

Intelligent Systems: Reasoning and Recognition

James L. Crowley

Ensimag 2
Exercise 9

Winter Semester 2018/2019
5 April 2019

The goal of this exercise is to write CLIPS rules that will make it possible to recognition categories of text in CLIPS. We will use these rules in a later example to classify texts.

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 (2-Grams of words)
  (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. scientific, legal, spam, etc)
  (slot NAME (type SYMBOL)) ; Name for Category of Text
  (slot M (type INTEGER)) ; Size of Training Set for Category
)
```

a) Create a rule named MakeCategory that has a slot for category name and M, the number of words of that category, with a default of 0.

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.

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.

c) Write a rule named MostFrequentWordPair to print the category and words of the most frequent word-pair for all categories, 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 is <W1> <W2> in category <C> with count <N>"

where <C>, <W1>, <W2>, <N> represent the category, words and count.

d) write a rule name GetProbe with salience -10 that asks the user for a word pair, and creates a word pair of category "Probe". Write a second rule named FindMostLikelyCategory that determines the most probable category for this word pair of type Probe.

```
;;; And 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)
)
```