The pseudo code in (PF)Specification-stage

The pseudo code of feature Sim(Q,KBQ)

Input: Query and Question
Output: the result of Sim(Q,KBQ)

```
    //Extract the semantic block of Query to form list1, extract the semantic block of
    Question to form list2.
1.  list1 = semantic(Query),list2 = semantic(Question)
    //Iterate through list1 to get each semantic block item item1 of the Query
2.  for i, item1 in enumerate(list1) do
    //Iterate through list2 to get each semantic block item item2 of the Question
3.      for i, item2 in enumerate(list2) do
    //Use the Synonyms toolkit to calculate the similarity between item1 and item2
semantic blocks to get the score, seg=False means no more splitting of semantic
blocks
4.          score = synonyms.compare(item1, item2, seg=False)
    //score_max is used to semantically preserve the maximum value of the similarity
score
5.          if score > score_max then
6.              score_max = score
7.          end if
    //To get the score_max of item1 on item2 and save it in the list
8.              list.append(score_max)
9.      endfor
10. endfor
11. Sim(Q,KBQ) = sum(list) / len(list)
12. return Sim(Q,KBQ)
```

The pseudo code of feature Sim(Q,KBA)

Input: Query and Answer
Output: the result of Sim(Q,KBA)

```
// Extract the semantic block of Query to form list1, extract the semantic block of Answer to form list2
1.    list1 = semantic(Query),list2 = semantic(Answer)
//Iterate through list1 to get each semantic block item item1 of the Query
2.    for i, item1 in enumerate(list1) do
//Iterate through list2 to get each semantic block item item2 of the Answer
3.        for i, item2 in enumerate(list2) do
//Use the Synonyms toolkit to calculate the similarity between item1 and item2 semantic blocks to get the score, seg=False means no more splitting of semantic blocks
4.            score = synonyms.compare(item1, item2, seg=False)
//score_max is used to semantically preserve the maximum value of the similarity score
5.            if score > score_max then
6.                score_max = score
7.            end if
//To get the score_max of item1 on item2 and save it in the list
8.                list.append(score_max)
9.        endfor
10.   endfor
11.   Sim(Q,KBA) = sum(list) / len(list)
12.   return Sim(Q,KBQ)
```

The pseudo code of feature Sim(QE,KBQE)

Input: Query and Question
Output: the result of Sim(QE,KBQE)

```
//Extract the entity of Query to form list1, extract the entity of Question to form list2
list1 = Entity(Query),list2 = Entity(Question)
//For each word in list1 and list2, look up the word in cilin_vocab and save the words not included in cilin_vocab to vocab_no_words
for word in list1+list2 do
    if word not in cilin_vocab then
        vocab_no_words.append(word)
    endif
endfor
//Iterate through list1 to get each entity item item1 of the Query
for i, item1 in enumerate(list1) do
//Iterate through list2 to get each entity item item2 of the Question
    for i, item2 in enumerate(list2) do
//Determine if item1 and item2 exist in vocab_no_words
        inter = (set([item1, item2]).intersection(set(vocab_no_words)))
        if inter then
//If it exists,use the Synonyms toolkit to calculate the similarity between item1 and item2 semantic blocks to get the score, seg=False means no more splitting of semantic blocks
            score = synonyms.compare(item1, item2, seg=False)
        else
//If not, calculate the similarity score of item1 and item2 using the HIT synonym word forest method
            score = ci_lin.sim2018(item1, item2)
        endif
        if score > score_max then
//score_max is used to store the maximum value of entity similarity.
            score_max = score
        endif
//To get the score_max of item1 on item2 and save it in the list
        list.append(score_max)
    endfor
endfor
Sim(QE,KBQE) = sum(list)/len(list)
return Sim(QE,KBQE)
```

The pseudo code of feature Sim(QR,KBQR)

Input: Query and Question
Output: the result of Sim(QR,KBQR)

```
  //Extract the Relation of Query to form list1, extract the Relation of Question to form list2
1.    list1 = Relation(Query),list2 = Relation(Question)
  //Iterate through list1 to get each Relation item item1 of the Query
2.    for i, item1 in enumerate(list1) do
  //Iterate through list2 to get each Relation item item2 of the Question
3.        for i, item2 in enumerate(list2) do
  //use the Synonyms toolkit to calculate the similarity between item1 and item2 semantic blocks to get the score, seg=False means no more splitting of semantic blocks
4.            score = synonyms.compare(item1, item2, seg=False)
5.            if score > score_max then
  //score_max is used to store the maximum value of entity similarity.
6.                score_max = score
7.            endif
8.        endfor
9.    endfor
10.   Sim(QR,KBQR) = score_max
11.   return Sim(QR,KBQR)
```