Understanding word “maul” from passage by SNePS

Hansheng Lei
State University of New York at Buffalo,
hlei@cse.buffalo.edu

Abstract

The task of this CVA (Context Vocabulary Acquisition[1]) project is to understand the meaning of word “maul” from a given passage. The report first gives a brief introduction on CVA and SNePS system which is the main tool for CVA project. Then, how to reason the meaning of “maul” from text is shown step by step by applying SNePS.

1. Introduction

The goal of CVA is to deliberately acquire the word meanings form text by reasoning from contextual clues, background knowledge, and hypotheses developed from prior encounters with the word, but without the external sources help such as dictionaries or people [1, 2, 3]. CVA was developed and still is developing mainly by Dr. William Rapaport and Dr. Michael Kibby at State University of New York at Buffalo. It is based on SNePS and Lisp, which is a universal Artificial Intelligence language.

People can guess or learn the meanings of words for contextual passage. Why are they able to do so? If they can “think aloud”, the learning can also be done by computer automatically. Although there is no generally accepted cognitive theory of learning words, AI algorithms for CVA can turn “guess” into "computing". How the contexts operate to indicate the meaning of unknown words can be taught in the classroom to improve students’ reading and comprehension.

My role in the CVA project is to reason the meaning of word “maul” for a passage by SNePS. The following sections will show how to represent the passage in SNePS and how to obtain the meaning of “maul” from the representation.

2. Understanding “maul” by SNePS

The given passage is as follows:

*The heavy **maul** used for pounding in fence posts lay unused at his side.*

Dulin, Kenneth L. (1970), "Using Context Clues in Word Recognition and Comprehension", *The Reading Teacher* 23(5):440-445, 469.

From above passage, we can get the following facts:

1. Maul is heavy.
2. Maul can be used for pounding in the fence posts.
3. Maul lay at his side and was unused.

To represent the facts in SNePS, we need express the facts in the following ways:
1. Object “Maul” has property “Heavy”.
2. 1). “Person” is a member of class “Human”
   2). “Person” use object “Maul” for purpose “pounding” in the fence posts.
3. Object “Maul” lay at the side of “Person” with status “Unused”.

2.1 Think-aloud Protocol

I asked friends (human, of course) to read the passage and tell me the meaning of “maul” by think-aloud. They all think “maul” is a kind of hammer, because it is heavy and used for pounding. Based on this, I think it is necessary to add the following background information to the knowledge base.

If people use X for pounding and X is heavy, then X is a kind of hammer, i.e., X is subclass of super class” hammer”. (Background rule)

3. Presenting Passage in SNePS

1. Background knowledge: if Person use X for purpose "pounding", Person is member of “Human” of course, and X is heavy, then X is a kind of hammers.

```
(define object property agent act action purpose member class lex)
(assert forall $p
     &ant ( (build object *p property (build lex "Heavy"))
                 (build member (build lex "Person") class (build lex "Human"))
               (build agent (build lex "Person") act (build action
                 (build lex "Use") object *p purpose
          (build lex "Pounding")))
            )
      cq (build subclass *p superclass (build lex "Hammer"))
)
```

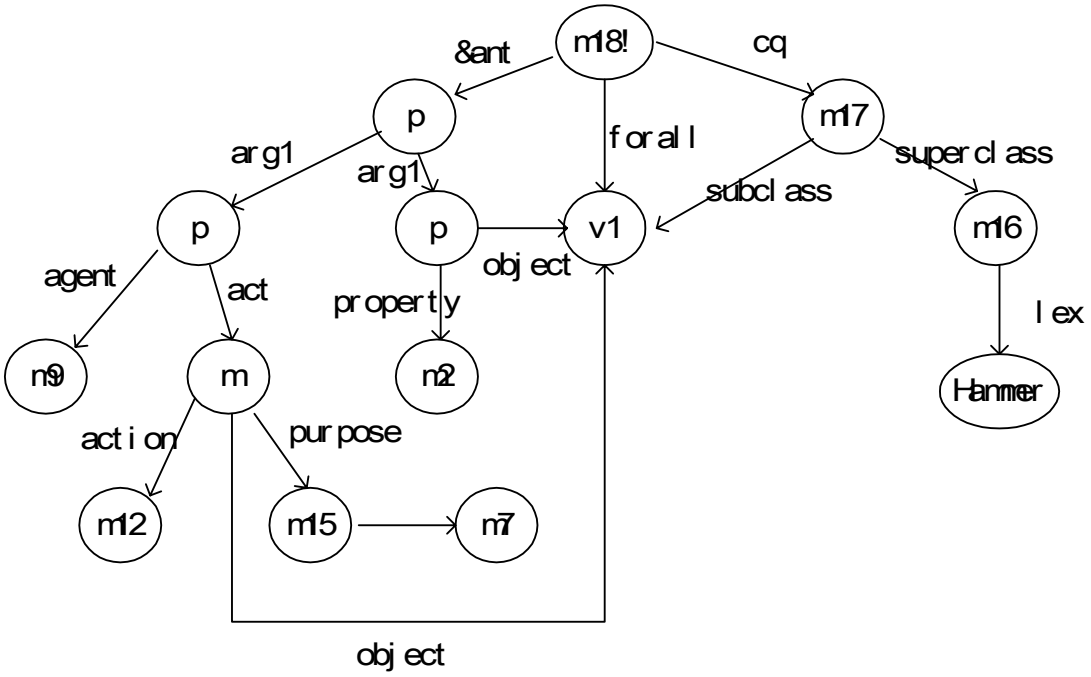

Figure 1 Background knowledge representation.

Most of our case frames are from [3]. However, it is necessary to define new case frames. Here, we defined a new case frame: agent/act// action//object//purpose to represent the background knowledge.

Syntax: (build agent i act (build action j object k purpose p))
Semantics: [[m]] is the proposition that agent [[i]] performs an action [[j]] on object [[k]] for purpose [[p]].

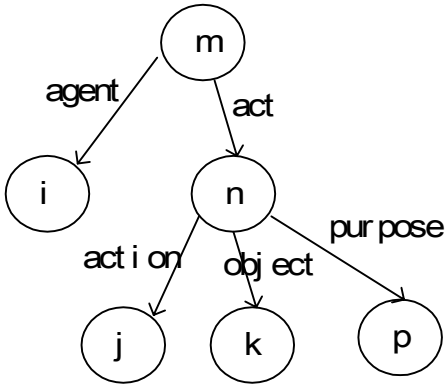

Figure 2 New defined case frame: agent/act//action//object//purpose

2. Object “Maul” has property “Heavy”.
(assert object (build lex "Maul") property (build lex "Heavy"))

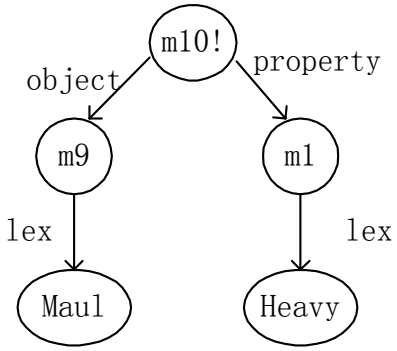

Figure 3 Object “Maul” has property “Heavy”

3. Person uses “Maul” for “Pounding” in the fence posts.

   1) "In" the "Fence_posts".

   (define preposition position)

   (assert preposition (build lex "In") position (build lex "Fence_posts"))

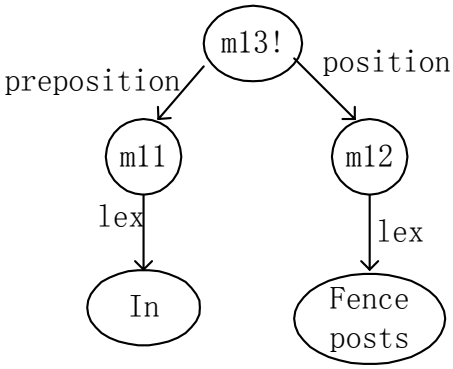

Figure 4 “In” the “Fence Posts”

Here, we defined case frame: preposition/position.

Syntax: (build preposition i position j)

Semantics: [[m]] is the proposition that [[i]] (such as ‘at’, ‘in’, etc) the position of [[j]].

2) "Person" is member of "Human"

(assert member (build lex "Person") class (build lex "Human"))

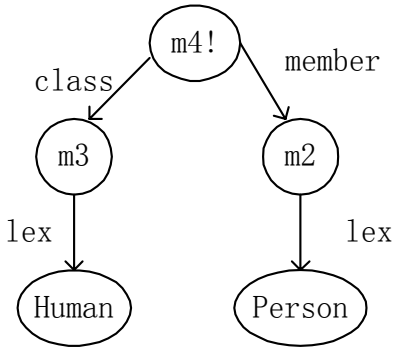

Figure 5 "Person" is member of "Human"

3) Person use “Maul” for purpose “Pounding” in the fence posts
(define location)
(assert agent (build lex "Person") act (build action (build lex "Use")
object (build lex "Maul")
purpose (build lex "Pounding") location (build preposition (build lex "In")
position (build lex "Fence_posts"))))

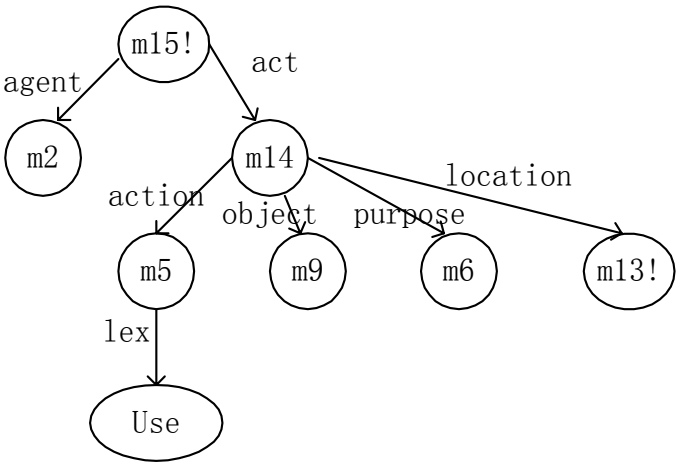

Figure 6 Person use “Maul” for purpose “Pounding” in the fence posts

Here, we add “location” to the first new case frame, making it as agent/act//action//object//purpose//location.

Syntax: (build agent i act (build action j object k purpose p in location q))
Semantics: [[m]] is the proposition that agent [[i]] performs an action [[j]] on object [[k]] for purpose [[p]] in location [[q]].

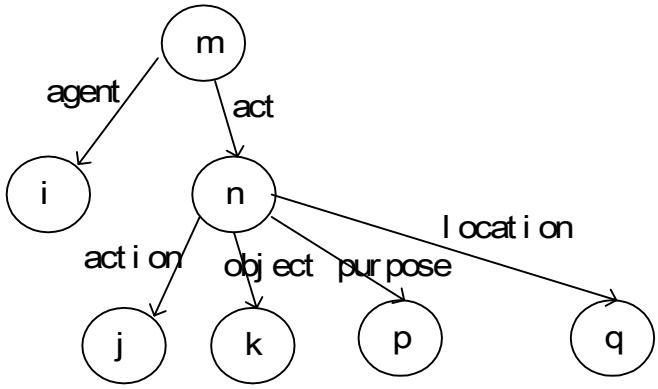

Figure 7 Case frame: agent/act//action//object//purpose//location

4. Maul “Lay” “Unused” at his(Person’s) side

(define direction status possesor)

(assert agent (build lex "Maul") act (build action (build lex "Lay")

status (build lex "Unused") location (build preposition (build lex "At")

position

(build possesor (build lex "Person") direction (build lex "Side")))))

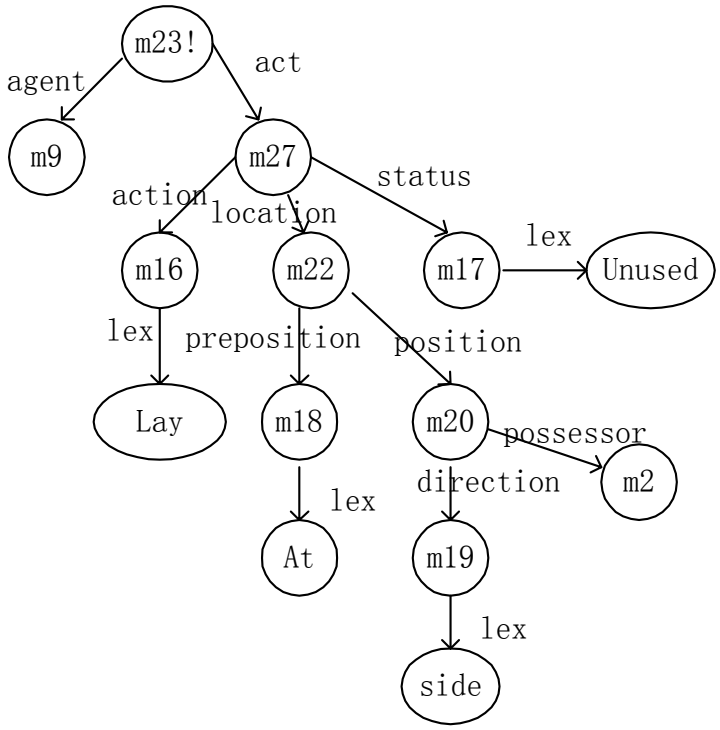

Figure 8 Maul “Lay” “Unused” at his(Person’s) side

Here, we define two new case frames. The first is: agent/act//action//location//status with definition as follows:

Syntax: (build agent i act (build action j location q status s))

Semantics: [[m]] is the proposition that agent [[i]] performs an action [[j]] on in location [[q]] with status [[s]].

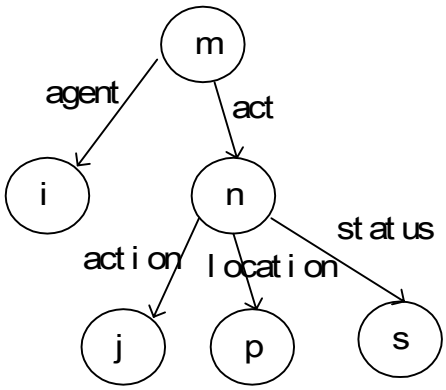

Figure 9 Case frame: agent/act//action//location//status

The second case frame we defined is possessor/direction with definition as follows:

Syntax: (build possessor i direction j)
Semantics: [[m]] is the proposition that i's direction j (such as "his side", "top of desk", etc).

Thus, through above representations with the help of new defined case frames, the meaning of "Maul" can be deduced automatically.

4. Conclusion and future work

Our experience on representing the passage and sentences told us that different people have different representations using SNePS. To represent a single sentence, users may try several ways, including defining different case frames.

In the immediate future, it is interesting to investigate this problem: if we try different representations on the passage, will the deduced meaning of the unknown word be the same?

In the long-term future, it is necessary to explore this problem: if we are provided with different passages which contains the same unknown word, how to update the deduced meaning of the word? We may find our current deduced meaning is incorrect or incomplete. Also, whether it is possible to make machine automatically represent the given passage and find out the meaning of work automatically. Of course, this may be the most challenging problem in CVA project as well as the final goal of CVA.

References

[1] http://www.cse.buffalo.edu/~rapaport/CVA/cvapassages.html

[2] Rapaport, William J., & Ehrlich, Karen (2000), "A Computational Theory of Vocabulary Acquisition", in Lucja M. Iwanska & Stuart C. Shapiro (eds.), Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language (Menlo Park, CA/Cambridge, MA: AAAI Press/MIT Press): 347-375.

[3]Rapaport, William J., & Kibby, Michael W. (2002), "ROLE: Contextual Vocabulary Acquisition: From Algorithm to Curriculum".

[4] Scott Napieralski's Dictionary of CVA SNePS Case Frames
http://www.cse.buffalo.edu/~stn2/cva/case-frames/

Demo

==========

```
; FILENAME:     MAUL.demo
; DATE:        11-15-2003
; PROGRAMMER:    Hansheng Lei

;
; To use this file: run SNePS; at the SNePS prompt (*), type:
;
;      (demo "MAUL.demo" :av)
;
;
============================================================


^(load "/projects/stn2/CVA/defun_noun.cl")

;set trace level
^(setTraceLevel 4)

; Reset the network
(resetnet t)


^(in-package snip)


^(defun broadcast-one-report (rep)
     (let (anysent)
        (do.chset (ch *OUTGOING-CHANNELS* anysent)
                (when (isopen.ch ch)
            (setq anysent (or (try-to-send-report rep ch) anysent)))))
            nil)



^(in-package sneps)

(intext "/projects/stn2/CVA/demos/rels")

; load all pre-defined path definitions:
(intext "/projects/rapaport/CVA/mkb3.CVA/paths/paths")


;;Backgroud knowledge:if person use X for purpose "pounding" and X is heavy, then
x is a kind of hammar
```

```
(define object property agent act action purpose member class lex)
(assert forall $p
            &ant ( (build object *p property (build lex "Heavy"))
                        (build member (build lex "Person") class (build lex "Human"))
                        (build agent (build lex "Person") act (build action
                    (build lex "Use") object *p purpose
            (build lex "Pounding")))
              )
            cq (build subclass *p superclass (build lex "Hammer"))
)

;Represent    maul is heavy
(assert object (build lex "Maul") property (build lex "Heavy"))

;;Represent 'Person uses Maul for pounding"
;1. "in" the "fence posts"
(define preposition position)
(assert preposition (build lex "In") position (build lex "Fence_posts"))



;2. "person" is member of "human"
(assert member (build lex "Person") class (build lex "Human"))

;3. person use maul for pupose pounding in the fence posts
(define location)
(assert agent (build lex "Person") act (build action (build lex "Use")
object (build lex "Maul")
purpose (build lex "Pounding") location (build    preposition (build lex
"In") position (build lex "Fence_posts")) ))

;; Maul lay unused at his side

(define direction status possesor)

(assert agent (build lex "Maul") act (build action (build lex "Lay")
status (build lex "Unused") location (build preposition (build lex "At") position
(build possesor (build lex "Person") direction (build lex "Side")))))

;Deduce: Is Maul a kind of Hammer?
(deduce subclass (build lex "Maul") superclass (build lex "Hammer"))

; Ask Cassie what "Maul" means:
; ========================
```

^(defineNoun "Maul")