**Pseudo-code for the 20 features implemented:**

**int f1():**

counter = 1

for token in sentence:

if token == “it”:

return counter

else:

counter = counter + 1

return -1 /// in case no “it” is found in sentence.

**Note**: token is a unit that composes a sentence, e.g., a word or a punctuation

**int f2():**

counter = 0

for token in sentence:

counter = counter + 1

return counter

**int f3():**

counter = 0

for token in sentence:

if token is punctuation:

counter = counter + 1

return counter

**Note**: f1,f2,f3 can be calculated together within 1 loop thus it is also reasonable to implement them in one function. (Pseudo Code only provides the logics)

**int f4():**

counter = 0

for token in sentence:

if (token’s index < index of “it”) and token is noun:

counter = counter + 1

return counter

**int f5():**

counter = 0

for token in sentence:

if (token’s index > index of “it”) and token is noun:

counter = counter + 1

return counter

**Note**: ① index of “it” is the result of f1, we aim to compute the features sequentially in this case. ② f4, f5 can be calculated together within 1 loop

**boolean f6():**

if index\_of\_it == sentence.size():

return false

else if the token next to the “it” has tag that is “IN”:

return true

else:

return false

**Note**: “IN” is the Pos tag for prepositions and subordinating conjunctions in Natural Language Processing

**String[] f7():**

String[] result = new String[] // Make an empty array

for four tokens preceding the “it” and four tokens succeeding the “it”:

if such token exists:

add that token into result sequentially

else: // Case that such token doesn’t exist

add ABS into result sequentially

**Note**: we can check whether tokens preceding “it” exists by looking if its index is greater than or equal to 0; and we can check whether tokens succeeding “it” exists by using “try and catch” in Java