Perform tokenization, stemming and lemmatization, stopwords and punctuation removal on the given text -

"A major drawback of statistical methods is that they require elaborate feature engineering. Since the early 2010s,[16] the field has thus largely abandoned statistical methods and shifted to neural networks for machine learning. Popular techniques include the use of word embeddings to capture semantic properties of words, and an increase in end-to-end learning of a higher-level task (e.g., question answering) instead of relying on a pipeline of separate intermediate tasks (e.g., part-of-speech tagging and dependency parsing). In some areas, this shift has entailed substantial changes in how NLP systems are designed, such that deep neural network-based approaches may be viewed as a new paradigm distinct from statistical natural language processing. For instance, the term neural machine translation (NMT) emphasizes the fact that deep learning-based approaches to machine translation directly learn sequence-to-sequence transformations, obviating the need for intermediate steps such as word alignment and language modeling that was used in statistical machine translation (SMT)."

```
#  Tokenization
doc="""A major drawback of statistical methods is that they require elaborate feature engi
Since the early 2010s,[16] the field has thus largely abandoned statistical methods and sh
Popular techniques include the use of word embeddings to capture semantic properties of wo
(e.g., question answering) instead of relying on a pipeline of separate intermediate tasks
 In some areas, this shift has entailed substantial changes in how NLP systems are designe
 such that deep neural network-based approaches may be viewed as a new paradigm distinct f
 For instance, the term neural machine translation (NMT) emphasizes the fact that deep lea
  machine translation directly learn sequence-to-sequence transformations, obviating the n
  and language modeling that was used in statistical machine translation (SMT)"""
```

```
doc.split()
```

```
      'neural',
      'network-based',
      'approaches',
      'may',
      'be',
      'viewed',
      'as',
      'a',
      'new',
      'paradigm',
      'distinct',
      'from',
      'statistical',
      'natural',
      'language',
      'processing.',
      'For',
      'instance,',
      'the',
      'term',
      'neural',
      'machine',
      'translation'
```

```
 'translation',
 '(NMT)',
 'emphasizes',
 'the',
 'fact',
 'that',
 'deep',
 'learning-based',
 'approaches',
 'to',
 'machine',
 'translation',
 'directly',
 'learn',
 'sequence-to-sequence',
 'transformations,',
 'obviating',
 'the',
 'need',
 'for',
 'intermediate',
 'steps',
 'such',
 'as',
 'word',
 'alignment',
 'and',
 'language',
 'modeling',
 'that',
 'was',
 'used',
 'in',
 'statistical',
 'machine',
 'translation',
 '(SMT)']
```

```
import nltk

nltk.download('punkt') #for word tokenization
nltk.download('stopwords') #for removing or getting list of stopwords
nltk.download('wordnet') #for lemmatization
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
from nltk.tokenize import word_tokenize


# nltk.download('punkt')
tokens = word_tokenize(doc)
```

```
tokens
```

```
'e.g.',
',',
'part-of-speech',
'tagging',
'and',
'dependency',
'parsing',
')',
'.',
'In',
'some',
'areas',
',',
'this',
'shift',
'has',
'entailed',
'substantial',
'changes',
'in',
'how',
'NLP',
'systems',
'are',
'designed',
',',
'such',
'that',
'deep',
'neural',
'network-based',
'approaches',
'may',
'be',
'viewed',
'as',
'a',
'new',
'paradigm',
'distinct',
'from',
'statistical',
'natural',
'language',
'processing',
'.',
'For',
'instance',
',',
'the',
'term',
'neural',
'machine',
'translation',
'(',
'NMT',
')',
'emphasizes',
'the',
```

```
from nltk.corpus import stopwords

from string import punctuation

stop = stopwords.words('english')

punc = list(punctuation)

stop
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
```

```
        'are',
        'was',
        'were',
        'be',
        'been',
        'being',
        'have',
        'has',
        'had',
        'having',
        'do',
        'does',
        'did',
```

```
punc
```

```
     ['!',
      '"',
      '#',
      '$',
      '%',
      '&',
      "'",
      '(',
      ')',
      '*',
      '+',
      ',',
      '-',
      '.',
      '/',
      ':',
      ';',
      '<',
      '=',
      '>',
      '?',
      '@',
      '[',
      '\\',
      ']',
      '^',
      '_',
      '`',
      '{',
      '|',
      '}',
      '~']
```

```
bad_tokens = stop + punc
```

```
clean_tokens = []
for t in tokens:
  if t not in bad_tokens:
    clean_tokens.append(t)
```

```
clean_tokens
```

```
['A',
 'major',
 'drawback',
 'statistical',
 'methods',
 'require',
 'elaborate',
 'feature',
 'engineering',
 'Since',
 'early',
 '2010s',
 '16',
 'field',
 'thus',
 'largely',
 'abandoned',
 'statistical',
 'methods',
 'shifted',
 'neural',
 'networks',
 'machine',
 'learning',
 'Popular',
 'techniques',
 'include',
 'use',
 'word',
 'embeddings',
 'capture',
 'semantic',
 'properties',
 'words',
 'increase',
 'end-to-end',
 'learning',
 'higher-level',
 'task',
 'e.g.',
 'question',
 'answering',
 'instead',
 'relying',
 'pipeline',
 'separate',
 'intermediate',
 'tasks',
 'e.g.',
 'part-of-speech',
 'tagging',
 'dependency',
 'parsing',
 'In',
 'areas',
 'shift',
 'entailed',
 'substantial',
 'changes',
 'NLP',
```

```
clean_tokens = [t for t in tokens if t not in bad_tokens]
```

```
len(tokens)
```

```
176
```

```
len(clean_tokens)
```

```
106
```

```
from nltk.stem import PorterStemmer
from nltk.stem import LancasterStemmer
```

```
porter = PorterStemmer()
for c in clean_tokens:
  print(porter.stem(c))
```

```
A
major
drawback
statist
method
requir
elabor
featur
engin
sinc
earli
2010
16
field
thu
larg
abandon
statist
method
shift
neural
network
machin
learn
popular
techniqu
includ
use
word
embed
captur
semant
properti
word
increas
end-to-end
learn
higher-level
task
e.g.
```

```
question
answer
instead
reli
pipelin
separ
intermedi
task
e.g.
part-of-speech
tag
depend
pars
In
area
shift
entail
substanti
chang
```

```
lancaster = LancasterStemmer()
[lancaster.stem(c) for c in clean_tokens]
```

```
 'task',
 'e.g.',
 'part-of-speech',
 'tag',
 'depend',
 'pars',
 'in',
 'area',
 'shift',
 'entail',
 'subst',
 'chang',
 'nlp',
 'system',
 'design',
 'deep',
 'neur',
 'network-based',
 'approach',
 'may',
 'view',
 'new',
 'paradigm',
 'distinct',
 'stat',
 'nat',
 'langu',
 'process',
 'for',
 'inst',
 'term',
 'neur',
 'machin',
 'transl',
 'nmt',
 'emphas',
 'fact',
 'deep',
```

```
 'learning-based',
 'approach',
 'machin',
 'transl',
 'direct',
 'learn',
 'sequence-to-sequenc',
 'transform',
 'obvy',
 'nee',
 'intermedy',
 'step',
 'word',
 'align',
 'langu',
 'model',
 'us',
 'stat',
 'machin',
 'transl',
 'smt']
```

```
from nltk.stem import WordNetLemmatizer


lemma = WordNetLemmatizer()
[lemma.lemmatize(c) for c in clean_tokens]
```

```
['A',
 'major',
 'drawback',
 'statistical',
 'method',
 'require',
 'elaborate',
 'feature',
 'engineering',
 'Since',
 'early',
 '2010s',
 '16',
 'field',
 'thus',
 'largely',
 'abandoned',
 'statistical',
 'method',
 'shifted',
 'neural',
 'network',
 'machine',
 'learning',
 'Popular',
 'technique',
 'include',
 'use',
 'word',
 'embeddings',
 'capture',
 'semantic',
```

```
 'property',
 'word',
 'increase',
 'end-to-end',
 'learning',
 'higher-level',
 'task',
 'e.g.',
 'question',
 'answering',
 'instead',
 'relying',
 'pipeline',
 'separate',
 'intermediate',
 'task',
 'e.g.',
 'part-of-speech',
 'tagging',
 'dependency',
 'parsing',
 'In',
 'area',
 'shift',
 'entailed',
 'substantial',
 'change',
```

✓ 1s completed at 9:16 AM