Crocodile River are places in nature.
(describe (assert forall ($x)
```

```
139    ant (build object *x loc *lveld)
140    ant (build object *x loc *criver)
141    cq (build object *x loc (build lex "nature"))))
142
143 (m2! (forall v1) (ant (p1 (object v1)))
144  (cq (p2 (loc (m1 (lex nature))) (object v1))))
145
146 (m2!)
147
148  CPU time : 0.00
149
150 *
151 ; If an action has duration "A WEEK OR TWO"
152 ; then it also has the duration "OVERNIGHT"
153 (describe (assert forall ($a $v $w $u)
154    ant (build agent *a act (build action *v object *w dur (build lex "a\ week\ or\ two") time *u))
155    cq   (build agent *a act (build action *v object *w dur (build lex "overnight") time *u))))
156
157 (m5! (forall v5 v4 v3 v2)
158  (ant
159   (p4
160    (act (p3 (action v3) (dur (m3 (lex a week or two))) (object v4) (time v5)))
161    (agent v2)))
162  (cq
163   (p6 (act (p5 (action v3) (dur (m4 (lex overnight))) (object v4) (time v5)))
164    (agent v2))))
165
166 (m5!)
167
168  CPU time : 0.00
169
170 *
171 ; If x stays at place y overnight, and y is in nature,
172 ; then y is a shelter of some kind.
173 (describe (assert forall ($x $y)
174    &ant
175    (
176         build agent *x act (build action (build lex "stay") object *y
177         dur (build lex "overnight"))
178    )
179    &ant (build object *y loc (build lex "nature"))
180    cq (build member *y class (build lex "shelter"))))
181
182 (m8! (forall v7 v6)
183  (&ant (p9 (loc (m1 (lex nature))) (object v7))
184   (p8
185    (act (p7 (action (m6 (lex stay))) (dur (m4 (lex overnight))) (object v7)))
186    (agent v6)))
187  (cq (p10 (class (m7 (lex shelter))) (member v7))))
188
189 (m8!)
190
191  CPU time : 0.01
192
193 *
194 ; If x stays at place y overnight in winter,
```

```
; then y is a warm place.
(describe (assert forall ($x $y)
   ant
   (
        build agent *x act (build action (build lex "stay") object *y
        dur (build lex "overnight")
        time (build lex "winter"))
   )
   cq (build object *y property (build lex "warm"))))

(m11! (forall v9 v8)
 (ant
  (p12
   (act (p11 (action (m6 (lex stay))) (dur (m4 (lex overnight))) (object v9)
         (time (m9 (lex winter)))))
   (agent v8)))
 (cq (p13 (object v9) (property (m10 (lex warm))))))

(m11!)

 CPU time : 0.00

*
; A warm shelter is some kind of house.
(describe (assert forall ($x)
   &ant (build member *x class (build lex "shelter"))
   &ant(build object *x property (build lex "warm"))
   cq (build member *x class (build lex "house"))))

(m13! (forall v10)
 (&ant (p15 (object v10) (property (m10 (lex warm))))
  (p14 (class (m7 (lex shelter))) (member v10)))
 (cq (p16 (class (m12 (lex house))) (member v10))))

(m13!)

 CPU time : 0.00

*
; CASSIE READS THE PASSAGE:
; ==========================

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; REPRESENTATION OF THE SENTENSE
;
; She had a pondokkie down in the Low Veld, on
; the Crocodile River, where she went sometimes
; in the winter to stay for a week or two.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; SENTENSE 1: She has a pondokkie

; First: Something is a pondokkie
(describe (add member #pondokkie class (build lex "pondokkie")))

(m15! (class (m14 (lex pondokkie))) (member b1))
```

```
251
252 (m15!)
253
254  CPU time : 0.00
255
256 *
257 ; Second: Create this "She"
258 (describe (add object #she propername "She"))
259
260 (m18! (loc (m1 (lex nature))) (object b2))
261 (m17! (object b2))
262 (m16! (object b2) (propername She))
263
264 (m18! m17! m16!)
265
266  CPU time : 0.00
267
268 *
269 ; Third: She owns this pondokkie
270 (describe (add possessor *she
271             object *pondokkie
272             rel (find lex "pondokkie")))
273
274 (m23! (loc (m1 (lex nature))) (object b1))
275 (m22! (object b1))
276 (m21! (object b1) (possessor b2) (rel (m14 (lex pondokkie))))
277
278 (m23! m22! m21!)
279
280  CPU time : 0.00
281
282 *
283 ; SENTENSE 2: The pondokkie is at "Crocodile River"
284
285 ; There is a place called "Crocodile River"
286 (describe (add object #criver propername "Crocodile\ River"))
287
288 (m28! (loc (m1 (lex nature))) (object b3))
289 (m27! (object b3))
290 (m26! (object b3) (propername Crocodile River))
291
292 (m28! m27! m26!)
293
294  CPU time : 0.00
295
296 *
297 (describe (add object *pondokkie loc *criver))
298
299 (m31! (loc b3) (object b1))
300 (m22! (object b1))
301
302 (m31! m22!)
303
304  CPU time : 0.00
305
306 *
```

```
307 ; SENTENSE 3: The pondokkie is at "Low Veld"
308
309 ; There is a place called "Low Veld"
310 (describe (add object #lveld propername "Low\ Veld"))
311
312 (m34! (loc (m1 (lex nature))) (object b4))
313 (m33! (object b4))
314 (m32! (object b4) (propername Low Veld))
315
316 (m34! m33! m32!)
317
318  CPU time : 0.01
319
320 *
321 (describe (add object *pondokkie loc *lveld))
322
323 (m37! (loc b4) (object b1))
324 (m22! (object b1))
325
326 (m37! m22!)
327
328  CPU time : 0.00
329
330 *
331 ; SENTENSE 4: She stays at pondokkie in winter for a week or two.
332 (describe (add agent *she
333     act (build action (build lex "stay") object *pondokkie
334     dur (build lex "a\ week\ or\ two")
335     time (build lex "winter"))))
336 (m45! (class (m12 (lex house))) (member b1))
337 (m43! (object b1) (property (m10 (lex warm))))
338 (m42! (class (m7 (lex shelter))) (member b1))
339 (m41!
340  (act (m24 (action (m6 (lex stay))) (dur (m4 (lex overnight))) (object b1)))
341  (agent b2))
342 (m40!
343  (act (m25 (action (m6)) (dur (m4)) (object b1) (time (m9 (lex winter)))))
344  (agent b2))
345 (m39!
346  (act (m38 (action (m6)) (dur (m3 (lex a week or two))) (object b1)
347        (time (m9))))
348  (agent b2))
349
350 (m45! m43! m42! m41! m40! m39!)
351
352  CPU time : 0.00
353
354 *
355
356 ; Ask Cassie what "pondokkie" means:
357 ^(
358 --> defineNoun "pondokkie")
359  Definition of pondokkie:
360  Possible Class Inclusions: shelter, house,
361  Actions performed on a pondokkie: b2 stay,
362  Possible Properties: warm,
```

```
363 nil
364
365  CPU time : 0.03
366
367 *
368
369 End of /home/unmgrad/jdhede/pondokkie.demo demonstration.
370
371  CPU time : 0.07
372
373 *
```

C Additional Uses of "Pondokkie"

For future work, these are the most important phrases containing information about the pondokkie. The final one does not mention the word exactly, but from a larger context it becomes apparent that the shack mentioned is her pondokkie.
If these were to be used in future human interviews, I suggest using a larger part of the short story[1].

---

*She had a **pondokkie** down in the Lw Veld, on the Crocodile River, where she went sometimes in the winter to stay for a week or two on her own.*

---

*She and Arthur, her husband, knew it; they got on almost wordlessly well, now and then turned instinctively to each other, and were alone, he with his work and she with her books and her **pondokkie**.*

---

*He took his pipe out of his mouth. "When you're down at the Crocodile," he said - they called her **pondokkie** that because it was on the Crocodile River - "What are you doing there, when you're alone?*
She lay under his gaze; she felt the quality she had for him, awkwardly, as if he had stuck a jewel on her forehead. She didn't answer.
"Reading, eh? Reading and thinking you own thoughts.

That winter she did not go once to the shack. It had become nothing but a shelter where they could have made love, whole nights and days.
