




# An Application of Natural Language Processing on forex gold spot News Analysis

## การประยุกต์ใช้การประมวลผลภาษาธรรมชาติเพื่อการวิเคราะห์ข่าวตลาด forex gold spot

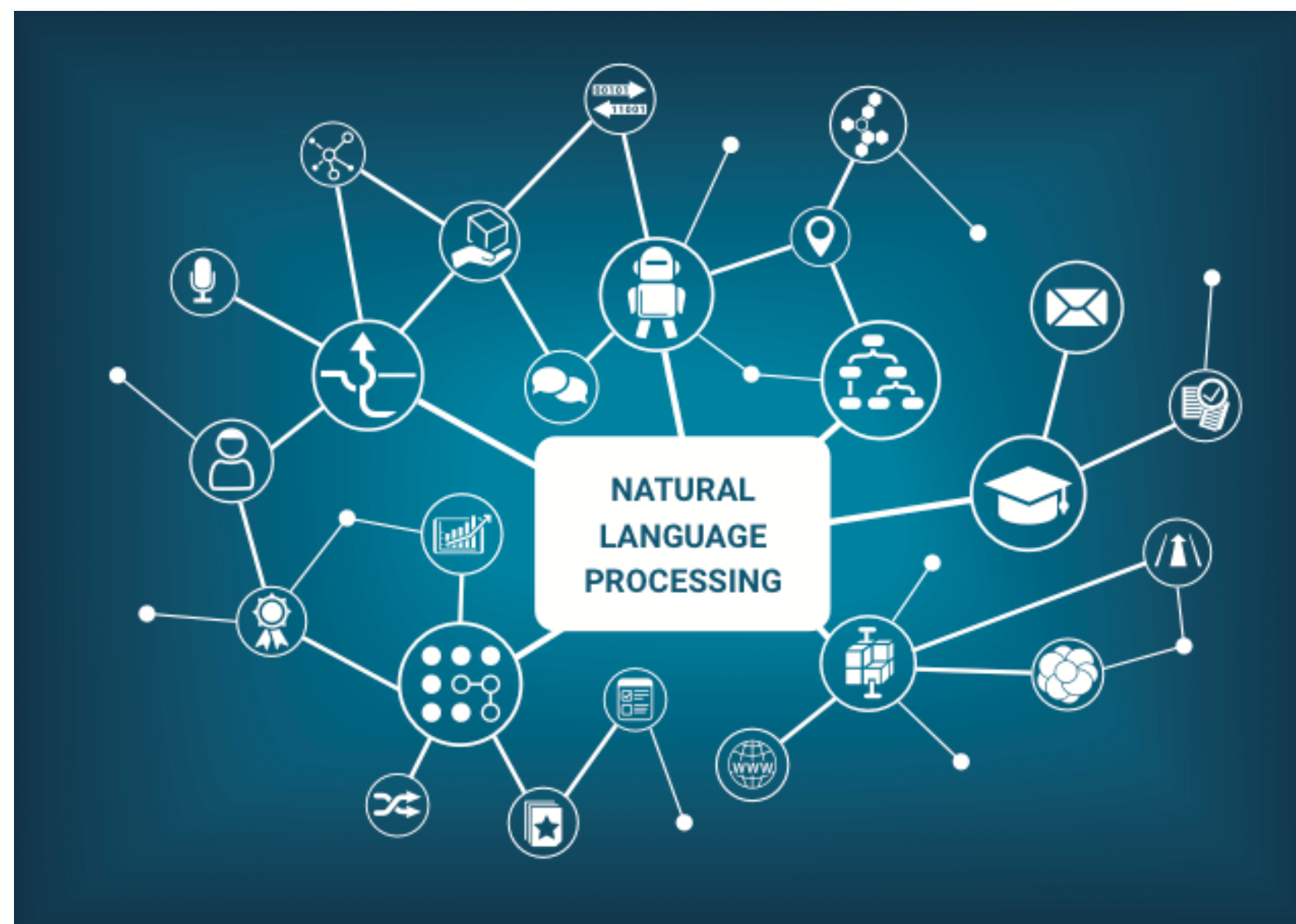
ผู้เสนอโครงการ  
นายชลชัย อภิชาติจิรุตติวรณ รหัสนักศึกษา 6210110646

อาจารย์ที่ปรึกษาโครงการ  
รศ.ดร.มนตรี กาญจนะเดชะ

# OUTLINE

- 
- ที่มาและความสำคัญ
  - แผนการดำเนินงาน
  - รายละเอียดการดำเนินงาน
  - ความก้าวหน้าการดำเนินงาน
  - สรุป

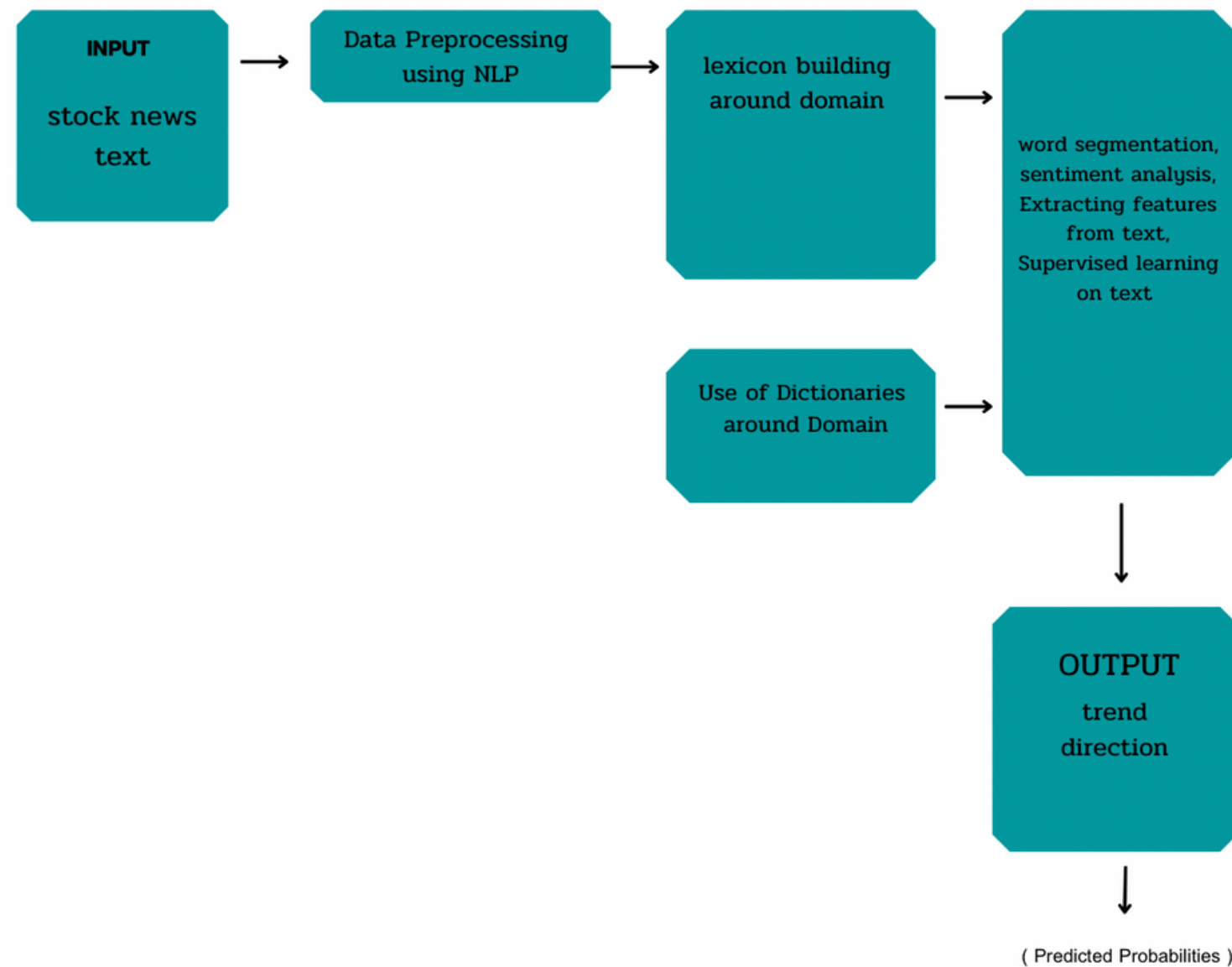
# ที่มาและความสำคัญ



# แผนการดำเนินงาน

<div>เวลา</div> <div>กิจกรรม</div>	PROJECT 1										PROJECT 2								
	มิ.ย. 65		ก.ค. 65		ส.ค. 65		ก.ย. 65		ต.ค. 65		ธ.ค. 65		ม.ค. 66		ก.พ. 66		มี.ค. 66		
(1) พัฒนาอัลกอริทึม																			
(2) พัฒนาเขียน โปรแกรมด้วย Python																			
(3) เทรนสมองกล																			
(4) ทดสอบระบบ																			
(5) ทดลองนำไปใช้จริง กับบัญชี Forex demo																			
(6) ปรับปรุงแก้ไขอุปกรณ์ให้สมบูรณ์																			
(7) จัดทำเว็บไซต์																			
(8) จัดทำรายงานให้สมบูรณ์																			

# รายละเอียดการดำเนินงาน



ขั้นตอน INPUT  
ข่าวสารtext

ขั้นตอน Process  
ใช้ algorithms

ขั้นตอน Output  
การคาดการณ์ล่วงหน้า  
สรุปข้อความ

# ขั้นตอน Input

ข่าวสาร	ข่าวประเภท	อ้างอิงค์
1. Census Bureau	กระทรวงพาณิชย์สหรัฐ	[7]
2. Us Department of <u>labo</u>	กระทรวงแรงงาน	[8]
3. Energy information Administration	<u>ข้อมูลด้านพลังงาน</u>	[9]
4. กองทุน SPDR	การซื้อขายทองคำของ กองทุน SPDR	[10]
5. Federal Reserve	ธนาคารกลางสหรัฐ	[11]
6. Bloomberg	รายงานข่าวทั่วไปรอบโลก	[12]
7. Twitter	ข่าวทั่วไป	[13]

รูปที่ 1 แหล่งข่าวสาร

# ขั้นตอน Process

- sentiment analysis
- Text Summarization
- Text extraction
- Text classification
- Machine translation

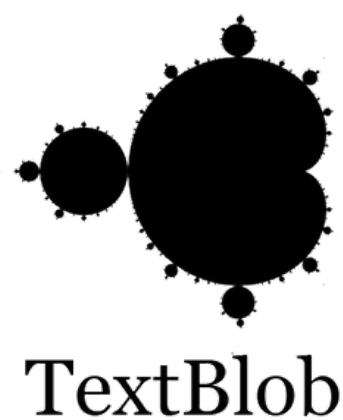
# ขั้นตอน Output

- สามารถคาดการณ์ล่วงหน้าได้
- สามารถแปลภาษาได้
- สามารถสรุปบทความได้



# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 1 ศึกษา Library Python NLP และทดสอบ



```
Welcome  nltnlp1.py x
nltnlp1.py > ...
1 from nltk.tokenize import sent_tokenize, word_tokenize
2
3 EXAMPLE_TEXT = "Hello Mr. Smith, how are you doing today? The weather is great, and Python is awesome. The sky is pinkish-blue. You
4
5 print(sent_tokenize(EXAMPLE_TEXT))
6 print(word_tokenize(EXAMPLE_TEXT))
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension
s/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 52987 -- /Users/whatbest/project/nltnlp1.py
['Hello Mr. Smith, how are you doing today?', 'The weather is great, and Python is awesome.', 'The sky is pinkish-blue.', "You shouldn't eat
cardboard."]
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'Python', 'is', 'awes
ome', '.', 'The', 'sky', 'is', 'pinkish-blue', '.', 'You', 'should', "n't", 'eat', 'cardboard', '.']
MacBook-Pro-khxng-Chualchai:project whatbest$
```

# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 1 ศึกษา Library Python NLP และทดสอบ

```
stopwords.py > ...
1  from nltk.corpus import stopwords
2  from nltk.tokenize import word_tokenize
3
4  example_sent = "This is a sample sentence, showing off the stop words filtration."
5
6  stop_words = set(stopwords.words('english'))
7
8  word_tokens = word_tokenize(example_sent)
9
10 filtered_sentence = [w for w in word_tokens if not w in stop_words]
11
12 filtered_sentence = []
13
14 for w in word_tokens:
15     if w not in stop_words:
16         filtered_sentence.append(w)
17 print("stop_words = ",stop_words)
18 print("word_tokens = ",word_tokens)
19 print("filtered_sentence = ",filtered_sentence)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53097 -- /Users/whatbest/project/stopwords.py
stop_words = {'hadn', 'itself', 'just', 'shan't', 'you', 'as', 'an', 'here', 'how', 'that'll', 'below', 'very', 'where', 'the', 'doesn't', 'theirs', 'won', 'wouldn't', 'is', 'any', 'nor', 'you've', 'he', 'then', 'yourselves', 'aren't', 'their', 'above', 'my', 'wasn't', 'while', 'you'll', 'too', 'over', 'some', 'isn', 'aren', 'your', 'been', 'were', 'do', 'have', 'no', 'hasn't', 'why', 'will', 'if', 'these', 'having', 'few', 'ma', 'or', 'herself', 'yourself', 'we', 'should've', 'nightn', 'she', 'ourselves', 'won't', 'are', 'did', 'so', 'couldn't', 'for', 'was', 'off', 'our', 'only', 'who', 'up', 'it', 'when', 'can', 'weren't', 'o', 'had', 'am', 'll', 'that', 'from', 'they', 'about', 'couldn', 'all', 'weren', 'me', 'now', 'further', 'and', 'haven', 'hers', 'not', 'should', 'most', 'i', 'm', 're', 'wasn', 'once', 'during', 'myself', 'on', 'mustn't', 'her', 'than', 'needn', 'shouldn't', 'in', 'again', 'between', 'haven't', 'his', 'into', 'then', 'you'd', 'but', 'themselves', 'don', 'does', 'at', 'own', 's', 'you're', 'because', 'himself', 'down', 'both', 'other', 'needn't', 'isn't', 'has', 'after', 'she's', 'of', 'each', 'it's', 'this', 'nightn't', 'to', 'its', 'through', 'doing', 'him', 'a', 'shan', 'what', 'whom', 'more', 'd', 'ain', 'mustn', 'same', 'which', 'didn't', 'out', 'hadn't', 'before', 'don't', 'against', 'under', 'shouldn', 'wouldn', 'ours', 'didn', 'until', 'there', 'yours', 'y', 'by', 'be', 'being', 'hasn', 't', 'such', 'doesn', 've', 'those', 'with'}
word_tokens = ['This', 'is', 'a', 'sample', 'sentence', ' ', ' ', 'showing', 'off', 'the', 'stop', 'words', 'filtration', '.']
filtered_sentence = ['This', 'sample', 'sentence', ' ', ' ', 'showing', 'stop', 'words', 'filtration', '.']
MacBook-Pro-khxng-Chualchai:project whatbest$
```

# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 2 ทดสอบ Library NLTK

```
stem.py > ...
1 from nltk.stem import PorterStemmer
2 from nltk.tokenize import sent_tokenize, word_tokenize
3
4 ps = PorterStemmer()
5
6 example_words = ["python", "pythoner", "pythoning", "pythoned", "pythonly, feeling"]
7
8 for w in example_words:
9     print(ps.stem(w))
10
11 new_text = "It is important to by very pythonly while you are pythoning with python. All pythoners have pythoned poorly at least onc
12 words = word_tokenize(new_text)
13
14 for w in words:
15     print(ps.stem(w))
16
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:project whatbest\$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 53446 -- /Users/whatbest/project/stem.py

python  
python  
python  
pythonly, feel  
it  
is  
import  
to  
by  
veri  
pythonli  
while  
you  
are  
python  
with  
python  
all  
python  
have  
python  
poorli  
at  
least  
onc  
.  
MacBook-Pro-khxng-Chualchai:project whatbest\$

Stemming words with NLTK

```
part of speech.py > ...
1 import nltk
2 from nltk.corpus import state_union
3
4 from nltk.tokenize import PunktSentenceTokenizer
5 train_text = state_union.raw("2005-GWBush.txt")
6 sample_text = state_union.raw("2006-GWBush.txt")
7
8 custom_sent_tokenizer = PunktSentenceTokenizer(train_text)
9 tokenized = custom_sent_tokenizer.tokenize(sample_text)
10 def process_content():
11     try:
12         for i in tokenized[:5]:
13             words = nltk.word_tokenize(i)
14             tagged = nltk.pos_tag(words)
15             print(tagged)
16
17     except Exception as e:
18         print(str(e))
19
20
21 process_content()
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

MacBook-Pro-khxng-Chualchai:project whatbest\$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extension/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57279 -- /Users/whatbest/project/part of speech.py

[('PRESIDENT', 'NNP'), ('GEORGE', 'NNP'), ('W.', 'NNP'), ('BUSH', 'NNP'), ('S', 'POS'), ('ADDRESS', 'NNP'), ('BEFORE', 'IN'), ('A', 'NNP'), ('JOINT', 'NNP'), ('SESSION', 'NNP'), ('OF', 'IN'), ('THE', 'NNP'), ('CONGRESS', 'NNP'), ('ON', 'NNP'), ('THE', 'NNP'), ('STATE', 'NNP'), ('O F', 'IN'), ('THE', 'NNP'), ('UNION', 'NNP'), ('January', 'NNP'), ('31', 'CD'), ('', ''), ('2006', 'CD'), ('THE', 'NNP'), ('PRESIDENT', 'NNP'), ('Mr.', 'NNP'), ('Speaker', 'NNP'), ('', ''), ('Vice', 'NNP'), ('President', 'NNP'), ('Cheney', 'NNP'), ('', ''), ('members', 'NNS'), ('of', 'IN'), ('Congress', 'NNP'), ('', ''), ('members', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('Supreme', 'NNP'), ('Court', 'NNP'), ('and', 'CC'), ('diplomatic', 'JJ'), ('corps', 'NN'), ('', ''), ('distinguished', 'JJ'), ('guests', 'NNS'), ('', ''), ('and', 'CC'), ('fellow', 'JJ'), ('citizens', 'NNS'), ('', ''), ('Today', 'VB'), ('our', 'PRPS'), ('nation', 'NN'), ('lost', 'VBD'), ('a', 'DT'), ('beloved', 'VBN'), ('', ''), ('graceful', 'JJ'), ('', ''), ('courageous', 'JJ'), ('woman', 'NN'), ('who', 'WP'), ('called', 'VBD'), ('America', 'NNP'), ('to', 'TO'), ('its', 'PRPS'), ('founding', 'NN'), ('ideals', 'NNS'), ('and', 'CC'), ('carried', 'VBD'), ('on', 'IN'), ('a', 'DT'), ('noble', 'JJ'), ('dream', 'NN'), ('', ''), ('Tonight', 'NN'), ('we', 'PRP'), ('are', 'VBP'), ('comforted', 'VBN'), ('by', 'IN'), ('the', 'DT'), ('hope', 'NN'), ('of', 'IN'), ('a', 'DT'), ('glad', 'JJ'), ('reunion', 'NN'), ('with', 'IN'), ('the', 'DT'), ('husband', 'NN'), ('who', 'WP'), ('was', 'VBD'), ('taken', 'VBN'), ('s o', 'RB'), ('long', 'RB'), ('ago', 'RB'), ('', ''), ('and', 'CC'), ('we', 'PRP'), ('are', 'VBP'), ('grateful', 'JJ'), ('for', 'IN'), ('the', 'DT'), ('good', 'JJ'), ('life', 'NN'), ('of', 'IN'), ('Coretta', 'NNP'), ('Scott', 'NNP'), ('King', 'NNP'), ('', ''), ('Applause', 'NNP'), ('', ''), ('', ''), ('President', 'NNP'), ('George', 'NNP'), ('W.', 'NNP'), ('Bush', 'NNP'), ('reacts', 'VBZ'), ('to', 'TO'), ('applause', 'VB'), ('during', 'IN'), ('his', 'PRPS'), ('State', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('Union', 'NNP'), ('Address', 'NNP'), ('at', 'IN'), ('the', 'DT'), ('Capitol', 'NNP'), ('', ''), ('Tuesday', 'NNP'), ('', ''), ('Jan', 'NNP'), ('', ''), ('MacBook-Pro-khxng-Chualchai:project whatbest\$

Part of Speech Tagging with NLTK

# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 2 ทดสอบ Library NLTK

```
Lemmatizing.py > ...
1  from nltk.stem import WordNetLemmatizer
2
3  lemmatizer = WordNetLemmatizer()
4
5  print(lemmatizer.lemmatize("cats"))
6  print(lemmatizer.lemmatize("cacti"))
7  print(lemmatizer.lemmatize("geese"))
8  print(lemmatizer.lemmatize("rocks"))
9  print(lemmatizer.lemmatize("python"))
10 print(lemmatizer.lemmatize("better", pos="a"))
11 print(lemmatizer.lemmatize("best", pos="a"))
12 print(lemmatizer.lemmatize("run"))
13 print(lemmatizer.lemmatize("run", 'v'))
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57403 -- /Users/whatbest/project/Lemmatizing.py
cat
cactus
goose
rock
python
good
best
run
run
MacBook-Pro-khxng-Chualchai:project whatbest$
```

Lemmatizing with NLTK



# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 3 ทดสอบ Library NLTK & TextBlob

```
Wordnet with NLTK.py > ...
1  from nltk.corpus import wordnet
2  syns = wordnet.synsets("program")
3  print(syns[0].name())
4  print(syns[0].lemmas()[0].name())
5  print(syns[0].definition())
6  print(syns[0].examples())
7  synonyms = []
8  antonyms = []
9
10 for syn in wordnet.synsets("good"):
11     for l in syn.lemmas():
12         synonyms.append(l.name())
13         if l.antonyms():
14             antonyms.append(l.antonyms()[0].name())
15
16 print(set(synonyms))
17 print(set(antonyms))
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:project whatbest$ cd /Users/whatbest/project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 57441 -- "/Users/whatbest/project/Wordnet with NLTK.py"
plan.n.01
plan
a series of steps to be carried out or goals to be accomplished
['they drew up a six-step plan', 'they discussed plans for a new bond issue']
{'near', 'effective', 'practiced', 'proficient', 'unspoiled', 'respectable', 'skilful', 'secure', 'unspoilt', 'estimable', 'beneficial', 'goodness', 'upright', 'right', 'in_effect', 'expert', 'honorable', 'dependable', 'ripe', 'serious', 'soundly', 'just', 'undecomposed', 'safe', 'full', 'dear', 'in_force', 'thoroughly', 'adept', 'sound', 'salutary', 'skillful', 'trade_good', 'commodity', 'good', 'well', 'honest'}
{'evil', 'bad', 'evilness', 'badness', 'ill'}
```

Wordnet with NLTK

# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 3 ทดสอบ Library NLTK & TextBlob

```
textbox > sentimentddd.py > ...
1  from textblob import TextBlob
2
3  # Preparing an input sentence
4  sentence = '''The platform provides universal access to the world's best education, partnering with top universities and organizations to help advance the world's intellectual and economic future.'''
5
6  analysisPol = TextBlob(sentence).polarity
7  analysisSub = TextBlob(sentence).subjectivity
8
9  print(analysisPol)
10 print(analysisSub)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:Project whatbest$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 61369 -- /Users/whatbest/Documents/GitHub/Project/textbox/sentimentddd.py
0.5
0.26666666666666666
MacBook-Pro-khxng-Chualchai:Project whatbest$
```

# ความก้าวหน้าการดำเนินงาน

## ความก้าวหน้า 3 ทดสอบ Library NLTK & TextBlob

🔍 main ▾Project / summarize / Auto-Summarize an article .py / <> Jump to ▾Go to file⋮

👤 Chualchai Apichatitiworn commitLatest commit 44eb6a6 28 minutes ago🕒 History

👤 0 contributors

51 lines (42 sloc) | 1.67 KBRawBlame📄📋🗑️

```
1 import bs4 as bs
2 import urllib.request
3 import re
4 import nltk
5
6 scraped_data = urllib.request.urlopen('https://www.eia.gov/petroleum/weekly/')
7 article = scraped_data.read()
8
9 parsed_article = bs.BeautifulSoup(article,'lxml')
10
11 paragraphs = parsed_article.find_all('p')
12
13 article_text = ""
14
15 for p in paragraphs:
16     article_text += p.text
17 # Removing Square Brackets and Extra Spaces
18 article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
19 article_text = re.sub(r'\s+', ' ', article_text)
20 # Removing special characters and digits
21 formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text )
22 formatted_article_text = re.sub(r'\s+', ' ', formatted_article_text)
23 sentence_list = nltk.sent_tokenize(article_text)
24 stopwords = nltk.corpus.stopwords.words('english')
25
26 word_frequencies = {}
27 for word in nltk.word_tokenize(formatted_article_text):
28     if word not in stopwords:
29         if word not in word_frequencies.keys():
30             word_frequencies[word] = 1
31         else:
```

```
32     word_frequencies[word] += 1
33 maximum_frequency = max(word_frequencies.values())
34
35 for word in word_frequencies.keys():
36     word_frequencies[word] = (word_frequencies[word]/maximum_frequency)
37 sentence_scores = {}
38 for sent in sentence_list:
39     for word in nltk.word_tokenize(sent.lower()):
40         if word in word_frequencies.keys():
41             if len(sent.split(' ')) < 30:
42                 if sent not in sentence_scores.keys():
43                     sentence_scores[sent] = word_frequencies[word]
44             else:
45                 sentence_scores[sent] += word_frequencies[word]
46 import heapq
47 summary_sentences = heapq.nlargest(7, sentence_scores, key=sentence_scores.get)
48
49 summary = ' '.join(summary_sentences)
50
51 print(summary)
```

## Text Summarization with NLTK in Python

PROBLEMSOUTPUTTERMINALDEBUG CONSOLE

```
MacBook-Pro-khxng-Chualchai:Project whatbest$ cd /Users/whatbest/Documents/GitHub/Project ; /usr/bin/env /usr/local/bin/python3 /Users/whatbest/.vscode/extensions/ms-python.python-2021.10.1365161279/pythonFiles/lib/python/debugpy/launcher 60443 -- "/Users/whatbest/Documents/GitHub/Project/summarize/Auto-Summarize an article .py"
In 2021, we estimate the average Eagle Ford formation well continued to produce more oil at 21,900 barrels in its first month (Figure 3). We developed these production decline curves for Eagle Ford formation wells from approximately 750 sub-county areas, called grids, which are approximately 14 square miles. This is based on the known number of wells already drilled in a grid, their past decline profile, and developing all future potential well sites. Virtually all of this production has occurred in 16 of the 30 counties and within a producing subset of that total area of approximately 7.2 million acres. Without a price capable of providing a return on investment, producers will not invest capital in drilling a well. A substantially larger amount of the area becomes more profitable as a result of higher prices in 2022. However, not all possible acreage will be developed because of future surface infrastructure considerations or leased acreage that is unavailable for development.
MacBook-Pro-khxng-Chualchai:Project whatbest$
```

# สรุป

## สรุปผลการดำเนินงาน

1. สามารถวิเคราะห์ประโยคว่าเป็นเชิงบวกหรือเชิงลบได้โดยใช้ sentiment analysis เบื้องต้น
2. สามารถสรุปบทความโดยนำลิงค์มาจากเว็บไซต์ต่างๆที่เราต้องการสรุปบทความโดยใช้ Text Summarization
3. ได้รู้หลักการเบื้องต้นการทำ sentiment analysis และ Text Summarization โดยการทดลองใช้ Library NLTK

## ปัญหาและอุปสรรค

1. การวิเคราะห์หรือการเปลี่ยนคำ มีคำที่ผิดพลาดอยู่บ้างเนื่องจาก NLTK พจนานุกรมศัพท์ที่ไม่ได้อัปเดต
2. การดึงข้อมูลจากเว็บข่าว เพื่อมาทำ Text Summarization บางเว็บไซต์ไม่อนุญาตให้ดึงข้อมูลมาได้

## งานที่จะดำเนินการต่อ

1. ปรับปรุงแก้ไขการทำ Sentiment analysis เพื่อให้ทำนายหรือคาดการณ์ล่วงหน้าของกราฟ Forex
2. เทรนสมองกล เรื่องคำศัพท์เฉพาะทาง Forex



# เอกสารอ้างอิง

[1] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms (2014), Cambridge University Press , เข้าถึงล่าสุด 15 มกราคม 2565

[2] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality (2013), Advances in Neural Information Processing Systems 26 เข้าถึงล่าสุด 15 มกราคม 2565

[3] J. Pennington, R. Socher, and C. D. Manning, GloVe: Global Vectors for Word Representation (2014), In EMNLP. เข้าถึงล่าสุด 16 มกราคม 2565

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information (2016), arXiv preprint เข้าถึงล่าสุด 17 มกราคม 2565

[5] NLP Implementations : URL :

<https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3> เข้าถึงล่าสุด 18มกราคม 2565

# เอกสารอ้างอิง

[6] The theory you need to know before you start an NLP : URL : <https://towardsdatascience.com/the-theory-you-need-to-know-before-you-start-an-nlp-project-1890f5bbb793> เข้าถึงล่าสุด 12 มีนาคม 2565

[7] Us Department of labor : URL : <https://www.dol.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[8] Energy information Administration : URL : <https://www.eia.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[9] กองทุน SPDR : URL : <https://traderider.com/forex/spdr-%E0%B8%81%E0%B8%AD%E0%B8%87%E0%B8%97%E0%B8%B8%E0%B8%99%E0%B8%97%E0%B8%AD%E0%B8%87%E0%B8%84%E0%B8%B3%E0%B9%81%E0%B8%97%E0%B9%88%E0%B8%87> เข้าถึงล่าสุด 12 มีนาคม 2565

[10] Federal Reserve : URL : <https://www.federalreserve.gov/> เข้าถึงล่าสุด 12 มีนาคม 2565

[11] Bloomberg : URL : <https://www.bloomberg.com/asia> เข้าถึงล่าสุด 12 มีนาคม 2565

[12] Twitter : URL : <https://twitter.com/> เข้าถึงล่าสุด 12 มีนาคม 2562



# คำถามและข้อเสนอแนะ

