

## ✓ Summarizing BBC news

In this lab, you will self-study two unsupervised graph-based summarization methods, namely LexRank and TextRank, and apply them to summarize news data.

First of all, download [data](#) and extract files.

```
# importing required modules
from zipfile import ZipFile

with ZipFile('bbc-fulltext.zip', 'r') as zip:
    # printing all the contents of the zip file
    zip.printdir()


    # extracting all the files
    print('Extracting all the files now...')
    zip.extractall()
    print('Done!')
```

 [Show hidden output](#)

Below, Politics news is selected. (Note that you are free to use other categories as you would like i.e. tech, sports, business, and entertainment.)

In the Politics category, there are 417 news articles. The goal is to summarize **each news article**, at least 10 news. The compression ratio should be within 25%-30%.

```
!pip install path
```

 Collecting path  
 Downloading path-17.0.0-py3-none-any.whl.metadata (6.4 kB)  
 Downloading path-17.0.0-py3-none-any.whl (24 kB)  
 Installing collected packages: path  
 Successfully installed path-17.0.0

ผมขออนุญาตเลือกในส่วนของหัวข้อข่าวที่เป็น Technology

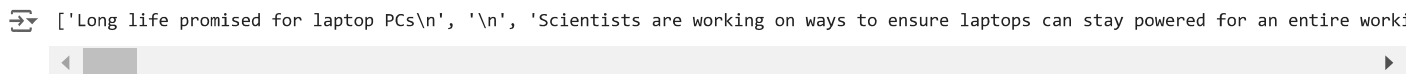
```
# !pip install path
from path import Path
import os

mydir = os.getcwd()

documents = []
documents_dir = Path(mydir+'/bbc/tech')
for file_path in documents_dir.files('*.*txt'):
    with file_path.open(mode='rt', encoding='utf-8') as fp:
        documents.append(fp.readlines())
```

Use sentences in one of the news *as an example*.

```
sentences = documents[0]
print(sentences)
```



Documents ที่เรามีทั้งหมด 13 เอกสาร (Sentences)

```
len(sentences)
```

 13

## ✓ LexRank

**TODO #1:** Study an algorithm of LexRank and describe how it works.

**TODO #2:** Use the LexRank library to summarize data as shown in the example below.

Note: Make sure that, in your final summary the selected sentences must be ordered chronologically.

Reference: [LexRank library](#).

---

ผมอ้างอิงจาก Paper LexRank : <https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume22/erkan04a-html/erkan04a.html#lpr-graph>

LexRank is an unsupervised approach to text summarization based on graph-based centrality scoring of sentences. The main idea is that sentences “recommend” other similar sentences to the reader. Thus, if one sentence is very similar to many others, it will likely be a sentence of great importance. The importance of this sentence also stems from the importance of the sentences “recommending” it. Thus, to get ranked highly and placed in a summary, a sentence must be similar to many sentences that are in turn also similar to many other sentences. This makes intuitive sense and allows the algorithms to be applied to any arbitrary new text.

- สรุปให้ได้ใจความคือ Algorithm ของ LexRank คือการนำ Graph-based มาให้คะแนน โดยถ้าประโยคนั้นเหมือนกับประโยคอื่นมากๆ ประโยคนั้นจะ **สำคัญ**
- โดยที่การวัดความใกล้เคียงของ LexRank จะใช้ Cosine Similarity (idf-modified-cosine) ในการวัด
- เหมาะสำหรับ Summarize แบบ Multi-document summarization เพราะรองรับ datasets ใหญ่ได้

Run LexRank to summarize input document.

```
!pip install lexrank
```


 [Show hidden output](#)

```
from lexrank import STOPWORDS, LexRank
lxr = LexRank(documents, stopwords=STOPWORDS['en'])
```

Get scores of each sentence.

```
# 'fast_power_method' speeds up the calculation, but requires more RAM
scores_cont = lxr.rank_sentences(sentences,
                                threshold=None,
                                fast_power_method=False,)


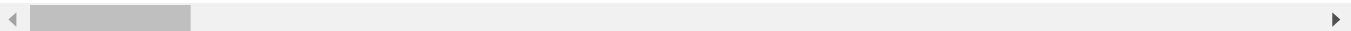
print(scores_cont)
```




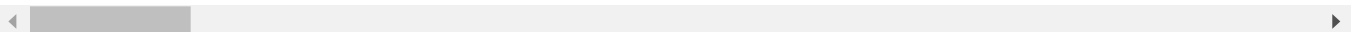
0.85852486	1.	0.81368322	1.	0.98366134	1.
0.96548046	1.	0.98685756	1.	1.13283449	1.
1.25895806					

Print high-ranked sentences.

```
summary = lxr.get_summary(sentences, summary_size=2, threshold=.25)
print(summary)
```


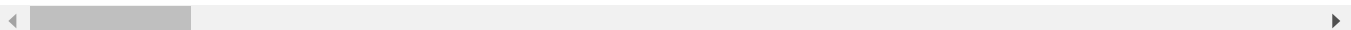
 ['This work has led to the creation of the Mobile PC Extended Battery Life (EBL) Working Group that shares information about buildir  


```
# get summary with continuous LexRank without summary_size and threshold
summary_cont = lxr.get_summary(sentences, threshold=None)
print(summary_cont)
```

 ['This work has led to the creation of the Mobile PC Extended Battery Life (EBL) Working Group that shares information about buildir  


ฉะนั้น Summarize ที่ได้จากหัวข้อ Technology เป็นดังนี้ โดยใช้วิธี Extractive Summarization ด้วย LexRank

```
print(summary_cont[0])
```

 This work has led to the creation of the Mobile PC Extended Battery Life (EBL) Working Group that shares information about building  


## ✓ TextRank

**TODO #3:** Study an algorithm of TextRank and describe how it works.

**TODO #4:** Use the TextRank library to summarize data as shown in the example below.

Note: Make sure that, in your final summary the selected sentences must be ordered chronologically.

Reference: [TextRank library](#)

---

ผมอ้างอิงจาก Paper ของ TextRank นั้นครับ : <https://arxiv.org/pdf/1602.03606>

TextRank is an unsupervised algorithm for the automated summarization of texts that can also be used to obtain the most important keywords in a document

TextRank algorithm using BM25 BM25 / Okapi-BM25 is a ranking function widely used as the state of the art for Information Retrieval tasks. BM25 is a variation of the TF-IDF model using a probabilistic model

สรุป

- TextRank ใช้การวัดแบบ overlap of words ระหว่าง sentences เพื่อเปรียบเทียบ "Shared words"
- TextRank เหมาะสำหรับงาน Single-document summarization เพราะ Similarity measure เขามีความ simple กว่า
- TextRank เร็วกว่าเพราะเป็น Algorithm ที่เล็กกว่า LexRank

TextRank

```
!pip install summa
```

```
Collecting summa
  Downloading summa-1.2.0.tar.gz (54 kB)
    54.9/54.9 kB 4.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: scipy>=0.19 in /usr/local/lib/python3.10/dist-packages (from summa) (1.13.1)
Requirement already satisfied: numpy<2.3,>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from scipy>=0.19->summa) (1.26.4)
Building wheels for collected packages: summa
  Building wheel for summa (setup.py) ... done
  Created wheel for summa: filename=summa-1.2.0-py3-none-any.whl size=54389 sha256=ad9e44982490cf0705f96bd251da1f460acff8507260c7741
  Stored in directory: /root/.cache/pip/wheels/4a/ca/c5/4958614cfba88ed6ceb7cb5a849f9f89f9ac49971616bc919f
Successfully built summa
Installing collected packages: summa
Successfully installed summa-1.2.0
```

Join all sentences into one piece of text.

```
text = ' '.join(sentences)
print(text)
```

```
Long life promised for laptop PCs

Scientists are working on ways to ensure laptops can stay powered for an entire working day.

Building batteries from new chemical mixes could boost power significantly, say industry experts. The changes include everything fr

A survey carried out in 2000 by Forrester Research found that the shortness of battery life was the most complained about feature c

"For most of the 90s battery life was stuck on two to 2.5 hours." But now, he said, laptops can last much longer. It was not just i

Intel has been working with component makers to test energy consumption on all the parts inside a laptop and find ways to make them

This work has led to the creation of the Mobile PC Extended Battery Life (EBL) Working Group that shares information about building
```

เราสามารถเลือก ratio ที่เราต้องการได้จาก Document ทั้งหมดผ่าน Parameter ratio

```
from summa.summarizer import summarize
summarize(text, ratio=0.25)
```

```
'The changes include everything from the way chips for laptops are made, to tricks that reduce the power consumption of display
s.\n"
The industry has done a great job of wringing all possible energy storage out of that technology that they can." Some new batt
ery chemistries promise to cram more power into the same space, said Mr Trainor, though work still needed to be done to get them su
ccessfully from the lab to manufacturing.\n
Intel has been working with component makers to test energy consumption on all the parts
inside a laptop and find ways to make them less power hungry.\n
Some of the improvements in power use come simply because components
on chips are shrinking, said Mr Trainor.\n
On a larger scale, said Mr Trainor, improvements in the way that voltage regulators are m
```

สามารถเลือก Words ที่ต้องการสำหรับ Summarize ได้ผ่าน parameter words

```
summarize(text, words=50)
```

```
summarize(text, split=True, words= 100, language="english")
```

↩ ["The industry has done a great job of wringing all possible energy storage out of that technology that they can." Some new battery chemistries promise to cram more power into the same space, said Mr Trainor, though work still needed to be done to get them successfully from the lab to manufacturing.',  
'On a larger scale, said Mr Trainor, improvements in the way that voltage regulators are made can reduce the amount of power lost as heat and make a notebook more energy efficient.',  
'Also, said Mr Trainor, research is being done on ways to cut energy consumption on displays - currently the biggest power guzzler on a laptop.']

---

## ✓ สรุปจาก Lab นี้ LexRank and TextRank

### 1. การวัดความคล้ายของคำ Similarity Measure

- LexRank ใช้ Cosine Similarity เพื่อวัดความเหมือนระหว่างคำ / ประโยค
- TextRank ใช้ BM25 ในการวัดความเหมือนตัว

### 2. การประยุกต์ใช้งาน

- LexRank เหมาะสำหรับการทำ Multi-document summarization
- TextRank เหมาะสำหรับ Single-document summarization

### 3. ประสิทธิภาพ

- LexRank ทำงานได้ดีกว่าในการที่เราให้ความสำคัญกับ Context และ Semantic
  - TextRank เร็วกว่าและมีประสิทธิภาพในกรณีที่ต้องการสรุปแบบด่วน
- 

นายศวิษฐ์ โกสิยวัฒน์ 65070507238