108-NLP @ NCTU

e-Lab 1

Boaz

Good morning

Zoom!

Please change your name to StudentID + English name

3 hours is a very long time... we'll take a break or two ☕

How to get my attention

- Raise hand (please remember to un-raise)
- Chat

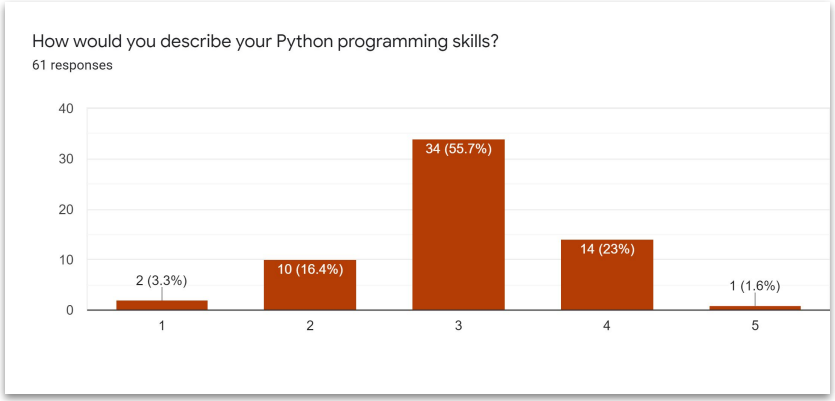
Communications in English only. Thank you!

Survey - thank you for participating!

84% Master Students (also undergrads, PhD students)

98% have Python experience

36% most comfortable with Python



Outline

Google Colab

Tokenization

Bag-of-Words

Similarity

News article recommendation

Programming assignment

Google Colab

Run Python code in the cloud

Use Google hardware, including GPUs

Easy to share code

Free!

https://colab.research.google.com

Google Colab

Integration with Google Drive

Code cells, text cells, scratch cells

Moving cells

Importing libraries

System shortcuts

```
!ls
!pip freeze
```

“Magics”

Tokenizer

Divide text into smaller units ("words")

```
>> s = "The quick brown fox jumps over the lazy dog" /* pangram */
>> tokenize(s)

["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]
```

Building our own tokenizer

Demo

Sentence Tokenizer

Split a document into a list of sentences

Is using “.” to split sentences good enough?

So we add ! and ?

Enough...

What about **?!**

Also, “I love my M.Sc. studies in NCTU!”

Bag-of-Words

There is someone at the door
Put the book on the table
...

Vocabulary →	someone	book	door	the	is	put	table	film	there
Index →	0	1	2	3	4	5	6	7	8
Document ↓									
There is someone at the door	1	0	1	1	1	0	0	0	1
Put the book on the table	0	1	0	1	0	1	1	0	0

Bag-of-Words

Demo

Similarity

Demo

Issues with our program...

Stopwords

Punctuation

Uppercase/lowercase

BoW

Short documents

Lab 1 programming homework

Task: a better news recommendation engine!

1. Remove punctuation
2. Convert all tokens to lowercase
3. Remove stopwords (https://bit.ly/nlp-stopwords)
4. Replace BoW vectors with TF-IDF vectors
   (use 1000 top features by term frequency across the whole corpus)
5. Instead of title, use content

Program from scratch: do not use any NLP or machine learning libraries

Warning: the system automatically detects plagiarism

Lab 1 programming homework

How we grade the assignment

- Correctness: runs correctly when using “Runtime/Restart and run all…”
- Clarity: logical, good documentation, meaningful variable names
- Zero tolerance for cheating/plagiarizing
- Submit by end-of-Thursday (23:59)

Submissions:

(Optional: start with a copy of the **`lab1-reuters.ipynb`** notebook)

Rename notebook to **`lab1-<studentID>.ipynb`** (for example, **`lab1-1234567.ipynb`**)

1. Submit Colab notebook link (https://colab.research.google.com/…)

   Make sure the Colab notebook is shared for viewing!

2. Upload Python file **`lab1-<studentID>.py`** (for example, **`lab1-1234567.py`**)
   (In Colab, use “File/Download .py”)
3. Submit **title** of your own seed story + **titles** of its 5 most similar stories
4. Note: your seed story is corpus[StudentID % 1000]

Tips

To speed up running time during development use:

- Smaller corpus
  - For example, instead of
    - `corpus = df['title'].to_list()`
  - Use
  - `corpus = df['title'].to_list()[0:200]`
- Fewer features
  - For example, instead of
    - `vocab = get_vocab(corpus).most_common(1000)`
  - Use
    - `vocab = get_vocab(corpus).most_common(100)`

Don’t forget to revert before submitting

Reminder: TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents