NATURAL LANGUAGE PROCESSING

REPORT FILE

AKASH KUMAR GAUTAM(2015011)

Text Processing

The given text in the document corpus has been converted to string data type in python and processed appropriately .

Assumptions

1.The text which is taken as input in any of the parts gives output which is case insensitive.In Fact all the processing of any word is not depended on any case whatsoever .This has been done to make sure that maximum outputs are produced based on a input value.

2. Given sentence in the corpus can end with .(full stop),!(exclamation mark) or ?(question mark).

3.All the words which contain any punctuation between them such as b.p.m. or tic-tac are considered to be a single word only .

4.The input word in any of the parts of questions may contain special characters like . , ”, ! , ? which is handled explicitly .

5.Whenever words are being searched only boundary occurrences \b are being checked .so let’s say of we are searching for foo and corpus contains hello:foo:bar then it will not be the part of output set.

6.All the processing of text is case insensitive except for finding the number of sentences in text where case sensitive text similar to document corpus is being used .

7.Numerical occurrences are considered to be distinct words only eg 160 or 175 will be considered a word only .

8.A single character is not considered a word ,at least two words are required to make it a word.

Approach/Implementation:

1.Number of words on text file:-Since a word is considered to be a sequence of characters which may contain .,- in between them so regular word expressions are used which are then combined with [-|.] which is optional .

Regex expression produced:**r"\w+[-|.|-|']?\w+"**

2.Number of paragraphs in text file:A paragraph will always be followed by \n so we need to find occurences of two \n’s together .

Regex expression produced:**r”\n\n”**

3.Total number of sentences:A logical sentence will always start from a capital letter and move on till a full stop is encountered or a sentence stopping punctuation.Now in between the capital letter and the full stop there should not be even a single period(.) .Before the start of capital letter it is necessary to ensure that previous char to capital letter is a sentence stopping punctuation i.e full stop,exclamation mark,question is checked before that .

Regex expression produced:**r"[\n\.!?]?[A-Z][^.]{6,}[?\.!] ?"**

4.Number of sentences beginning with a word:Check for closing of prior sentence with a sentence finishing punctuation and match the word .If the word is the starting of string then it is handled explicitly .

Regex expression produced:**r"[.|?|!|\n] ?"+re.escape(str1)+"\\b"+"|\G"+str1+"\\b**"

5.Number of sentences ending with a word:The punctuation succeeding the word should be a sentence stopping punctuation.

Regex expression produced:**r" ?"+re.escape(str2)+"\\b"+" ?[.|?|!]"**

6.Frequency of a word:Positive look behind assertion is used to find the boundary occurences of a word .

Regex expression produced:**r"(?<!\\S)"+re.escape(str3)+"(?!\\S)"**