```
# pip install -U spacy
# python -m spacy download en_core_web_lg
# python -m spacy download en_core_web_sm
# python -m spacy validate

import spacy


!python -m spacy download en_core_web_lg
```

```
Collecting en_core_web_lg==2.2.5
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_lg
     |████████████████████████████████| 827.9MB 1.1MB/s
Requirement already satisfied: spacy>=2.2.2 in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.6/dis
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.8" in /usr/
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-packages (from
Building wheels for collected packages: en-core-web-lg
  Building wheel for en-core-web-lg (setup.py) ... done
  Created wheel for en-core-web-lg: filename=en_core_web_lg-2.2.5-cp36-none-any.whl size
  Stored in directory: /tmp/pip-ephem-wheel-cache-23tfp562/wheels/2a/c1/a6/fc7a877b1efca
Successfully built en-core-web-lg
Installing collected packages: en-core-web-lg
Successfully installed en-core-web-lg-2.2.5
✓ Download and installation successful
You can now load the model via spacy.load('en_core_web_lg')
```

```
from collections import Counter
from string import punctuation


import en_core_web_lg
nlp = en_core_web_lg.load()


def get_topwords(text):
```

```
  # declare an empty list for results
  result = []

  # declare a list containing the POS (part of speech)
  # tags, thta we would like to extract
  # PROPN = Proper Noun
  # ADJ = Adjective
  # NOUN = Noun
  pos_tag = ['PROPN', 'ADJ', 'NOUN']

  doc = nlp(text.lower())
  # the above line converts the text to lowercase and tokenizes it
  # using the spacy model we loaded above

  for token in doc:
    # now we loop over(go through each token) and check if it is a stopword/
    # punctuation. If yes, just ignore it and move to the next token
    if (token.text in nlp.Defaults.stop_words or token.text in punctuation):
      continue

    # Now, store the result if POS tag of the tokenized text is matching
    # with any1 we specified in pos_tag
    if (token.pos_ in pos_tag):
      result.append(token.text)

  return result


output = get_topwords('''Natural language processing (NLP) is a subfield of
linguistics, computer science, information engineering, and artificial
intelligence concerned with the interactions between computers and human
(natural) languages, in particular how to program computers to process and
analyze large amounts of natural language data.Challenges in natural language
processing frequently involve speech recognition, natural language
understanding, and natural-language generation.The history of natural language
processing (NLP) generally started in the 1950s, although work can be found
from earlier periods. In 1950, Alan Turing published an article titled
"Computing Machinery and Intelligence" which proposed what is now called the
Turing test as a criterion of intelligence[clarification needed].The Georgetown
experiment in 1954 involved fully automatic translation of more than sixty
Russian sentences into English. The authors claimed that within three or five
years, machine translation would be a solved problem. However, real progress
was much slower, and after the ALPAC report in 1966, which found that
ten-year-long research had failed to fulfill the expectations, funding for
machine translation was dramatically reduced. Little further research in
machine translation was conducted until the late 1980s when the first
statistical machine translation systems were developed.Some notably successful
natural language processing systems developed in the 1960s were SHRDLU, a
natural language system working in restricted "blocks worlds" with restricted
vocabularies, and ELIZA, a simulation of a Rogerian psychotherapist, written by
Joseph Weizenbaum between 1964 and 1966. Using almost no information about
```

human thought or emotion, ELIZA sometimes provided a startlingly human-like interaction. When the "patient" exceeded the very small knowledge base, ELIZA might provide a generic response, for example, responding to "My head hurts" with "Why do you say your head hurts?".During the 1970s, many programmers began to write "conceptual ontologies", which structured real-world information into computer-understandable data. Examples are MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), QUALM (Lehnert, 1977), Politics (Carbonell, 1979), and Plot Units (Lehnert 1981). During this time, many chatterbots were written including PARRY, Racter, and Jabberwacky.Up to the 1980s, most natural language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language processing. This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language processing. Some of the earliest-used machine learning algorithms, such as decision trees, produced systems of hard if-then rules similar to existing hand-written rules. However, part-of-speech tagging introduced the use of hidden Markov models to natural language processing, and increasingly, research has focused on statistical models, which make soft, probabilistic decisions based on attaching real-valued weights to the features making up the input data. The cache language models upon which many speech recognition systems now rely are examples of such statistical models. Such models are generally more robust when given unfamiliar input, especially input that contains errors (as is very common for real-world data), and produce more reliable results when integrated into a larger system comprising multiple subtasks.Many of the notable early successes occurred in the field of machine translation, due especially to work at IBM Research, where successively more complicated statistical models were developed. These systems were able to take advantage of existing multilingual textual corpora that had been produced by the Parliament of Canada and the European Union as a result of laws calling for the translation of all governmental proceedings into all official languages of the corresponding systems of government. However, most other systems depended on corpora specifically developed for the tasks implemented by these systems, which was (and often continues to be) a major limitation in the success of these systems. As a result, a great deal of research has gone into methods of more effectively learning from limited amounts of data.Recent research has increasingly focused on unsupervised and semi-supervised learning algorithms. Such algorithms can learn from data that has not been hand-annotated with the desired answers or using a combination of annotated and non-annotated data. Generally, this task is much more difficult than supervised learning, and typically produces less accurate results for a given amount of input data. However, there is an enormous amount of non-annotated data available (including, among other things, the entire content of the World Wide Web), which can often make up for the inferior results if the algorithm used has a low enough time complexity to be practical.In the 2010s, representation learning and deep neural network-style machine learning methods became widespread in natural language processing, due in part to a flurry of results showing that such techniques can achieve state-of-the-art results in many natural language tasks, for example in language modeling, parsing, and many

```
natural language tasks, for example in language modeling, parsing, and many
others. 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).''')


# The result
print(output)
```

```
['natural', 'language', 'processing', 'nlp', 'subfield', 'linguistics', 'computer', 'sci
```

```
# ['natural', 'language', 'processing', 'nlp', 'subfield', 'linguistics',
# 'computer', 'science', 'information', 'engineering', 'artificial',
#  'intelligence', 'interactions', 'computers', 'human', 'natural', 'languages',
#  'particular', 'computers', 'large', 'amounts', 'natural', 'language',
#  'data.challenges', 'natural', 'language', 'processing', 'speech',
#  'recognition', 'natural', 'language', 'understanding', 'natural', 'language',
#  'history', 'natural', 'language', 'processing', 'nlp', '1950s', 'work',
#  'earlier', 'periods', 'alan', 'turing', 'article', 'machinery',
#  'intelligence', 'turing', 'test', 'criterion',
#  'intelligence[clarification', 'needed].the', 'georgetown', 'experiment',
#  'automatic', 'translation', 'russian', 'sentences', 'english', 'authors',
#  'years', 'machine', 'translation', 'problem', 'real', 'progress', 'slower',
#  'alpac', 'report', 'year', 'long', 'research', 'expectations', 'machine',
#  'translation', 'little', 'research', 'machine', 'translation', 'late',
#  '1980s', 'statistical', 'machine', 'translation', 'systems', 'developed.some',
#  'successful', 'natural', 'language', 'processing', 'systems', '1960s',
#  'shrdlu', 'natural', 'language', 'system', 'blocks', 'worlds', 'restricted',
#  'vocabularies', 'eliza', 'simulation', 'rogerian', 'psychotherapist',
#  'joseph', 'weizenbaum', 'information', 'human', 'thought', 'emotion',
#  'eliza', 'human', 'like', 'interaction', 'patient', 'small', 'knowledge',
#  'base', 'eliza', 'generic', 'response', 'example', 'head', 'head', '1970s',
#  'programmers', 'conceptual', 'ontologies', 'real', 'world', 'information',
#  'computer', 'understandable', 'data', 'examples', 'margie', 'schank', 'sam',
#  'cullingford', 'pam', 'wilensky', 'talespin', 'meehan', 'qualm', 'lehnert',
#  'politics', 'carbonell', 'plot', 'units', 'lehnert', 'time', 'chatterbots',
#  'parry', 'racter', 'jabberwacky.up', '1980s', 'natural', 'language',
#  'processing', 'systems', 'complex', 'sets', 'hand', 'rules', 'late', '1980s',
#  'revolution', 'natural', 'language', 'processing', 'introduction', 'machine',
#  'algorithms', 'language', 'processing', 'steady', 'increase', 'computational',
#  'power', 'moore', 'law', 'gradual', 'lessening', 'dominance', 'chomskyan',
#  'theories', 'linguistics', 'transformational', 'grammar', 'theoretical',
```

```
#  'underpinnings', 'sort', 'corpus', 'linguistics', 'machine', 'approach',
#  'language', 'processing', 'machine', 'algorithms', 'decision', 'trees',
#  'systems', 'hard', 'rules', 'similar', 'hand', 'rules', 'speech', 'tagging',
#  'use', 'markov', 'models', 'natural', 'language', 'processing', 'research',
#  'statistical', 'models', 'soft', 'probabilistic', 'decisions', 'weights',
#  'features', 'input', 'data', 'cache', 'language', 'models', 'speech',
#  'recognition', 'systems', 'examples', 'statistical', 'models', 'models',
#  'robust', 'unfamiliar', 'input', 'input', 'errors', 'common', 'real', 'world',
#  'data', 'reliable', 'results', 'larger', 'system', 'multiple',
#  'subtasks.many', 'notable', 'early', 'successes', 'field', 'machine',
#  'translation', 'ibm', 'research', 'complicated', 'statistical', 'models',
#  'systems', 'able', 'advantage', 'multilingual', 'textual', 'corpora',
#  'parliament', 'canada', 'european', 'union', 'result', 'laws', 'translation',
#  'governmental', 'proceedings', 'official', 'languages', 'systems',
#  'government', 'systems', 'corpora', 'tasks', 'systems', 'major', 'limitation',
#  'success', 'systems', 'result', 'great', 'deal', 'research', 'methods',
#  'limited', 'amounts', 'data.recent', 'research', 'focused', 'unsupervised',
#  'supervised', 'learning', 'algorithms', 'algorithms', 'data', 'hand',
#  'answers', 'combination', 'annotated', 'non', 'annotated', 'data', 'task',
#  'difficult', 'learning', 'accurate', 'results', 'input', 'data', 'enormous',
#  'non', 'annotated', 'data', 'available', 'things', 'entire', 'content',
#  'world', 'wide', 'web', 'inferior', 'results', 'algorithm', 'low', 'time',
#  'complexity', 'practical.in', 'representation', 'deep', 'neural', 'network',
#  'style', 'machine', 'learning', 'methods', 'widespread', 'natural',
#  'language', 'processing', 'flurry', 'results', 'techniques', 'state', 'art',
#  'results', 'natural', 'language', 'tasks', 'example', 'language', 'modeling',
#  'parsing', 'popular', 'techniques', 'use', 'word', 'embeddings', 'semantic',
#  'properties', 'words', 'increase', 'end', 'end', 'learning', 'higher',
#  'level', 'task', 'question', 'answering', 'pipeline', 'separate',
#  'intermediate', 'tasks', 'speech', 'tagging', 'dependency', 'parsing',
#  'areas', 'shift', 'substantial', 'changes', 'nlp', 'systems', 'deep',
#  'neural', 'network', 'approaches', 'new', 'paradigm', 'distinct',
#  'statistical', 'natural', 'language', 'processing', 'instance', 'term',
#  'neural', 'machine', 'translation', 'nmt', 'fact', 'deep', 'learning',
#  'approaches', 'machine', 'translation', 'sequence', 'sequence',
#  'transformations', 'need', 'intermediate', 'steps', 'word', 'alignment',
#  'language', 'modeling', 'statistical', 'machine', 'translation', 'smt']

# the output


# now we can see many duplicates, lets get rid of them using the 'set' function
output1 = set(get_topwords('''Natural language processing (NLP) is a subfield of
linguistics, computer science, information engineering, and artificial
intelligence concerned with the interactions between computers and human
(natural) languages, in particular how to program computers to process and
analyze large amounts of natural language data.Challenges in natural language
processing frequently involve speech recognition, natural language
understanding, and natural-language generation.The history of natural language
processing (NLP) generally started in the 1950s, although work can be found
from earlier periods. In 1950, Alan Turing published an article titled
"Computing Machinery and Intelligence" which proposed what is now called the
```

Turing test as a criterion of intelligence[clarification needed].The Georgetown experiment in 1954 involved fully automatic translation of more than sixty Russian sentences into English. The authors claimed that within three or five years, machine translation would be a solved problem. However, real progress was much slower, and after the ALPAC report in 1966, which found that ten-year-long research had failed to fulfill the expectations, funding for machine translation was dramatically reduced. Little further research in machine translation was conducted until the late 1980s when the first statistical machine translation systems were developed.Some notably successful natural language processing systems developed in the 1960s were SHRDLU, a natural language system working in restricted "blocks worlds" with restricted vocabularies, and ELIZA, a simulation of a Rogerian psychotherapist, written by Joseph Weizenbaum between 1964 and 1966. Using almost no information about human thought or emotion, ELIZA sometimes provided a startlingly human-like interaction. When the "patient" exceeded the very small knowledge base, ELIZA might provide a generic response, for example, responding to "My head hurts" with "Why do you say your head hurts?".During the 1970s, many programmers began to write "conceptual ontologies", which structured real-world information into computer-understandable data. Examples are MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), QUALM (Lehnert, 1977), Politics (Carbonell, 1979), and Plot Units (Lehnert 1981). During this time, many chatterbots were written including PARRY, Racter, and Jabberwacky.Up to the 1980s, most natural language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language processing. This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language processing. Some of the earliest-used machine learning algorithms, such as decision trees, produced systems of hard if-then rules similar to existing hand-written rules. However, part-of-speech tagging introduced the use of hidden Markov models to natural language processing, and increasingly, research has focused on statistical models, which make soft, probabilistic decisions based on attaching real-valued weights to the features making up the input data. The cache language models upon which many speech recognition systems now rely are examples of such statistical models. Such models are generally more robust when given unfamiliar input, especially input that contains errors (as is very common for real-world data), and produce more reliable results when integrated into a larger system comprising multiple subtasks.Many of the notable early successes occurred in the field of machine translation, due especially to work at IBM Research, where successively more complicated statistical models were developed. These systems were able to take advantage of existing multilingual textual corpora that had been produced by the Parliament of Canada and the European Union as a result of laws calling for the translation of all governmental proceedings into all official languages of the corresponding systems of government. However, most other systems depended on corpora specifically developed for the tasks implemented by these systems, which was (and often continues to be) a major limitation in the success of these systems. As a result, a great deal of research has gone into methods of more effectively learning from limited amounts of data.Recent research has

```
more effectively learning from limited amounts of data.Recent research has
increasingly focused on unsupervised and semi-supervised learning algorithms.
Such algorithms can learn from data that has not been hand-annotated with the
desired answers or using a combination of annotated and non-annotated data.
Generally, this task is much more difficult than supervised learning, and
typically produces less accurate results for a given amount of input data.
However, there is an enormous amount of non-annotated data available
(including, among other things, the entire content of the World Wide Web),
which can often make up for the inferior results if the algorithm used has a
low enough time complexity to be practical.In the 2010s, representation
learning and deep neural network-style machine learning methods became
widespread in natural language processing, due in part to a flurry of results
showing that such techniques can achieve state-of-the-art results in many
natural language tasks, for example in language modeling, parsing, and many
others. 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).'''))


print(output1)
```

```
{'recognition', 'work', 'russian', 'shrdlu', 'plot', 'ontologies', 'notable', 'gradual',
```

```
# # the output of the previous cell

# 'lessening', 'time', 'wilensky', 'task', 'research', 'history', 'proceedings',
# 'language', '1970s', 'linguistics', 'sets', 'corpora', 'steps', 'parsing',
# 'dominance', 'thought', 'year', 'slower', 'algorithm', 'understandable',
# 'joseph', 'ibm', 'textual', 'earlier', 'available', 'sequence', 'chomskyan',
# 'unfamiliar', 'words', 'margie', 'computers', 'grammar', 'pipeline',
# 'parliament', 'real', 'understanding', 'data', 'techniques', 'non',
# 'combination', 'chatterbots', 'shrdlu', 'successes', 'term', 'small', 'larger',
# 'complicated', 'large', 'success', 'authors', 'focused', 'system',
# 'multilingual', 'properties', 'politics', 'automatic', 'alignment',
# 'intelligence', 'machine', 'higher', 'emotion', 'pam', 'cache', 'european',
# 'word', '1980s', 'robust', 'nlp', 'dependency', 'expectations', 'union',
# 'style', 'response', 'probabilistic', 'new', 'art', 'fact', 'vocabularies',
# 'semantic', 'problem', 'conceptual', 'end', 'transformations', 'knowledge',
# 'world', 'head', 'systems', 'sam', 'units', 'shift', 'reliable', 'substantial',
# 'revolution', 'embeddings', 'talespin', 'turing', 'english', 'machinery',
# 'patient', 'annotated', 'markov', 'answers', 'tagging', 'processing', 'level',
# 'popular', 'algol', 'like', 'weights', 'years', 'programmers',
```

```
# 'developed.some', 'cullingford', 'theories', 'neural', 'modeling',
# 'data.challenges', 'little', 'meehan', 'use', 'models', 'limitation', 'areas',
# 'interactions', 'successful', 'multiple', 'similar', 'need', 'human',
# 'algorithms', 'government', 'parry', 'content', 'progress', 'carbonell',
# 'rules', 'article', 'early', 'increase', 'long', 'low', 'supervised',
# 'amounts', 'criterion', 'common', 'smt', '1950s', 'computational',
# 'decision', 'lehnert', 'deal', 'translation', 'laws', 'russian', 'flurry',
# 'able', 'rogerian', 'late', 'worlds', 'introduction', 'recognition',
# 'widespread', 'sentences', 'plot', 'sort', 'examples', 'georgetown',
# 'experiment', 'corpus', 'example', 'transformational', 'moore', 'steady',
# 'input', 'approaches', 'statistical', 'law', 'gradual', 'theoretical',
# 'instance', 'notable', 'languages', 'speech', 'science', 'hand', 'hard',
# 'report', 'official', 'difficult', 'changes', 'psychotherapist', 'inferior',
# 'schank', 'results', 'enormous', 'computer', 'result', 'ontologies', '1960s',
# 'underpinnings', 'things', 'soft', 'complexity', 'information', 'deep', 'field',
# 'blocks', 'intelligence[clarification', 'nmt', 'governmental', 'major',
# 'limited', 'separate', 'complex', 'natural', 'weizenbaum', 'great',
# 'data.recent', 'subfield', 'accurate', 'practical.in', 'jabberwacky.up',
# 'representation', 'needed].the', 'alan', 'particular', 'generic',
# 'interaction', 'approach', 'learning', 'unsupervised', 'engineering',
# 'errors', 'wide', 'paradigm', 'entire', 'answering', 'state', 'restricted',
# 'test', 'advantage', 'web', 'eliza', 'power', 'racter', 'features', 'qualm',
# 'subtasks.many', 'network', 'question', 'distinct', 'artificial', 'periods',
# 'methods', 'trees', 'simulation', 'decisions', 'intermediate', 'tasks',
# 'work', 'base', 'canada'}


# sort by frequency (get the top 10 words)
# using Counter module that has most_common function

freq = [x[0] for x in Counter(output).most_common(10)]

print(' '.join(freq))
```

```
language natural machine processing systems translation data research statistical models
```