

# 자연어처리 프로젝트 2차발표

2조

2013210043 권민규

2014210035 전수혁

2014210064 변지석

2019/May/22

# Python Document Search

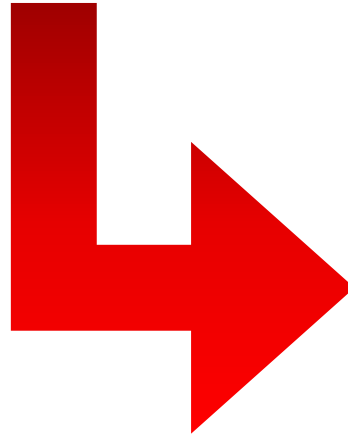
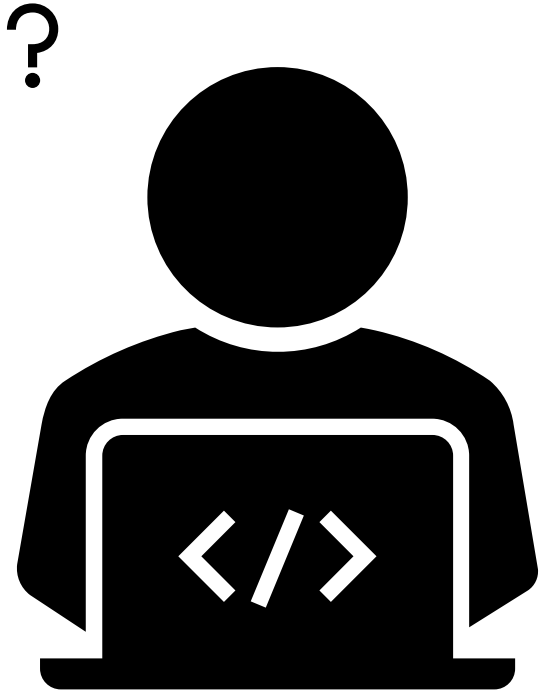
## 파이썬 문서 검색기

# Why Python Document Search

I'm pretty new to Python and only want to extract the city for these clients' addresses:

```
clients = ["Peter, Calle Fantasia 15, Madrid", "Robert, Plaza de Perdas 2,  
Sevilla", "Paul, Calle Polo, Madrid", "Francesco, Plaza de Opo I, Segovia"]
```

Can someone help? Thank you very much in advance!



**stackoverflow**

```
[i.split(',')[ -1].strip() for i in clients]  
# ['Madrid', 'Sevilla', 'Madrid', 'Segovia']
```

Answer URL

<https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions>

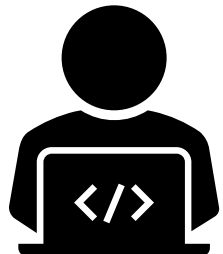
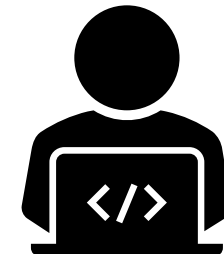
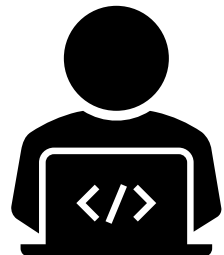
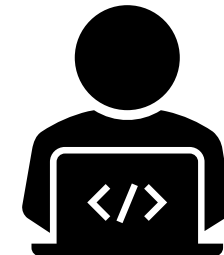
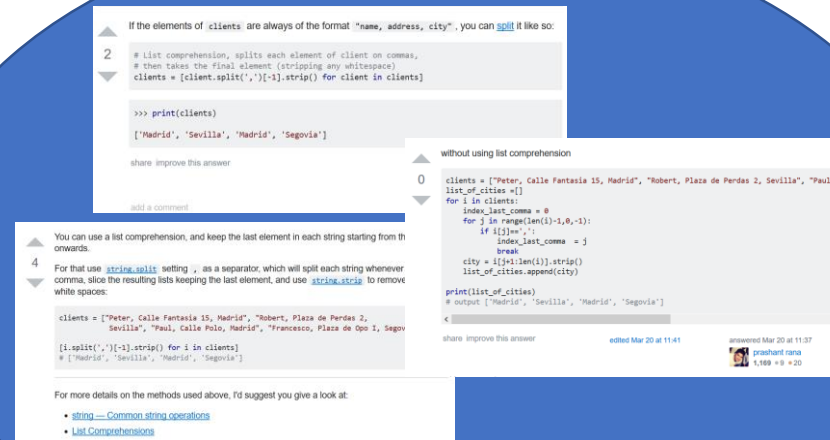
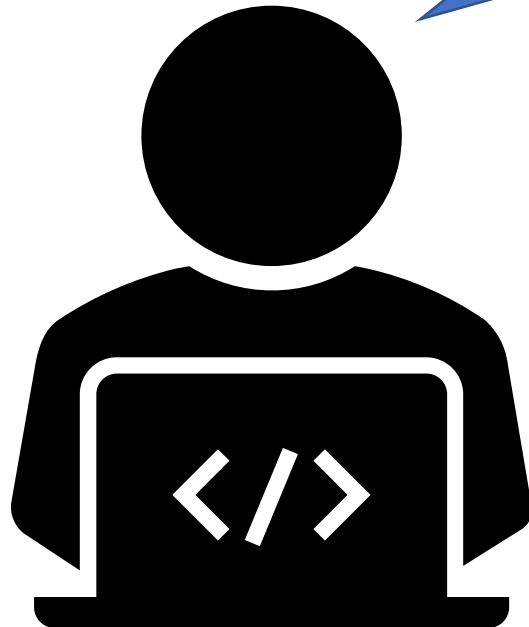
<https://docs.python.org/3/library/stdtypes.html#str.split>

# Goal

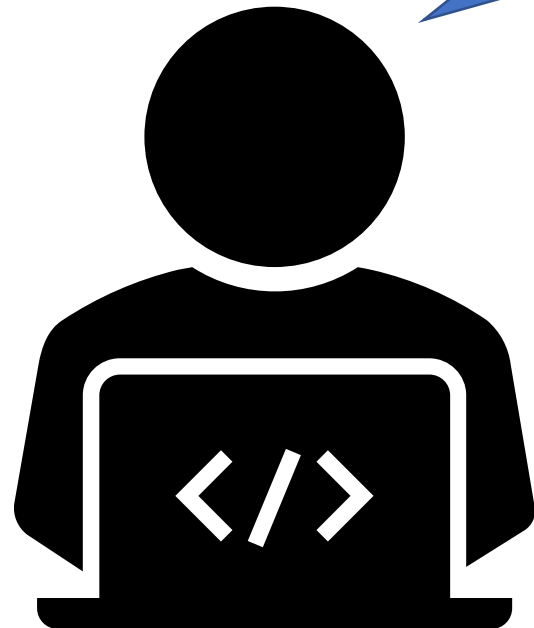
[Python] Extract certain values from a list



stackoverflow



# Goal



[Python] Extract certain values from a list

Look at these URLs

<https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions>

<https://docs.python.org/3/library/stdtypes.html#str.split>

stackoverflow



# Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - Web Service
- Evaluation
- Future Works

# Contents

- **Stackoverflow Python Q&A Dataset Review.**
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - Web Service
- Evaluation
- Future Works

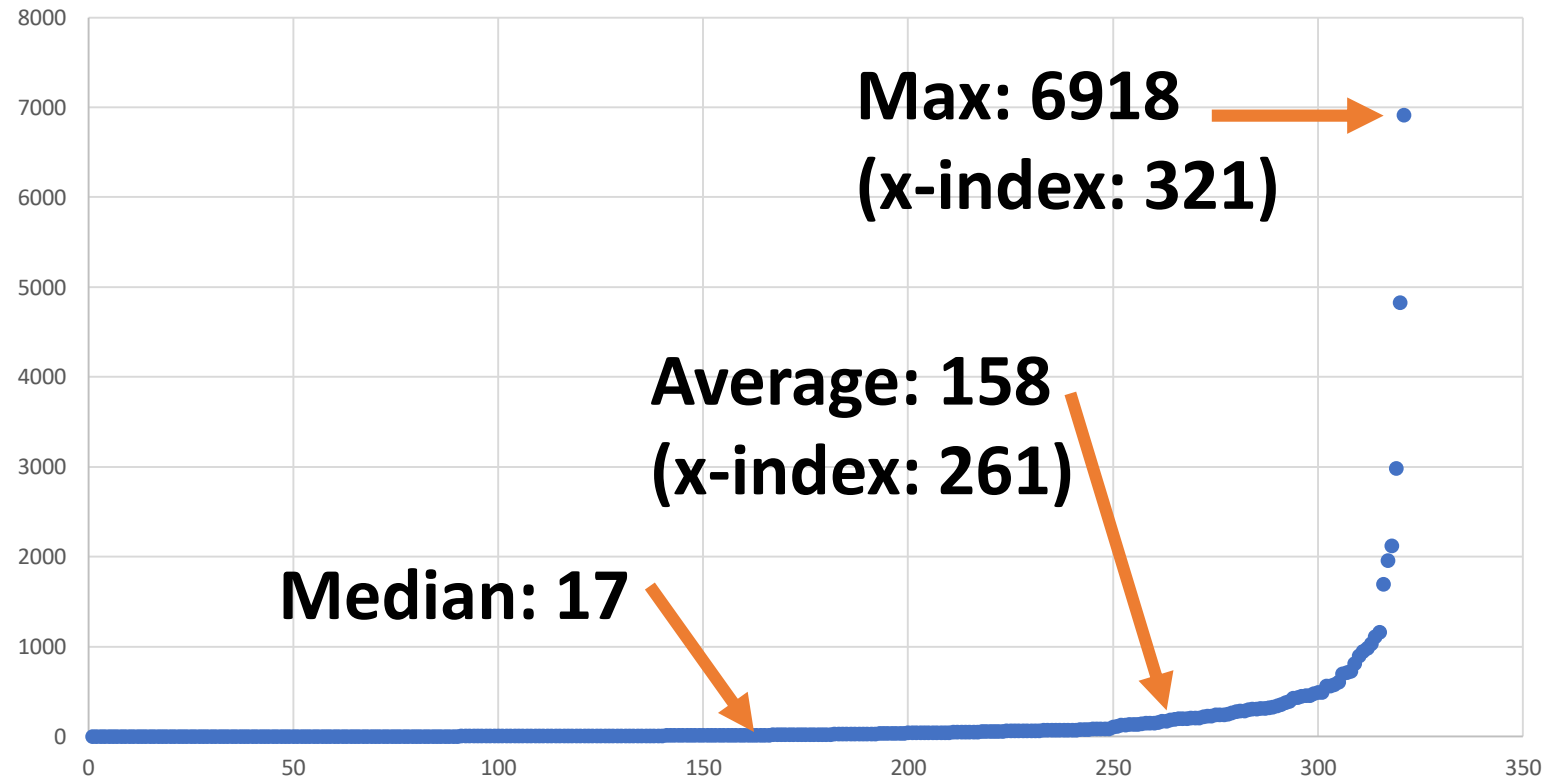
# Stackoverflow Python Q&A Dataset

- **47491 Q&As, 87 MB size**
- Some are duplicated Q&A (redirected question)
- Constraints
  - [Python] tagged Q&A questions.
  - The page should contain the link to “docs.python.org/3/”
- You can download this dataset at
  - <https://github.com/cushionbadak/PyMaker/tree/master/PyMaker/datas/object3>
- Unrefined data is also available at
  - <https://github.com/cushionbadak/PyMaker/tree/master/PyMaker/datas/object2>
- Stackoverflow’s all user contributions are licensed under [Creative Commons Attribution-Share Alike](#).



# Stackoverflow Python Q&A Dataset

- Chart: The Number of times mentioned for each link.
  - X – index number of each link
  - Y – # of times
- Small & Biased



# Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - **Data mining and Preprocessing**
    - Word/Document Vector Representation
    - Web Service
- Evaluation
- Future Works

# Previous(April) Data mining Works

- Construct Stackoverflow Python Q&A Dataset
- We need to collect **Python Document itself** too.

# Collecting Python Document - contents

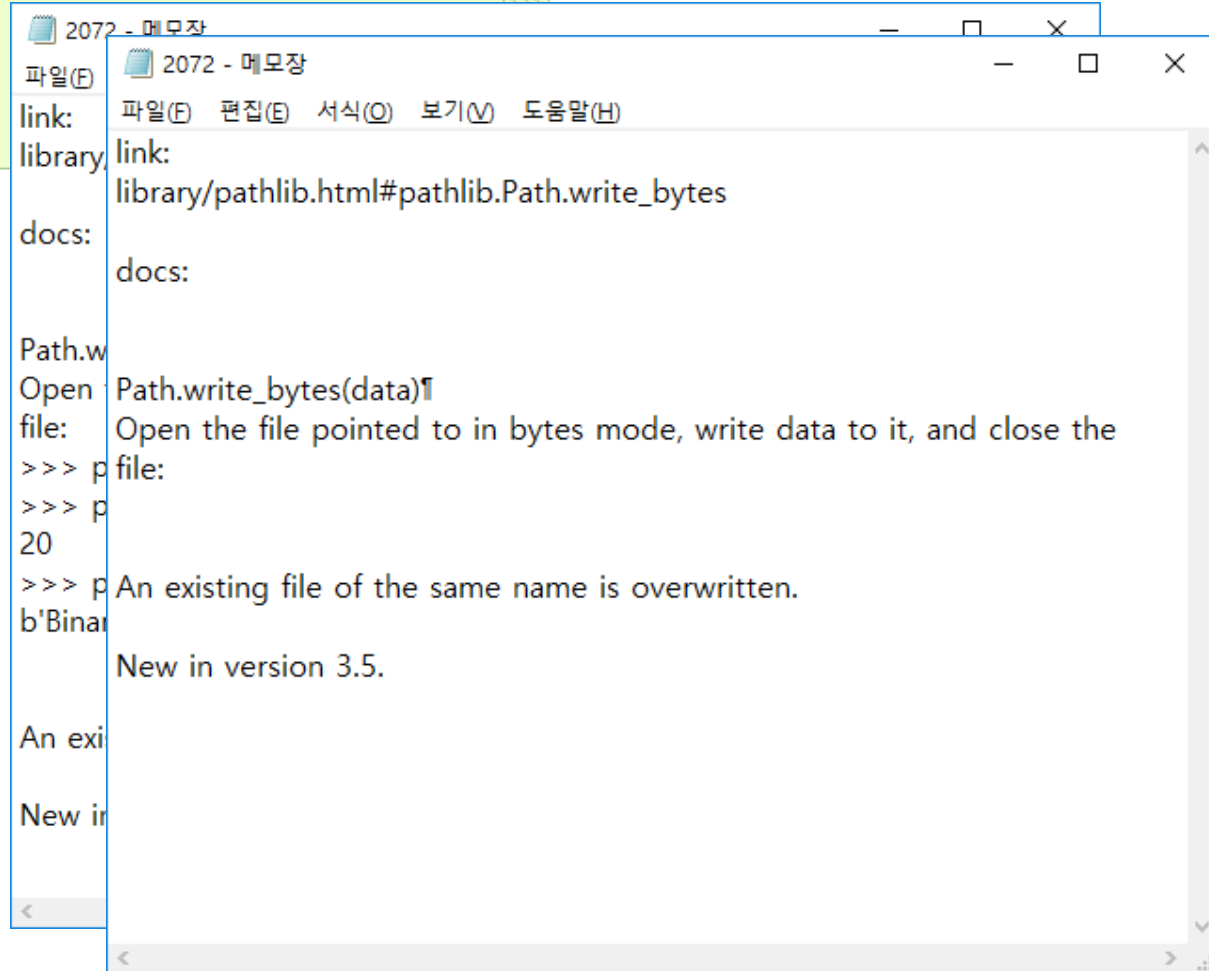
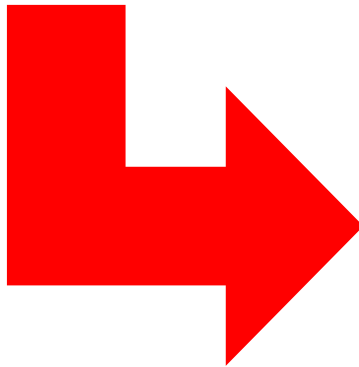
## Path.write\_bytes(data)

Open the file pointed to in bytes mode, write *data* to it, and close the file:

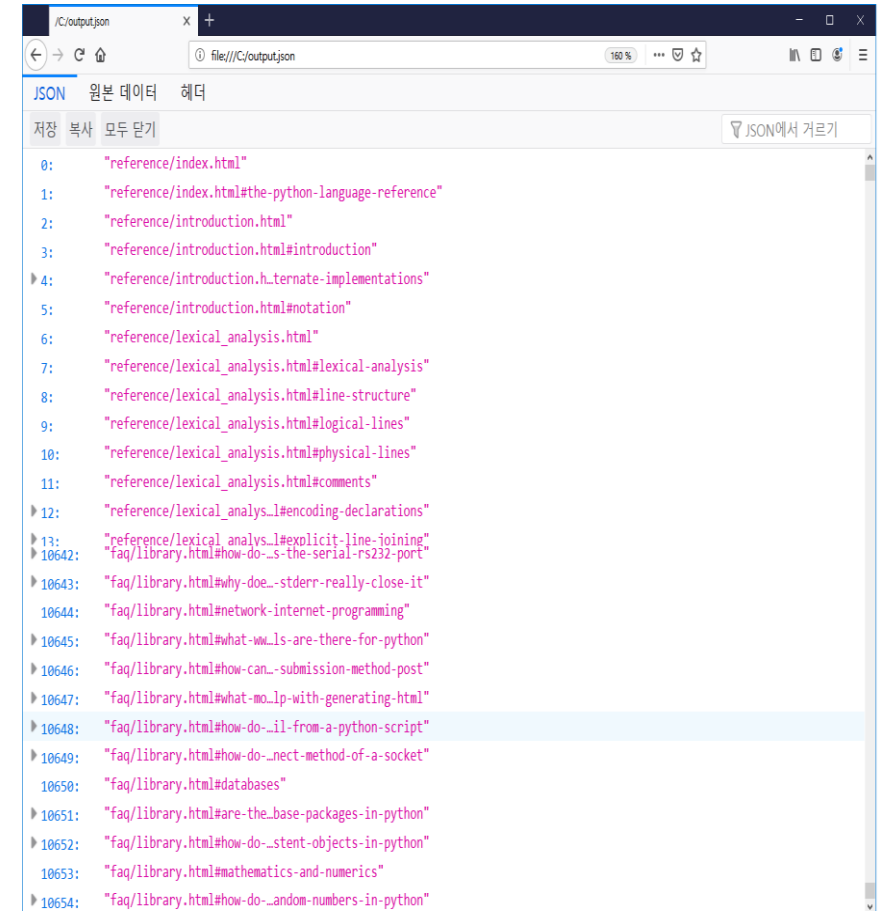
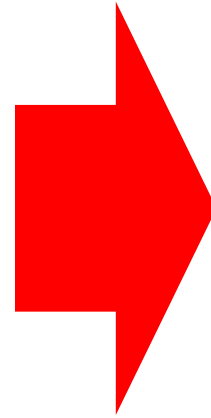
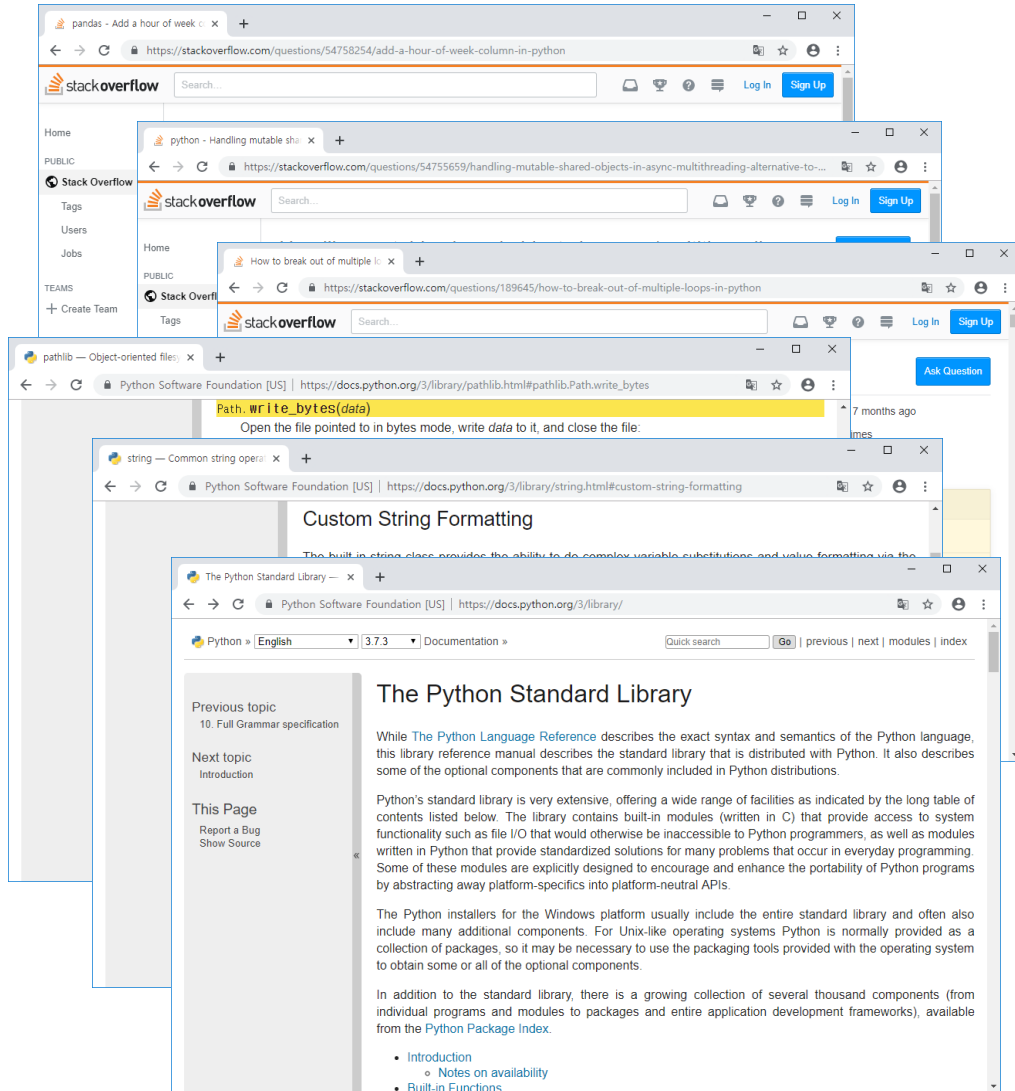
```
>>> p = Path('my_binary_file')
>>> p.write_bytes(b'Binary file contents')
20
>>> p.read_bytes()
b'Binary file contents'
```

An existing file of the same name is overwritten.

*New in version 3.5.*



# Collecting Python Document - URLs



# Collecting Python Document

- **10655 Documents, 28 MB size**
- Some are duplicated contents
  - For example, [https://docs.python.org/3/reference/lexical\\_analysis.html#keywords](https://docs.python.org/3/reference/lexical_analysis.html#keywords)
  - contents are included in [https://docs.python.org/3/reference/lexical\\_analysis.html](https://docs.python.org/3/reference/lexical_analysis.html)
- Constraints
  - We only collected URLs and content at <https://docs.python.org/3/library/> and <https://docs.python.org/3/reference/>
- You can download this dataset at
  - <https://github.com/cushionbadak/PyMaker/tree/master/PyMaker/datas/object4>
- Licenses for Python documentation are located at the following links: <https://docs.python.org/3/license.html>

# Contents

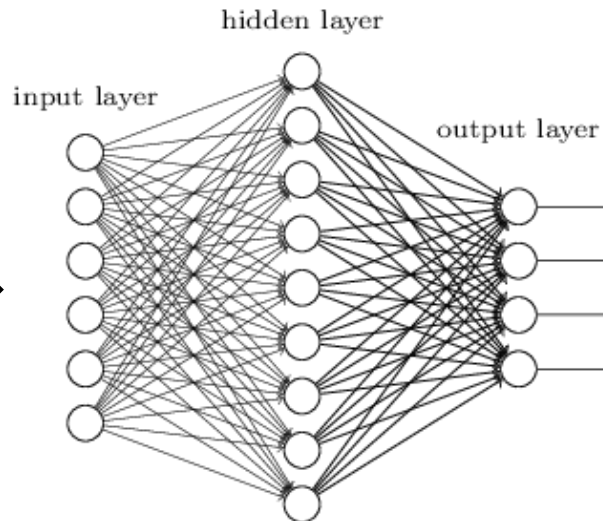
- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - **Word/Document Vector Representation**
  - Web Service
- Evaluation
- Future Works

# First Attempt (Failed)

- We first believed that bi-gram sentence classification would solve our problem.

**Bi-gram representation**

**hash('Hello-Python') →**



**Scores for each URL**

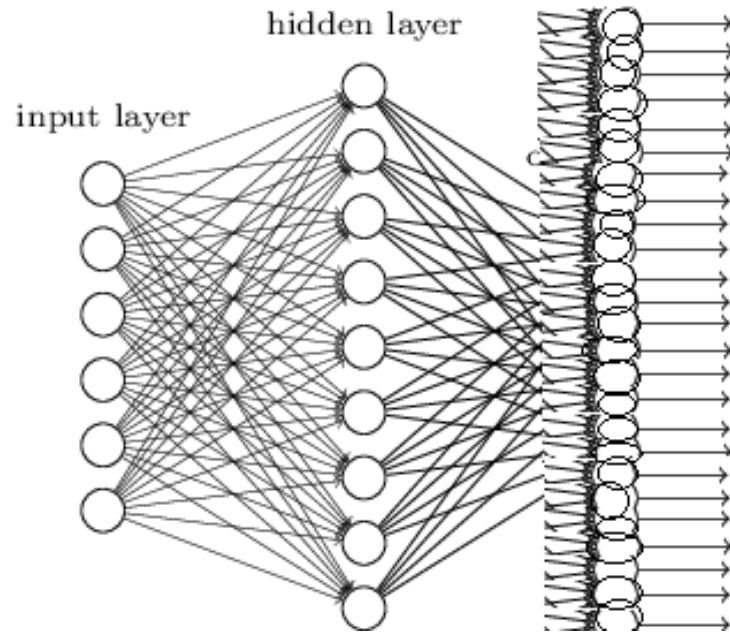
**→ <1.1, 2.0, -0.7, ..., 0.2>**

- Image source: <https://www.extremetech.com/wp-content/uploads/2016/05/tikz35.png>



# First Attempt (Failed)

- There were two big obstacles. One is the size of the classes.



There are **10655** URLs in  
<https://docs.python.org/3>

So we reduced it to **321** URLs

# First Attempt (Failed)

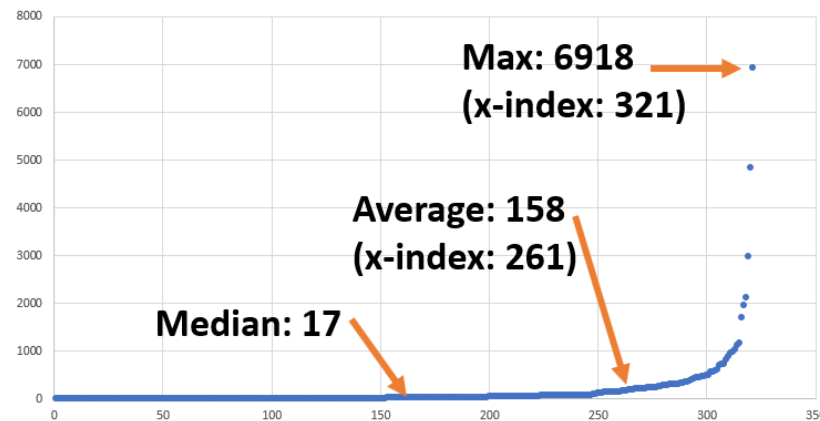
- The other is lack of training data.

## Stackoverflow Python Q&A Dataset

- Chart: The Number of times mentioned for each link.

- X – index number of each link
- Y – # of times

- Small & Biased



# First Attempt (Failed)

```
result_so_zerobase_learned_full_r.log
1 ITEATION 1 => CORRECT: 0 / 0 FILE: break-out-of-a-python-for-loop-using-boolean-flag-logic-and-indentation-error
2 CANDIDATES:
3   library/tkinter.scrolledtext.html
4   library/quopri.html
5   library/array.html
6   reference/simple_stmts.html
7   library/_future_.html
8 ANSWERS:
9
10 ITEATION 2 => CORRECT: 0 / 1 FILE: pygame-load-images-that-change-filename
11 CANDIDATES:
12   library/tkinter.scrolledtext.html
13   library/quopri.html
14   reference/simple_stmts.html
15   library/array.html
16   library/spwd.html
17 ANSWERS:
18   library/os.html
19
20 ITEATION 3 => CORRECT: 0 / 1 FILE: write-filenames-of-different-extention-into-different-text-files
21 CANDIDATES:
22   library/tkinter.scrolledtext.html
23   library/quopri.html
24   reference/simple_stmts.html
25   library/array.html
26   library/email.errors.html
27 ANSWERS:
28   library/functions.html
29
30 ITEATION 4 => CORRECT: 0 / 1 FILE: getting-typeerror-when-attempting-to-open-chrome-using-webbrowser-python
31 CANDIDATES:
32   library/tkinter.scrolledtext.html
33   library/quopri.html
34   reference/simple_stmts.html
35   library/array.html
36   library/email.errors.html
37 ANSWERS:
38   library/webbrowser.html
39
40 ITEATION 5 => CORRECT: 0 / 1 FILE: how-do-i-remove-whitespace-to-balance
41 CANDIDATES:
```

```
result_so_zerobase_learned_full_r.log
5034 library/iterertools.html
5035 library/python.html
5036 library/math.html
5037 ANSWERS:
5038 library/asyncio-eventloop.html
5039 library/asyncio-task.html
5040 library/concurrent.futures.html
5041
5042 ITEATION 498 => CORRECT: 0 / 2 FILE: unexpected-long-decimal-after-converting-to-float
5043 CANDIDATES:
5044 library/faulthandler.html
5045 library/iterertools.html
5046 library/math.html
5047 library/python.html
5048 library/codecs.html
5049 ANSWERS:
5050 library/codecs.html
5051
5052 ITEATION 500 => CORRECT: 0 / 1 FILE: python-unittest-assert-called-with-false-despite-identical-calls
5053 CANDIDATES:
5054 library/faulthandler.html
5055 library/io.html
5056 library/python.html
5057 library/iterertools.html
5058 library/etree.elementtree.html
5059 ANSWERS:
5060 library/mock.html
```

# Second Attempt

- Since our computing resource and data are limited, we decided to use pre-trained word2vec model.
- We use GoogleNews-vectors-negative300.bin (**3.6 GB, 300-dimensional, 3 million words and phrases**, trained with 100 billion Google News dataset)
- The model is available at <https://code.google.com/archive/p/word2vec/>

# Second Attempt

## < Classification Algorithm >

1. Convert every python document into vectors
  - **The vector is obtained by adding all the words in the document as vectors.**

1> 각각의 파이썬 문서에 대해 벡터 표현을 만든다.  
(파이썬 문서를 이루는 각 단어의 벡터 표현을 더해서 구한다.)

# Second Attempt

## < Classification Algorithm >

1. Convert every python document into vectors
    - **The vector is obtained by adding all the words in the document as vectors.**
  2. Convert query string into vector
    - Using the same method as 1
- 2> 질문 문장에 대해서도 1과 똑같은 방법으로 벡터 표현을 만든다.

# Second Attempt

## < Classification Algorithm >

1. Convert every python document into vectors
    - **The vector is obtained by adding all the words in the document as vectors.**
  2. Convert query string into vector
    - Using the same method as 1
  3. Find the most similar python document using **cosine similarity**.
- 3> 코사인 유사도를 이용해서 질문 문장과 가장 유사한 파이썬 문서를 찾는다.

# Second Attempt

< Classification Algorithm >

1. Convert every python document into vectors
  - **The vector is obtained by adding all the words in the document as vectors.**
2. Convert query string into vector
  - Using the same method as 1
3. Find the most similar python document using **cosine similarity**.



# Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - **Web Service**
- Evaluation
- Future Works

# Web Service

- You can run the current version of our document search program at <http://13.125.156.35:8000/Query/> (Opened temporarily.)

## PyMaker

ex) returns the time in seconds

urns the time in seconds

Type: #1 ☒ #2 ☐

Query

# Web Service

- Result Page

## **Python Reference to "returns the time in seconds"**

- <https://docs.python.org/3/library/time.html>
- <https://docs.python.org/3/library/datetime.html>
- <https://docs.python.org/3/library/calendar.html>
- <https://docs.python.org/3/library/sched.html>
- <https://docs.python.org/3/library/asyncio-queue.html>

[New Query](#)

# Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - Web Service
- **Evaluation**
- Future Works

# Evaluation

- There are no well-known benchmarks for python.
- So we measured our algorithm's **classification accuracy** with Stackoverflow Python Q&A Dataset.
- For now, our model successfully classifies **5781 of 41568 (13.9%)**.
  - The model gives **5-candidates** for each given Question-title sentence.
  - 5 Random selection will show only **1.6%** of accuracy.)
- We hope to evaluate the accuracy of other web search engines on this Stackoverflow benchmark.
  - It is hard to gather the result pages of search engines.

# Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - Web Service
- Evaluation
- **Future Works**

# Future Works

- It is too naive to sum word representations to represent document into vector.
  - We'll continue to explore other method for better document-to-vector representation.
- If possible, we'll evaluate our Stackoverflow Q&A benchmark on other search engines.
- We will increase the usability of our Python document search engine.

# Summary

## Contents

- Stackoverflow Python Q&A Dataset Review.
- Actual Works Explained
  - Data mining and Preprocessing
  - Word/Document Vector Representation
  - Web Service
- Evaluation
- Future Works



# Summary

- Creativity
  - It's Python Document Search Engine!
  - We made our own benchmark to evaluate this problem.
- Technical Completeness
  - Document classification: Base 1.6% → Our approach 13.9%
- Contribution
  - We publicly open our source codes and data at github.
  - We provide web services to better understand the project.

Q&A

감사합니다