Intelligent Systems: Reasoning and Recognition

James L. Crowley

MoSIG M1 — Winter Semester 2017/2018
Exercise 8 — 5 April 2018

The goal of this exercise is to write CLIPS rules that will make it possible to recognize categories of text in CLIPS with a ratio of histograms. These rules compute an occurrence matrix, which is the first step in Latent Semantic Analysis.

An N-Gram is a sequence of N symbols. N-grams of words are a common feature used to classify text. In this exercise we will use clips rules to count the frequency of occurrence of word 2-Grams (word-pairs) in different categories of text. Example categories can include email, SMS, technical writing, publicity, spam, etc.

You can assume that text is provided as a fact in working memory of the form:

```
      (Paragraph class  w1 w2 … wN)
```

where the `<wn>` are the individual words of the paragraph.

Your system should use the following templates for Word-Pair and Category.

```
(deftemplate WordPair ; structure for ccounting Word Pairs
      (slot CATEGORY (type SYMBOL))
      (slot WORD1 (type SYMBOL))
      (slot WORD2 (type SYMBOL))
      (slot COUNT (type INTEGER))  ; Number of instances of word pair
)

(deftemplate Category   ; A category of text (e.g. legal, spam, etc)
      (slot NAME) (type SYMBOL)) ; Name for Category of Text
      (slot M (type INTEGER)) ; Size of Training Set for Category
```

a) Write a rule named MakeWordPair to generate a fact of type WordPair for each new word pair in a paragraph of a sample of a category. Be sure to include the category when you create each Word-Pair. Be sure not to create more than one fact for each word pair. Test this rule with the following paragraph and show the facts list.

```
(assert (Paragraph Test This is a test.  This is another test.))
```

b) Write a rule named CountWordPairs that updates the count for each word-pair in the paragraph. Make sure that this rule also updates the size of the training set, M, for the category. Test the rule and show the trace.

c) Assume that you have a large corpus (training set) of text that includes several paragraphs of several categories. Write a rule or set of rules to determine the sum of the counts for each word-pair for all categories. Store the result as facts of Word-Pair with a category name of "All". Name your rule(s) CountPairsForAllCategories.

d) Write a rule to print the most frequent word-pair in each category, along with its count. If several word-pairs have the same most frequent count, then print them all. The printed message should say:

"The most frequent word pair in category <C> is <W1> <W2> with count <N>"

where <C>, <W1>, <W2>, <N> represent the category, words and count. Test the rule with the following.

(assert (Paragraph Test2 This is also a test.))

```
;;;;
;;;;  Here are some rules to open an close text files.
;;;;
;;;;  Rule to open a file of text


(defrule init
  (initial-fact)
=>
  (printout t "Name of file to read? ")
  (bind  ?filename (read))
  (printout t "Catagory of text? ")
  (bind ?category (read))
  (bind ?flag (open ?filename data "r"))
  (printout t "(file " ?category ?flag ")" crlf)
  (assert (file ?category ?flag))
)

;;; If file does not exist

(defrule no-file
   ?f <- (file ?c FALSE)
=>
   (retract ?f)
   (printout t "File not found" crlf)
)

;;; Read a paragraph of text ;;

(defrule ReadLineOfText
   ?f<-(file ?class TRUE)
    (not (line ?class EOF))
=>
    (bind ?line (readline data))
    (printout t ?line crlf)
    (assert (line ?class ?line))
    (retract ?f)
    (assert (file ?class TRUE))
)

(defrule eof
   (declare (salience 10))
   ?f <- (file ?class TRUE)
   ?eof <- (line ?class EOF)
=>
   (retract ?f ?eof)
   (close data)
)

(defrule ConverLineToParagraph
    ?l <- (line ?class ?line)
=>
   (assert (Paragraph ?class (explode$ ?line)))
   (retract ?l)
)
```