```
hospital="""Medical text corpus datasets available on Kaggle are useful for NLP and healthcare research. The Medical Text Datas
```

```
hospital="""Medical text corpus datasets available on Kaggle are useful for NLP and healthcare research. The Medical Text Datas
print(hospital)
```

```
Medical text corpus datasets available on Kaggle are useful for NLP and healthcare research. The Medical Text Dataset contains l
```

```
import nltk
nltk.download('punkt')
nltk.download('punkt_tab') # Added to download the missing resource
from nltk.tokenize import word_tokenize
word_tokenize(hospital)
```

```
 'labeled',
 'abstracts',
 'suitable',
 'for',
 'beginners',
 '.',
 'For',
 'multilingual',
 'NLP',
 ',',
 'the',
 'Multilingual',
 'Healthcare',
 'Text',
 'Dataset',
 'offers',
 'healthcare',
 'text',
 'in',
 'Hindi',
 ',',
 'English',
 ',',
 'and',
 'Punjabi',
 '.',
 'The',
 'NBME',
 'Clinical',
 'Patient',
 'Notes',
 'Dataset',
 'contains',
 'annotated',
 'clinical',
 'notes',
 'for',
 'advanced',
 'medical',
 'text',
 'understanding',
 '.',
 'These',
 'datasets',
 'can',
 'be',
 'downloaded',
 'using',
 'the',
 'Kaggle',
 'API',
 'and',
 'used',
 'for',
 'medical',
 'NLP',
 'projects',
 '.']
```

```
from nltk.tokenize import sent_tokenize
sent_tokenize(hospital)
```

```
['Medical text corpus datasets available on Kaggle are useful for NLP and healthcare research.',
 'The Medical Text Dataset contains labeled medical transcriptions for text classification, while the PubMed 200k RCT Dataset
includes over 200,000 PubMed abstracts with sentence-level labels for scientific text analysis.',
 'The Medical Abstract Classification Dataset provides disease-based labeled abstracts suitable for beginners.',
 'For multilingual NLP, the Multilingual Healthcare Text Dataset offers healthcare text in Hindi, English, and Punjabi.',
 'The NBME Clinical Patient Notes Dataset contains annotated clinical notes for advanced medical text understanding.',
 'These datasets can be downloaded using the Kaggle API and used for medical NLP projects.']
```

```
nltk.download("stopwords")
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
words_in_quote = word_tokenize(hospital)
words_in_quote
```

```
 'labeled',
 'abstracts',
 'suitable',
 'for',
 'beginners',
 '.',
 'For',
 'multilingual',
 'NLP',
 ',',
 'the',
 'Multilingual',
 'Healthcare',
 'Text',
 'Dataset',
 'offers',
 'healthcare',
 'text',
 'in',
 'Hindi',
 ',',
 'English',
 ',',
 'and',
 'Punjabi',
 '.',
 'The',
 'NBME',
 'Clinical',
 'Patient',
 'Notes',
 'Dataset',
 'contains',
 'annotated',
 'clinical',
 'notes',
 'for',
 'advanced',
 'medical',
 'text',
 'understanding',
 '.',
 'These',
 'datasets',
 'can',
 'be',
 'downloaded',
 'using',
 'the',
 'Kaggle',
 'API',
 'and',
 'used',
 'for',
 'medical',
 'NLP',
 'projects',
 '.']
```

```
stop_words = set(stopwords.words("english"))
filtered_list = []
for word in words_in_quote:
  if word.casefold() not in stop_words:
    filtered_list.append(word)
filtered_list
```

```
['Medical',
 'text',
 'corpus',
 'datasets',
 'available',
 'Kaggle',
 'useful',
 'NLP',
 'healthcare',
 'research',
 '.',
 'Medical',
 'Text',
 'Dataset',
 'contains',
 'labeled',
 'medical',
 'transcriptions',
 'text',
 'classification',
 ',',
 'PubMed',
 '200k',
 'RCT',
 'Dataset',
 'includes',
 '200,000',
 'PubMed',
 'abstracts',
 'sentence-level',
 'labels',
 'scientific',
 'text',
 'analysis',
 '.',
 'Medical',
 'Abstract',
 'Classification',
 'Dataset',
 'provides',
 'disease-based',
 'labeled',
 'abstracts',
 'suitable',
 'beginners',
 '.',
 'multilingual',
 'NLP',
 ',',
 'Multilingual',
 'Healthcare',
 'Text',
 'Dataset',
 'offers',
 'healthcare',
 'text',
 'Hindi',
```

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
stemmer = PorterStemmer()
words = word_tokenize(hospital)
stemmed_words = [stemmer.stem(word) for word in words]
stemmed_words
```

```
['medic',
 'text',
 'corpu',
 'dataset',
 'avail',
 'on',
 'kaggl',
 'are',
 'use',
 'for',
 'nlp',
 'and',
 'healthcar',
 'research',
 '.',
 'the',
 'medic',
 'text',
 'dataset',
 'contain',
 'label',
 'medic',
 'transcript',
 'for',
 'text',
 'classif',
 ',',
 'while',
 'the',
 'pubm',
 '200k',
 'rct',
 'dataset',
 'includ',
 'over',
 '200,000',
 'pubm',
 'abstract',
 'with',
 'sentence-level',
 'label',
 'for',
 'scientif',
 'text',
 'analysi',
 '.',
 'the',
 'medic',
 'abstract',
 'classif',
 'dataset',
 'provid',
 'disease-bas',
 'label',
 'abstract',
 'suitabl',
 'for',
```

```
from nltk.stem import SnowballStemmer
snowball = SnowballStemmer(language='english')
words = word_tokenize(hospital)
for word in words:
    print(word,"--->",snowball.stem(word))
```

```
advanced ---> advanc
advanced ---> advanc
medical ---> medic
text ---> text
understanding ---> understand
. ---> .
These ---> these
datasets ---> dataset
can ---> can
be ---> be
downloaded ---> download
using ---> use
the ---> the
Kaggle ---> kaggl
API ---> api
and ---> and
used ---> use
for ---> for
medical ---> medic
NLP ---> nlp
projects ---> project
. ---> .
```

```
from nltk import LancasterStemmer
Lanc = LancasterStemmer()
words = word_tokenize(hospital)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
Medical ---> med
text ---> text
corpus ---> corp
datasets ---> dataset
available ---> avail
on ---> on
Kaggle ---> kaggl
are ---> ar
useful ---> us
for ---> for
NLP ---> nlp
and ---> and
healthcare ---> healthc
research ---> research
. ---> .
The ---> the
Medical ---> med
Text ---> text
Dataset ---> dataset
contains ---> contain
labeled ---> label
medical ---> med
transcriptions ---> transcrib
for ---> for
text ---> text
classification ---> class
, ---> ,
while ---> whil
the ---> the
PubMed ---> pubm
200k ---> 200k
RCT ---> rct
Dataset ---> dataset
includes ---> includ
over ---> ov
200,000 ---> 200,000
PubMed ---> pubm
abstracts ---> abstract
with ---> with
sentence-level ---> sentence-level
labels ---> label
for ---> for
scientific ---> sci
text ---> text
analysis ---> analys
. ---> .
The ---> the
Medical ---> med
Abstract ---> abstract
Classification ---> class
Dataset ---> dataset
provides ---> provid
disease-based ---> disease-based
labeled ---> label
abstracts ---> abstract
suitable ---> suit
```

```
for ---> for
```

```
from nltk.stem import RegexpStemmer
regexp = RegexpStemmer('ing|e', min=4) # Corrected: Removed newlines from the string literal
words = word_tokenize(hospital)
for word in words:
    print(word,"--->",Lanc.stem(word))
```

```
labeled ---> label
abstracts ---> abstract
suitable ---> suit
for ---> for
beginners ---> begin
. ---> .
For ---> for
multilingual ---> multil
NLP ---> nlp
, ---> ,
the ---> the
Multilingual ---> multil
Healthcare ---> healthc
Text ---> text
Dataset ---> dataset
offers ---> off
healthcare ---> healthc
text ---> text
in ---> in
Hindi ---> hind
, ---> ,
English ---> engl
, ---> ,
and ---> and
Punjabi ---> punjab
. ---> .
The ---> the
NBME ---> nbme
Clinical ---> clin
Patient ---> paty
Notes ---> not
Dataset ---> dataset
contains ---> contain
annotated ---> annot
clinical ---> clin
notes ---> not
for ---> for
advanced ---> adv
medical ---> med
text ---> text
understanding ---> understand
. ---> .
These ---> thes
datasets ---> dataset
can ---> can
be ---> be
downloaded ---> download
using ---> us
the ---> the
Kaggle ---> kaggl
API ---> ap
and ---> and
used ---> us
for ---> for
medical ---> med
NLP ---> nlp
projects ---> project
. ---> .
```

```
nltk.download('omw-1.4')
nltk.download('wordnet') # Added to download the missing resource
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
words = word_tokenize(hospital)
for word in words:
    print(word,"--->",lemmatizer.lemmatize(word))
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
Medical ---> Medical
text ---> text
corpus ---> corpus
datasets ---> datasets
```

```
available ---> available
on ---> on
Kaggle ---> Kaggle
are ---> are
useful ---> useful
for ---> for
NLP ---> NLP
and ---> and
healthcare ---> healthcare
research ---> research
. ---> .
The ---> The
Medical ---> Medical
Text ---> Text
Dataset ---> Dataset
contains ---> contains
labeled ---> labeled
medical ---> medical
transcriptions ---> transcription
for ---> for
text ---> text
classification ---> classification
, ---> ,
while ---> while
the ---> the
PubMed ---> PubMed
200k ---> 200k
RCT ---> RCT
Dataset ---> Dataset
includes ---> includes
over ---> over
200,000 ---> 200,000
PubMed ---> PubMed
abstracts ---> abstract
with ---> with
sentence-level ---> sentence-level
labels ---> label
for ---> for
scientific ---> scientific
text ---> text
analysis ---> analysis
. ---> .
The ---> The
Medical ---> Medical
Abstract ---> Abstract
Classification ---> Classification
Dataset ---> Dataset
provides ---> provides
disease-based ---> disease-based
labeled ---> labeled
```

```
lemmatizer.lemmatize("worst")
```

```
'worst'
```

```
lemmatizer.lemmatize("worst", pos="a")
```

```
'bad'
```