COMP90042

Workshop Week 02

Workshops

- 10 sessions in total

Mon 11-12pm	Alice Hoy-108
Mon 6:15-7:15pm	Alice Hoy-108
Mon 7:15-8:15pm	Alice Hoy-108 *
Tues 10-11am	Alice Hoy-222 *
Tues 6:15-7:15pm	Alice Hoy-108 *
Wed 8-9am	Alice Hoy-109
Fri 1-2pm	Alice Hoy-222
Fri 3:15-4:15pm	Alice Hoy-210
Fri 5:15-6:15pm	Alice Hoy-211
Fri 6:15-7:15pm	Alice Hoy-222

Questions…

- Post on the LMS discussion board

- Trevor / Daniel
  - t.cohn@unimelb.edu.au / d.beck@unimelb.edu.au
  - Weekly office hour, Wed *12pm-1pm*, DMD 7.02 (new time)

- My contact
  - Yuan Li
  - yuanl4@student.unimelb.edu.au

Workshop materials

- Published on the website
  - https://trevorcohn.github.io/comp90042/
    - Jupyter notebooks are also there…

- Discussion/Programming/Catch-up/Get ahead, etc.
  - Discussion: our main focus, but may not cover all
    - Some parts of the solutions are shown in my slides
    - The official full solutions will be released after 1 or 2 weeks
  - Programming/Catch-up/Get ahead, etc.
    - No solutions will be released

Python

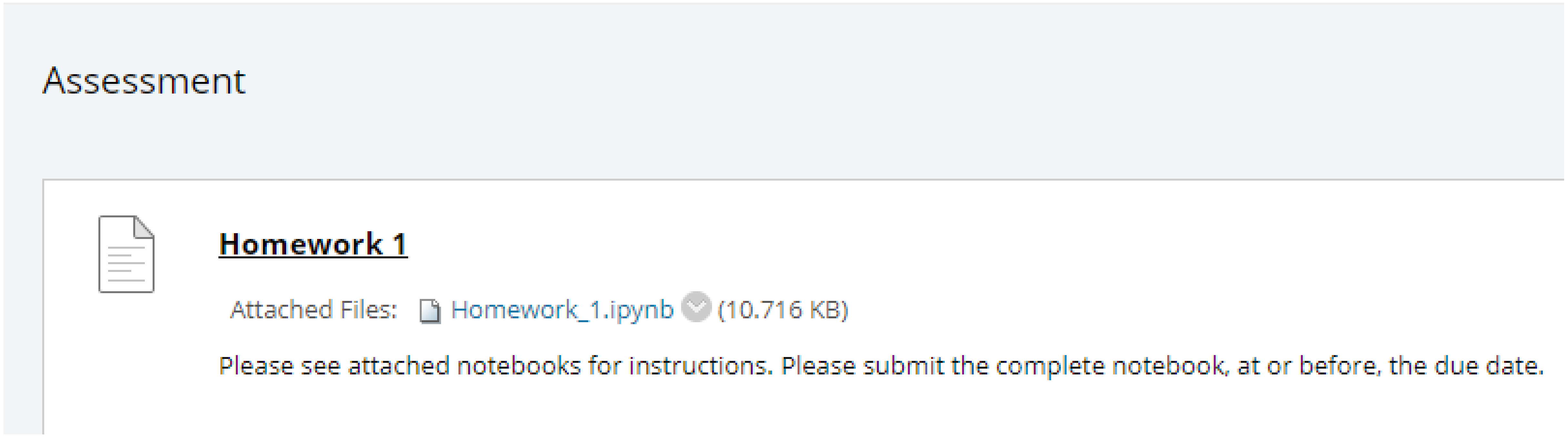
- ❑ Python 2.7 and 3.4 / 3.5 are officially supported by NLTK
- ❑ Python 3.6+ may not be compatible with NLTK

- ❑ *Python 2.7 is the recommended version*

Homework 1 released

- Due data: 11pm, Sunday March 18th
- We accept submissions written in Python 2.7 or 3.5
  - But 2.7 is still the recommended version

- LMS -> Assessment



Assessment

Homework 1

Attached Files: Homework_1.ipynb (10.716 KB)

Please see attached notebooks for instructions. Please submit the complete notebook, at or before, the due date.

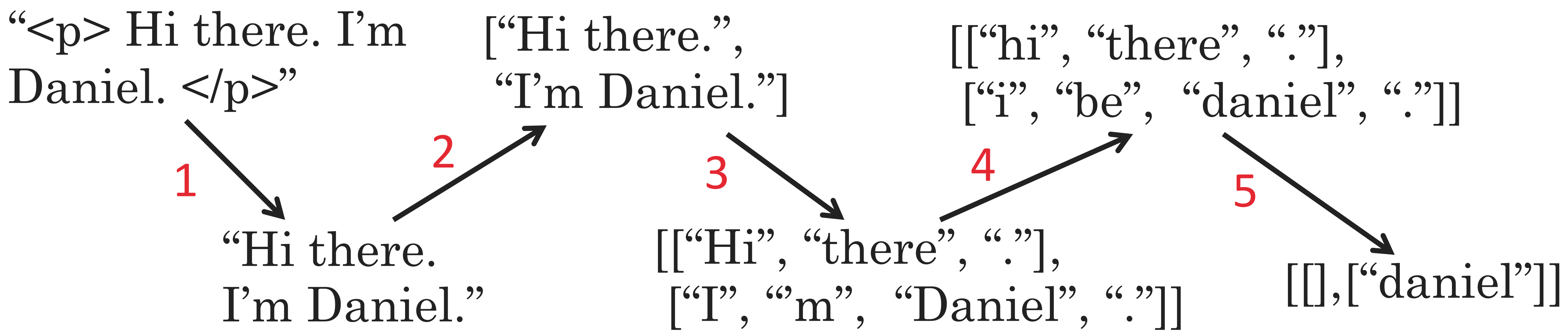
Syllabus

1	Introduction and Preprocessing	Text classification
2	Lexical semantics	Distributional semantics
3	Part of Speech Tagging	Probabilistic Sequence Modelling
4	Probabilistic Sequence Modelling	Context-Free Grammars
5	Probabilistic Parsing	Dependency parsing
	Easter holiday break	
6	N-gram language modelling	Deep learning for language models and tagging
7	Information Extraction	Question Answering
8	Topic Models	*ANZAC day holiday*
9	Information Retrieval -- Boolean search and the vector space model	Indexing and querying in the vector space model, evaluation
10	Index and vocabulary compression	Efficient query processing
11	The Web as a Graph: Page-rank & HITS	Machine Translation (word based)
12	Machine translation (phrase based) and neural encoder-decoder	Subject review

Outline

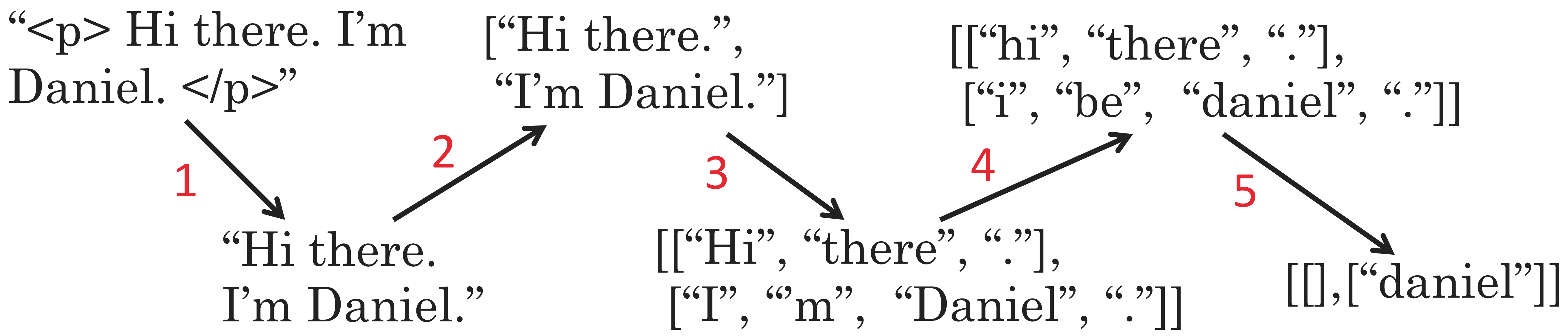
- `WSTA_N1B_preprocessing.ipynb`
  - Preprocessing steps
  - Tokenization
    - Stemming / Lemmatization
- Text classification
  - Train/validation/test split
- Use jupyter on lab computers

Text Normalisation


"<p> Hi there. I'm Daniel. </p>"
1
"Hi there. I'm Daniel."
2
["Hi there.", "I'm Daniel."]
3
[["Hi", "there", "."], ["I", "'m", "Daniel", "."]]
4
[["hi", "there", "."], ["i", "be", "daniel", "."]]
5
[[],["daniel"]]

Text Normalisation

- 1 Remove unwanted formatting (e.g. HTML)
- 2 Segment structure (e.g. sentences)
- 3 Tokenise words
- 4 Normalise words
- 5 Remove unwanted words

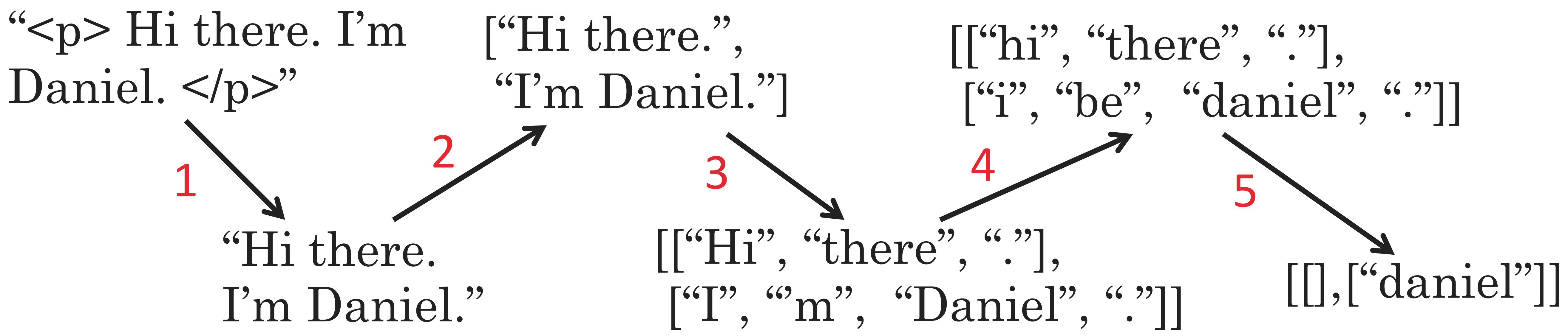


"<p> Hi there. I'm Daniel. </p>"
1
"Hi there. I'm Daniel."
2
["Hi there.", "I'm Daniel."]
3
[["Hi", "there", "."], ["I", "'m", "Daniel", "."]]
4
[["hi", "there", "."], ["i", "be", "daniel", "."]]
5
[[],["daniel"]]

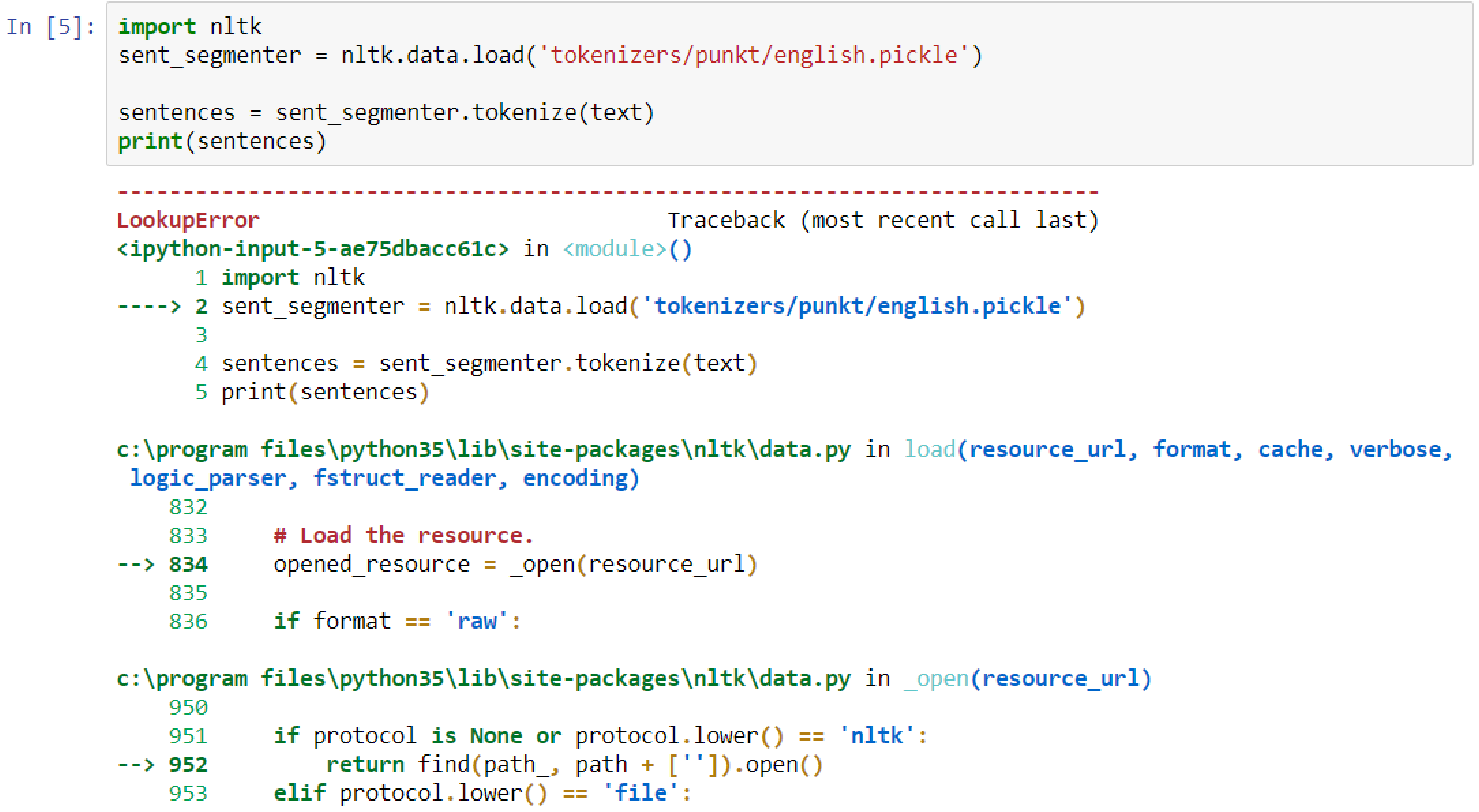
Text Normalisation

- ❑ 1 Remove unwanted formatting (e.g. HTML)
- ❑ 2 Segment structure (e.g. sentences)
- ❑ 3 Tokenise words
- ❑ 4 Normalise words
- ❑ 5 Remove unwanted words

We may not use all of them in practice.

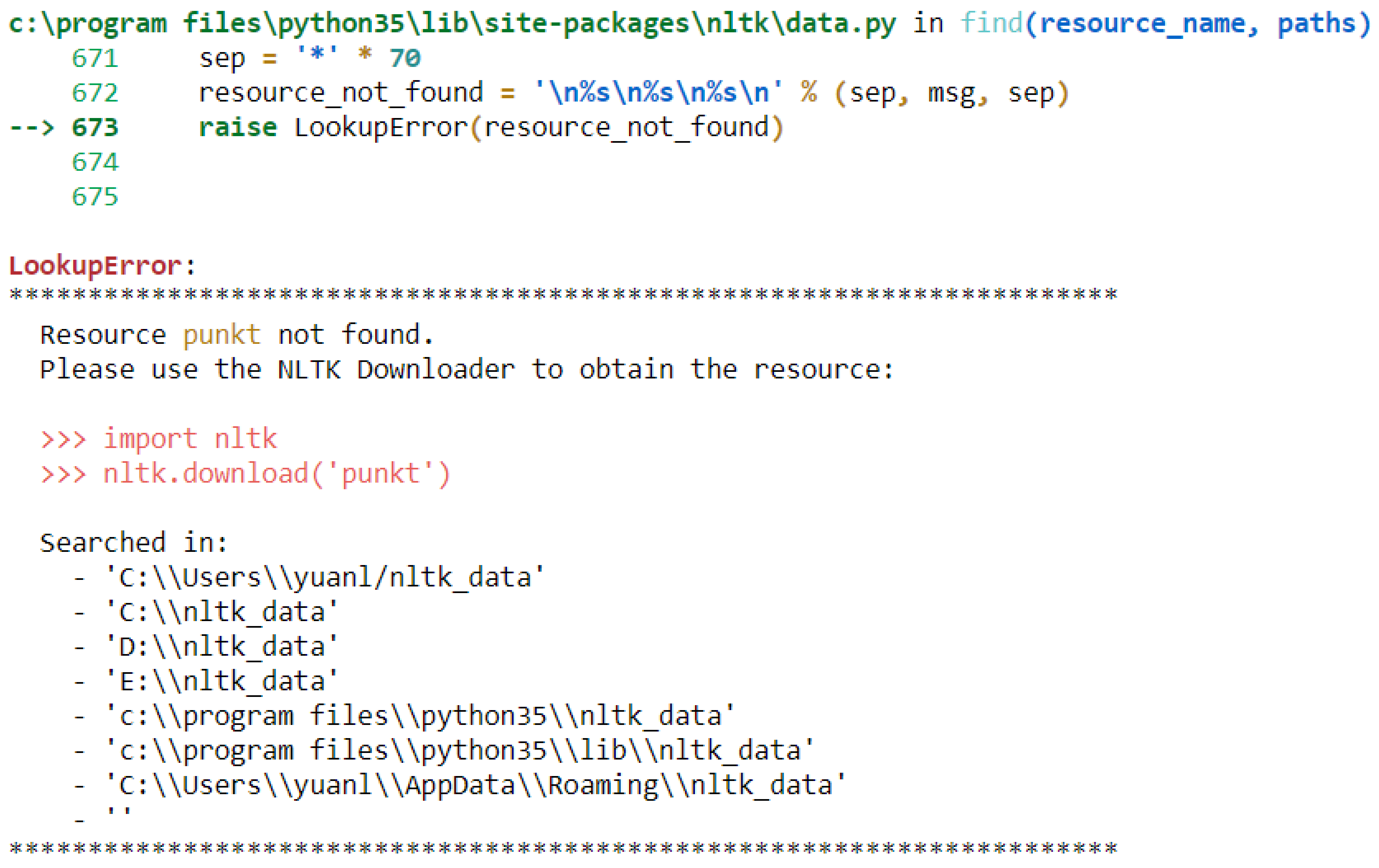


"<p> Hi there. I'm Daniel. </p>"
1
"Hi there. I'm Daniel."
2
["Hi there.", "I'm Daniel."]
3
[["Hi", "there", "."], ["I", "'m", "Daniel", "."]]
4
[["hi", "there", "."], ["i", "be", "daniel", "."]]
5
[[],["daniel"]]

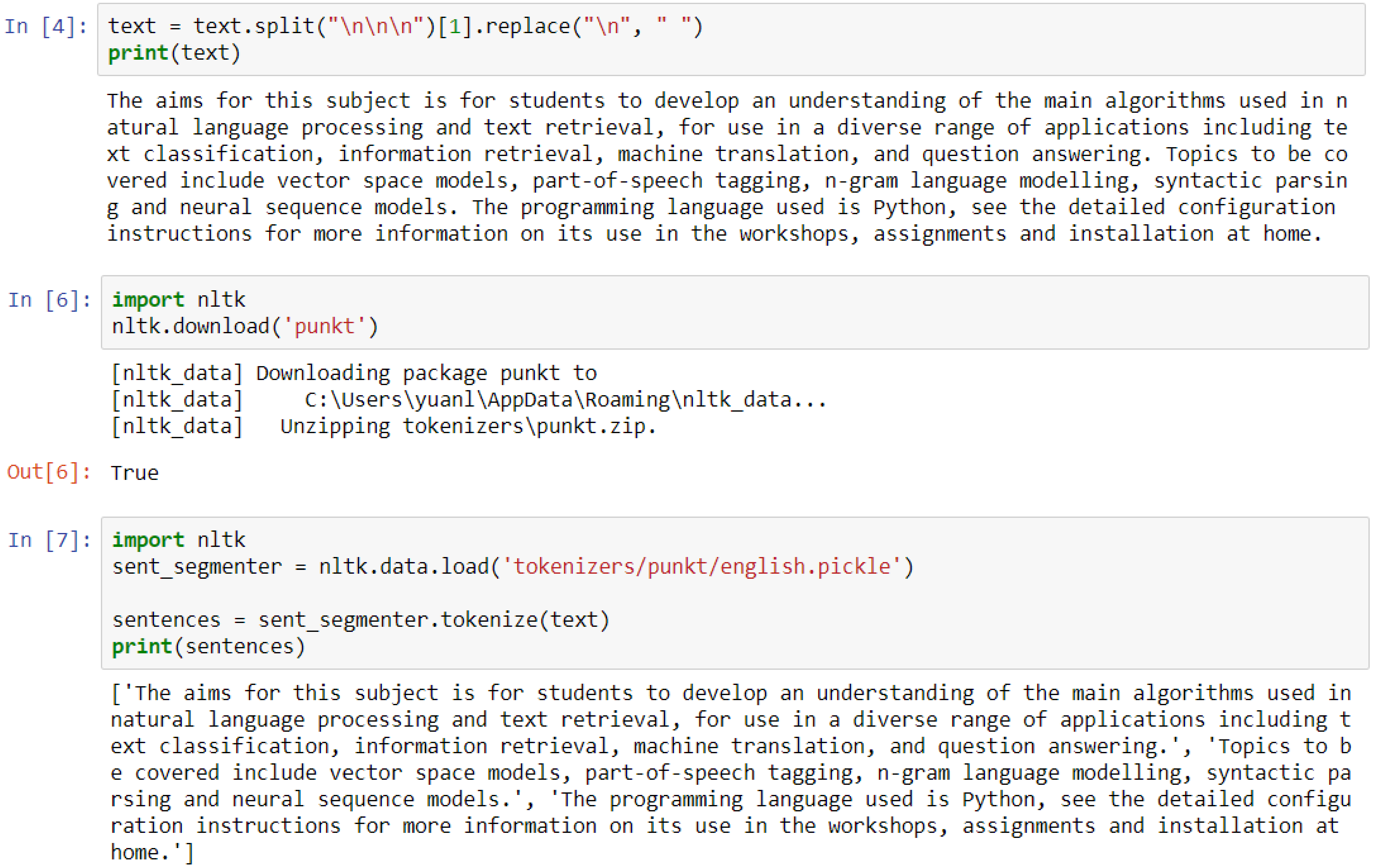


```
In [5]: import nltk
        sent_segmenter = nltk.data.load('tokenizers/punkt/english.pickle')

        sentences = sent_segmenter.tokenize(text)
        print(sentences)
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
<ipython-input-5-ae75dbacc61c> in <module>()
      1 import nltk
----> 2 sent_segmenter = nltk.data.load('tokenizers/punkt/english.pickle')
      3 
      4 sentences = sent_segmenter.tokenize(text)
      5 print(sentences)

c:\program files\python35\lib\site-packages\nltk\data.py in load(resource_url, format, cache, verbose,
 logic_parser, fstruct_reader, encoding)
    832 
    833     # Load the resource.
--> 834     opened_resource = _open(resource_url)
    835 
    836     if format == 'raw':

c:\program files\python35\lib\site-packages\nltk\data.py in _open(resource_url)
    950 
    951     if protocol is None or protocol.lower() == 'nltk':
--> 952         return find(path_, path + ['']).open()
    953     elif protocol.lower() == 'file':
```



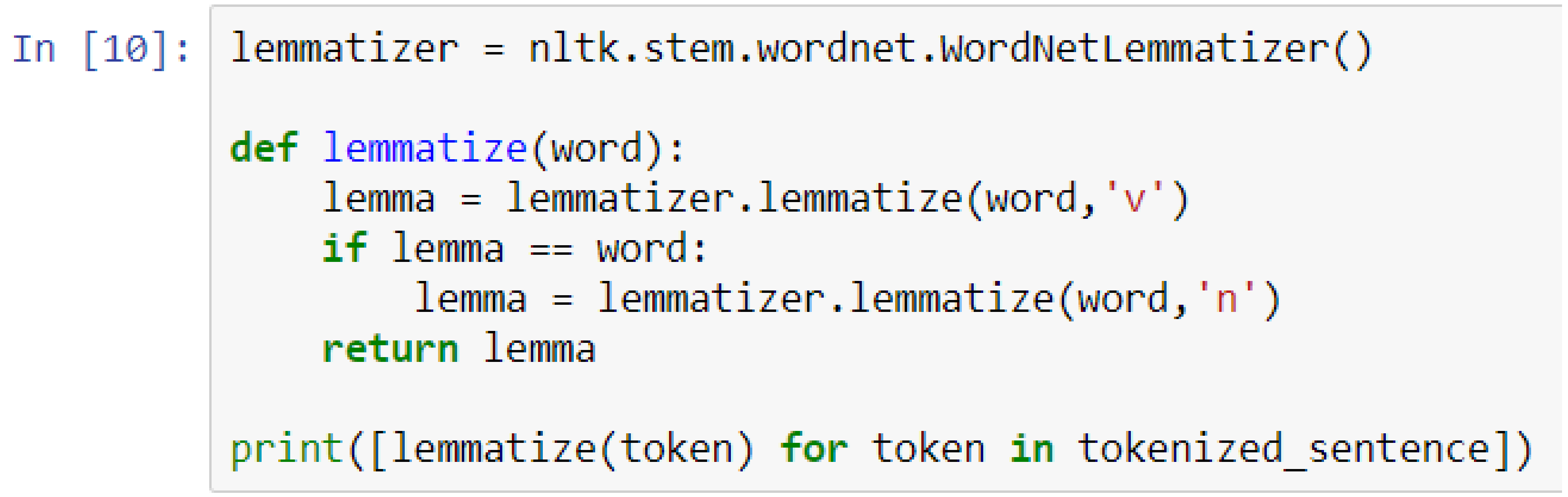
```
c:\program files\python35\lib\site-packages\nltk\data.py in find(resource_name, paths)
    671     sep = '*' * 70
    672     resource_not_found = '\n%s\n%s\n%s\n' % (sep, msg, sep)
--> 673     raise LookupError(resource_not_found)
    674
    675

LookupError:
**********************************************************************
  Resource punkt not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('punkt')

  Searched in:
    - 'C:\\Users\\yuanl/nltk_data'
    - 'C:\\nltk_data'
    - 'D:\\nltk_data'
    - 'E:\\nltk_data'
    - 'c:\\program files\\python35\\nltk_data'
    - 'c:\\program files\\python35\\lib\\nltk_data'
    - 'C:\\Users\\yuanl\\AppData\\Roaming\\nltk_data'
    - ''
**********************************************************************
```



```
In [4]: text = text.split("\n\n\n")[1].replace("\n", " ")
        print(text)

        The aims for this subject is for students to develop an understanding of the main algorithms used in n
        atural language processing and text retrieval, for use in a diverse range of applications including te
        xt classification, information retrieval, machine translation, and question answering. Topics to be co
        vered include vector space models, part-of-speech tagging, n-gram language modelling, syntactic parsin
        g and neural sequence models. The programming language used is Python, see the detailed configuration
        instructions for more information on its use in the workshops, assignments and installation at home.
```

```
In [6]: import nltk
        nltk.download('punkt')

        [nltk_data] Downloading package punkt to
        [nltk_data]     C:\Users\yuanl\AppData\Roaming\nltk_data...
        [nltk_data]   Unzipping tokenizers\punkt.zip.

Out[6]: True
```

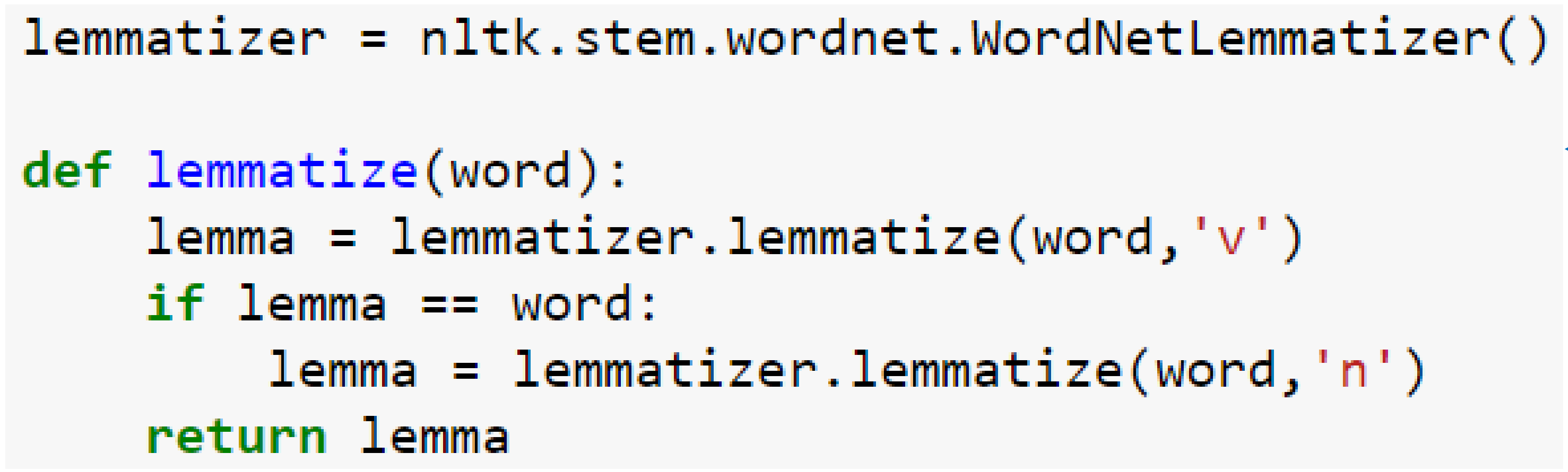
```
In [7]: import nltk
        sent_segmenter = nltk.data.load('tokenizers/punkt/english.pickle')

        sentences = sent_segmenter.tokenize(text)
        print(sentences)

        ['The aims for this subject is for students to develop an understanding of the main algorithms used in
        natural language processing and text retrieval, for use in a diverse range of applications including t
        ext classification, information retrieval, machine translation, and question answering.', 'Topics to b
        e covered include vector space models, part-of-speech tagging, n-gram language modelling, syntactic pa
        rsing and neural sequence models.', 'The programming language used is Python, see the detailed configu
        ration instructions for more information on its use in the workshops, assignments and installation at
        home.']
```



```
In [10]: lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()

         def lemmatize(word):
             lemma = lemmatizer.lemmatize(word,'v')
             if lemma == word:
                 lemma = lemmatizer.lemmatize(word,'n')
             return lemma

         print([lemmatize(token) for token in tokenized_sentence])
```

- We encourage you to *reuse the code snippets* in the provided ipynb files.

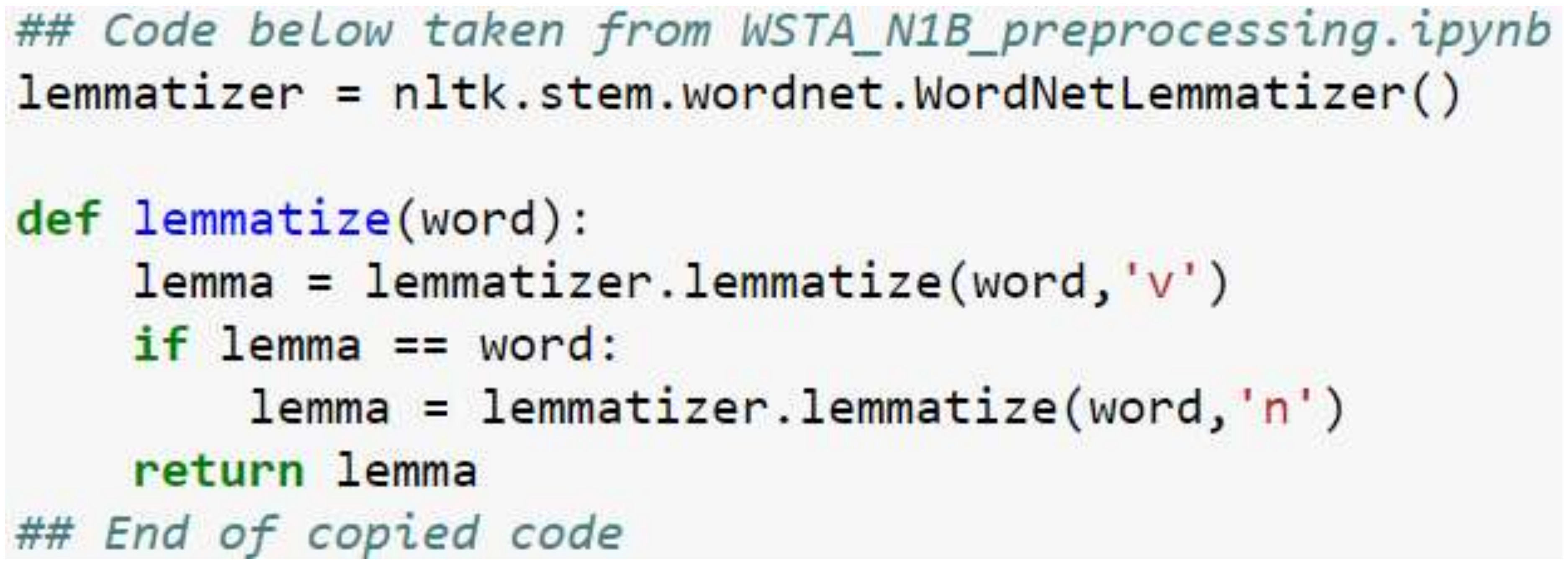
When using code from notebooks…



```python
lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()

def lemmatize(word):
    lemma = lemmatizer.lemmatize(word,'v')
    if lemma == word:
        lemma = lemmatizer.lemmatize(word,'n')
    return lemma
```

Code from WSTA_N1B_preprocessing.ipynb

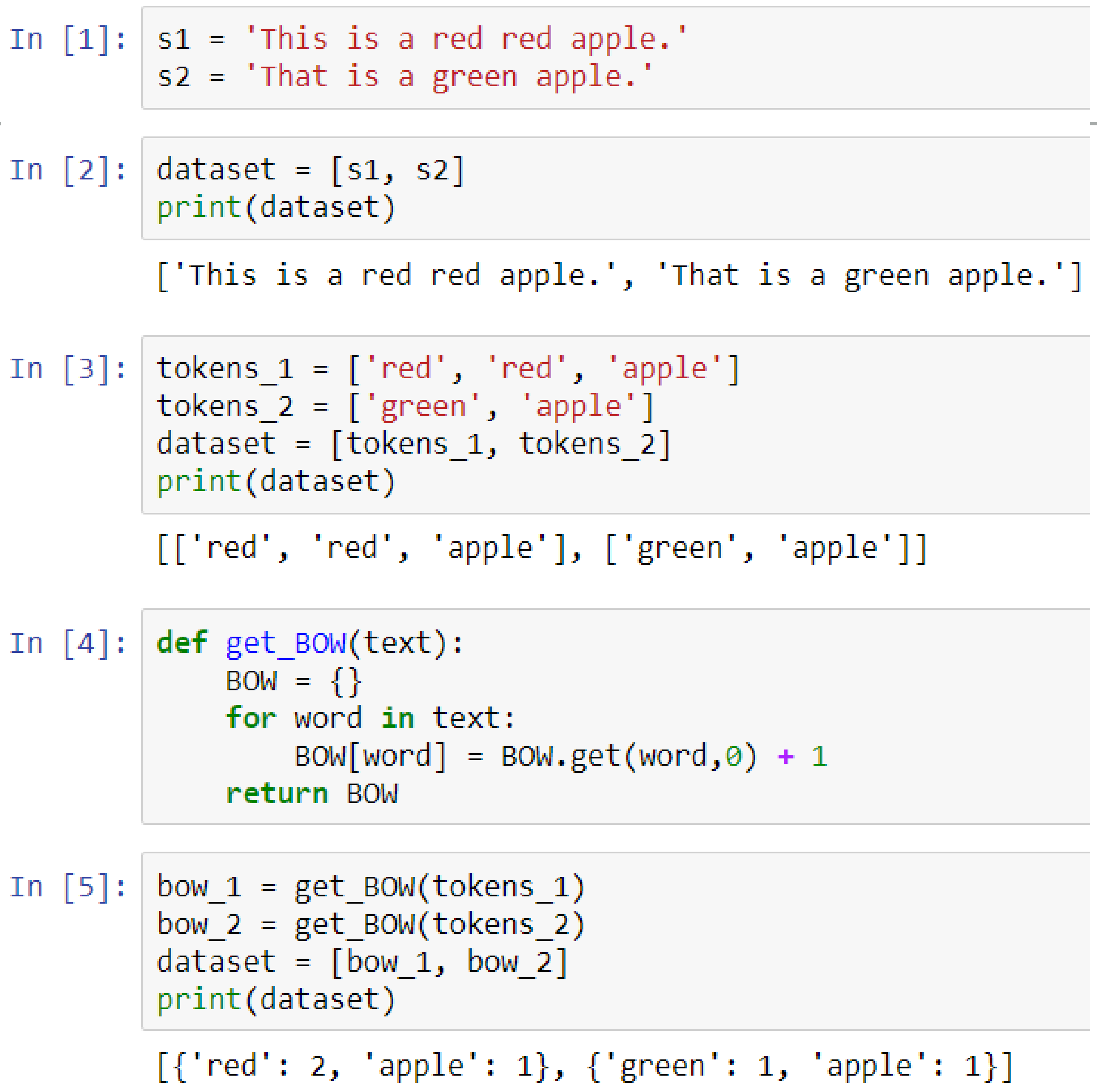
- According to Trevor's reply on LMS: … *indicate with comments what code is not original*, at the top and bottom of the snippet, and attribute the source clearly…



```python
## Code below taken from WSTA_N1B_preprocessing.ipynb
lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()

def lemmatize(word):
    lemma = lemmatizer.lemmatize(word,'v')
    if lemma == word:
        lemma = lemmatizer.lemmatize(word,'n')
    return lemma
## End of copied code
```

Outline

- `WSTA_N1B_preprocessing.ipynb`
  - Preprocessing steps
  - Tokenization
    - Stemming / Lemmatization
- Text classification
  - Bag of words representation & feature matrix
  - Train/validation/test split
- Use jupyter on lab computers



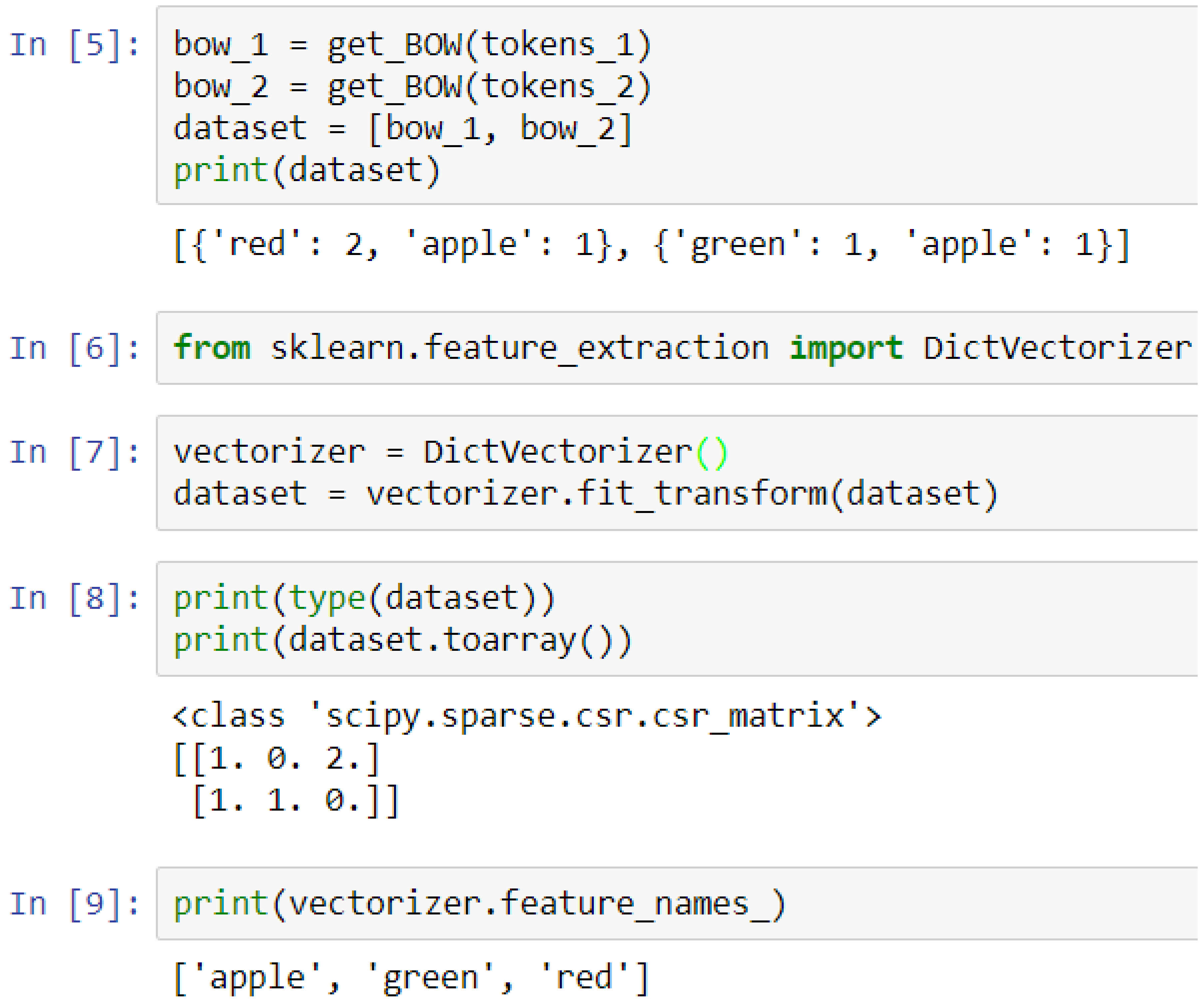
```
In [1]: s1 = 'This is a red red apple.'
        s2 = 'That is a green apple.'
```

```
In [2]: dataset = [s1, s2]
        print(dataset)

        ['This is a red red apple.', 'That is a green apple.']
```

```
In [3]: tokens_1 = ['red', 'red', 'apple']
        tokens_2 = ['green', 'apple']
        dataset = [tokens_1, tokens_2]
        print(dataset)

        [['red', 'red', 'apple'], ['green', 'apple']]
```

```
In [4]: def get_BOW(text):
            BOW = {}
            for word in text:
                BOW[word] = BOW.get(word,0) + 1
            return BOW
```

```
In [5]: bow_1 = get_BOW(tokens_1)
        bow_2 = get_BOW(tokens_2)
        dataset = [bow_1, bow_2]
        print(dataset)

        [{'red': 2, 'apple': 1}, {'green': 1, 'apple': 1}]
```



```
In [5]: bow_1 = get_BOW(tokens_1)
        bow_2 = get_BOW(tokens_2)
        dataset = [bow_1, bow_2]
        print(dataset)

        [{'red': 2, 'apple': 1}, {'green': 1, 'apple': 1}]

In [6]: from sklearn.feature_extraction import DictVectorizer

In [7]: vectorizer = DictVectorizer()
        dataset = vectorizer.fit_transform(dataset)

In [8]: print(type(dataset))
        print(dataset.toarray())

        <class 'scipy.sparse.csr.csr_matrix'>
        [[1. 0. 2.]
         [1. 1. 0.]]

In [9]: print(vectorizer.feature_names_)

        ['apple', 'green', 'red']
```

Train/validation/test split

- Dataset
  - Preprocessing / normalization / feature selection
  - Split into train*/test, where test served as the held-out set
    - Split train* into train/validation (or *k* folds train/validation, CV)

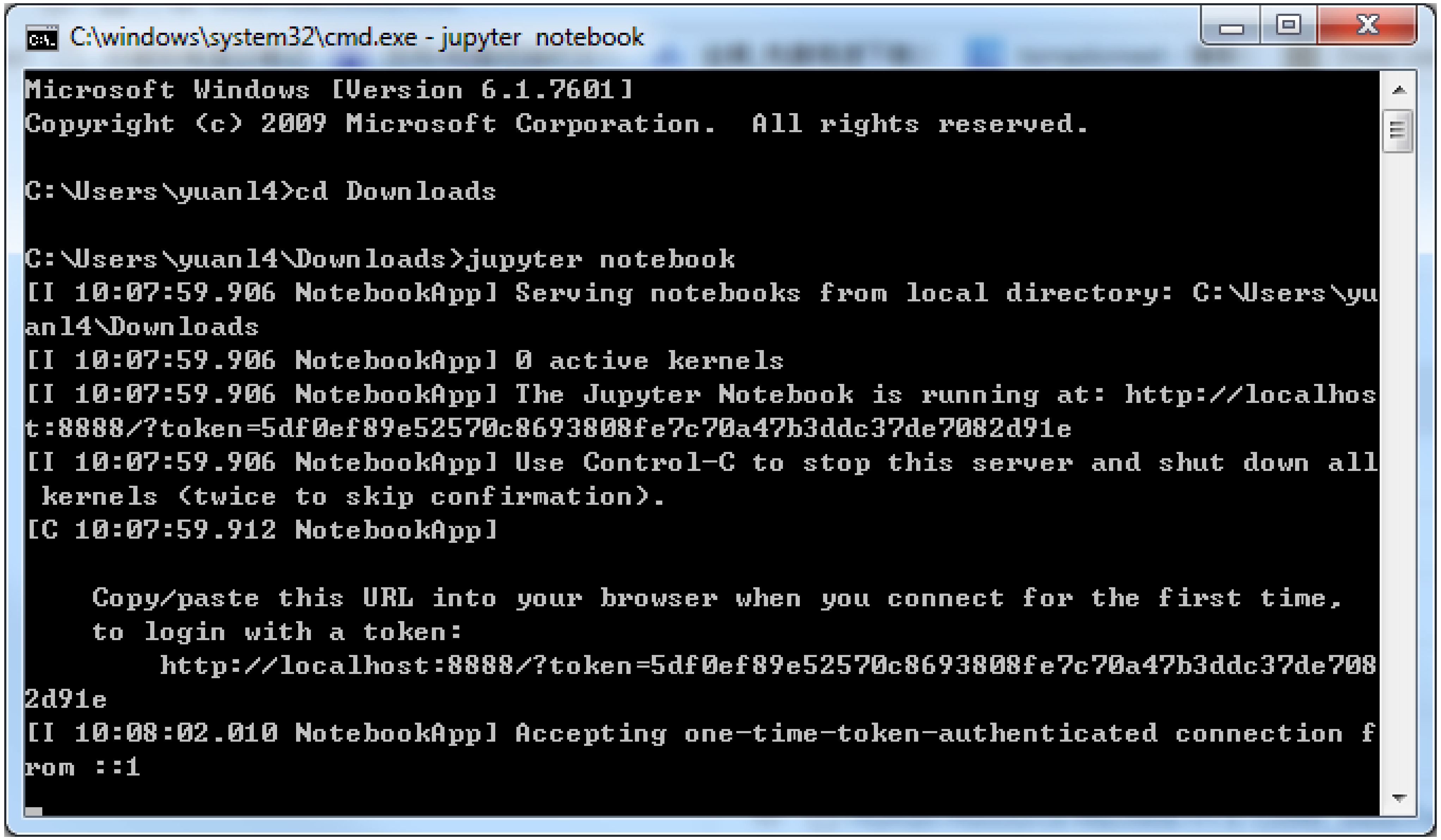
- Example:
- Train: exam papers of 2015-2016
- Development: the exam paper of 2017
- Test: the final exam of this year (2018)

Outline

- ❑ `WSTA_N1B_preprocessing.ipynb`
  - ❑ Preprocessing steps
  - ❑ Tokenization
    - ❑ Stemming / Lemmatization
- ❑ Text classification
  - ❑ Bag of words representation & feature matrix
  - ❑ Train/ validation/test split
- ❑ Use jupyter on lab computers

To launch jupyter on the lab computer.

- ❑ Open a command line prompt
- ❑ “cd” to your working directory
- ❑ Type “jupyter notebook”



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\yuanl4>cd Downloads

C:\Users\yuanl4\Downloads>jupyter notebook
[I 10:07:59.906 NotebookApp] Serving notebooks from local directory: C:\Users\yu
anl4\Downloads
[I 10:07:59.906 NotebookApp] 0 active kernels
[I 10:07:59.906 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/?token=5df0ef89e52570c8693808fe7c70a47b3ddc37de7082d91e
[I 10:07:59.906 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 10:07:59.912 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=5df0ef89e52570c8693808fe7c70a47b3ddc37de708
2d91e
[I 10:08:02.010 NotebookApp] Accepting one-time-token-authenticated connection f
rom ::1
_
```

jupyter is installed, but is not in PATH

- ❑ Windows users (now in `C:\Users\yuanl4\Downloads`)

```
C:\Users\yuanl4\Downloads>where python
C:\Program Files\Python35\python.exe

C:\Users\yuanl4\Downloads>
"C:\Program Files\Python35\Scripts\jupyter.exe"
notebook
```

- ❑ Linux users (now in `~/comp90051-2017`)

```
yuanl4@slug:~/comp90051-2017$ which python3
/home/yuanl4/python35env/bin/python3

yuanl4@slug:~/comp90051-2017$
/home/yuanl4/python35env/bin/jupyter notebook
```

jupyter is running, but no browser opened

```
C:\Users\yuanl4\Downloads>jupyter notebook

[I 15:50:13.236 NotebookApp] Serving notebooks
from local directory: C:\Users\yuanl4\Downloads

[I 15:50:13.236 NotebookApp] 0 active kernels

[I 15:50:13.236 NotebookApp] The Jupyter Notebook
is running at:
http://localhost:8888/?token=8a45ae92166791fbe4868
f6575ca958bf6ff3c300df3ab1c

[I 15:50:13.236 NotebookApp] Use Control-C to stop
this server and shut down all kernels (twice to
skip confirmation).

...
```