```
Input: All comments under one req C
Output: Comments contain questions Cq

Main Procedure:
 foreach comments c∈C:
      if(∃ sentence s in c contains “?” && s not start with "{quote}" &&  s not start with "bq.")
           Cq=Cq+{c}
     c'=removeURL&Code(c)
     c'=removeQuote(c')
     cmarked=StanfordNLPGrammarParser(c)
     if(∃ sentence s in cmarked  && (scontains “SBARQ” || s contains “SQ”) && s not start with
"{quote}" &&  s not start with "bq.")
           Cq=Cq+{c}
 Cq'=coreferenceQ(C)
 Cq=Cq+Cq'
  return Cq

removeURL&Code(c)
   if(c contains URL u)
        c=c.replace(u,"")
  if(c contains XML code x)
        c=c.replace(x,"")
  if(sub-string csin c && (cs start and end with “{code}” || cs start and end with “{noformat}” )
        c=c.replace(cs,"")
  if(paragraph p in c && (p start with “+” || p start with “-”)
        c=c.replace(p,"")
  if(term t in c && (t end with any file extensions)
        c=c.replace(t,"")
  return c

removeQuote(c)
   if(c contains bq.)
        remove the paragraph after bq.
  if(c contains {quote} pairs)
        remove text between {quote} pairs

coreferenceQ(C)[Note 3]
   Cmarked=StanfordNLPCoreferenceParser(C)
   foreach c∈C
      if(c∈Cmarked)
          foreach sentence s in c
              if(s contains co-reference)
                 if(∃ coreference coin s & !isPerson(co,c))
                    Cq'=Cq'+{c}
   return Cq'


isPerson(co, c)[Note 4]
   if(co start with “[~” and end with “])
       return True
```

```
persons = StarnfordNLPNamedEntityRecognisionParser(c)
if(∃ person p in persons& p==co)
    return True
if(co is a person pronoun)
    return True
```

StanfordNLPGrammarParser: returns labeled sentences (Part-of-speech)
StanfordNLPCoreferenceParser: return labeled sentences (Co-reference)
StarnfordNLPNamedEntityRecognisionParser: return labeled sentences (entities)