國立高雄科技大學
National Kaohsiung University of Science and Technology

Introduction to NLP and Its Applications

Speaker: 蘇佳益

Advisor: 陳聰毅

國立高雄科技大學建工校區電子工程系

https://github.com/chiayisu/Artificial_Intelligence_Course

Agenda

- Natural Language Processing
- The Design and Implementation of XiaoIce, an Empathetic Social Chatbot
- Google Assistance
- Recipes for building an open-domain chatbot
- Applications of the GPT-3 Model
- Introduction to Dialog System
- Regular Expression
- Introduction to API
- Reference

Natural Language Processing

What is NLP?

Applications of Natural Language Processing

Evolution of Natural Language Processing

Evolution of Natural Language Processing

TF & TF-IDF	Co-Matrix	Word2Vec	GPT	BERT	GPT-2	GPT-3
		Mikolov et al., 2013.	Radford et al., 2018.	Devlin et al., 2018	Radford et al., 2019	Brown et al., 2020
		Google	OpenAI	Google AI	OpenAI	OpenAI

Contextual Representation (Transformer)

Why is NLP hard?

- Ambiguity of Natural Language
  - Polysemy
  - Formal and Informal Language
  - Writing Taiwanese in the Chinese Language
  - SMS Language
- Evolution of Language
- Difference in Structure
  - Following the grammar V.S. Not following the grammar

Why is NLP hard? - Example

https://cdn-images-1.medium.com/max/800/1*gmL2WA-hoXe8HokFZ9wXIA.gif

- Ambiguity of Natural Language
  - 88、98、白泡泡（台語：皮膚很白)、水噹噹
  - 你真的是我們的老鼠屎耶!
  - Steph Curry發火起來可以在一場得40分，非
- Difference in Structure
  - I am going to watch a movie. < - > I'm ganna watch a movie.
  - I want to watch a movie. < - > I wanna watch a movie.

Corpora Introduction

- Corpora
  - https://github.com/fighting41love/funNLP
  - https://github.com/InsaneLife/ChineseNLPCorpus
  - https://github.com/SophonPlus/ChineseNlpCorpus
  - https://github.com/brightmart/nlp_chinese_corpus
  - http://asbc.iis.sinica.edu.tw/ - 中研院
  - Google 自行搜尋
- Stop Words
  - https://github.com/goto456/stopwords

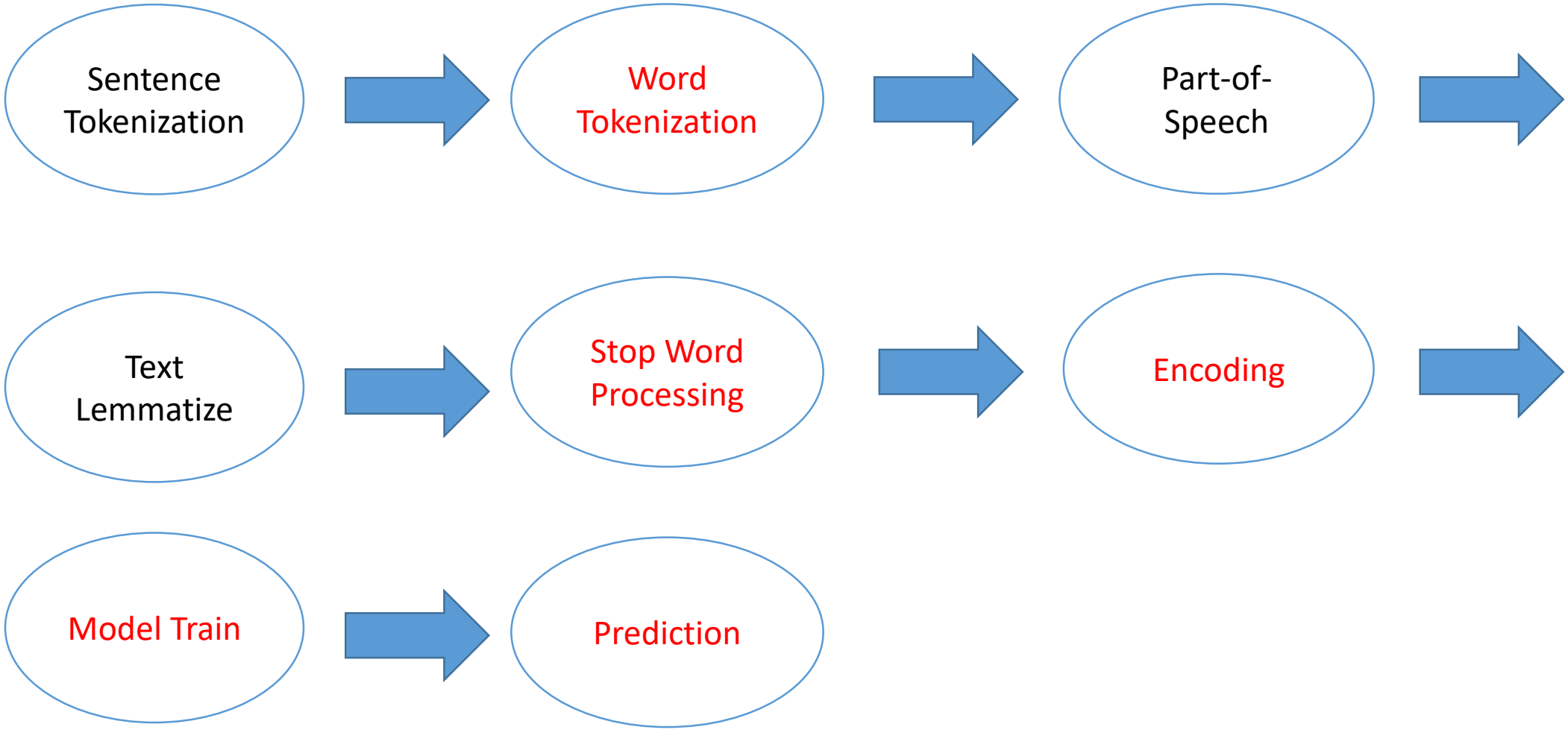
Introduction to JSON

- Similar to the dictionary in the programing language
- JSON object contains key and value.
- JSON Object
  - {“School”: “NKUST”, “Country” : ”Taiwan”}
- JSON Array
  - {“School”: [“NKUST”, “NTU”], “Country” : ”Taiwan”}
- JSON Introduction
  - https://www.json.org/json-en.html

中研院語料庫DEMO及重要參數說明

- Collocation: 統計出與關鍵詞共同出現的機率，可以設定前後幾個字但必須包含關鍵詞以及最高上限為10個詞。
- MI值: Mi值愈大表示與關鍵詞同時出現的機率越高
- FREQ(x): 關鍵詞在整個語料庫中出現的次數
- FREQ(y): 該詞在整個與料庫中出現的次數
- FREQ(x,y): 關鍵詞和該詞在限定的範圍內所出現的次數
- http://asbc.iis.sinica.edu.tw/
- 以上參數的重要嗎?

Flow of Natural Language Processing



Tokenization

- 簡單來說，斷詞是指將句子分成數個有意義的語詞，以便我們做後續的分析。
- 試著以電腦的角度來思考以下句子及問題
  - 郭台銘創立了鴻海公司。
  - 郭台銘 創立 了 鴻海 公司 。
  - 請問郭台銘公司的名字?
- 斷詞就像是人在閱讀文章，絕對不是一次看一整篇，而是由詞開始然後理解一整段文章，進而理解這篇文章的意義。

Tokenization Tool for the Chinese Language

- 中研院斷詞系統
- Jieba 斷詞系統

Part-of-Speech

- 將語詞的結果作詞性的標註
- 電腦在標註詞性絕對不是用人類的想法來標註。
- 我們可以經由詞性標註更清楚瞭解句子的意義，如同問你住哪裡，你絕對會去找名詞而不是動詞。 Sequence Label
- 較複雜的自然語言處理問題也會先經過Part-of-Speech來塞選不重要的詞性。

Text Lemmatize

- 主要用在英文
- 將每個意思相同但有變化的單字轉成最原始的單字。
- E.g.: Went -> Go

Stop Word Processing

- 去除比較無意義、詞頻過高、標點符號等等來提升正確率。
- 停用字會隨著問題不同而增加或減少。
- 試想以下句子的停用字
  - 欸~現在幾點呢?
  - 我現在住高雄耶!

Term Frequency

- 計算出語料庫中每個語詞所出現的次數
- 為何要計算詞頻而不是將所出現的做編碼如0、1、2、就好?這樣不是可以減少運算?程式也更好寫不是嗎?
- 最近較常使用的方法為將語詞轉換成密度向量。

TF-IDF

- Formula of TF-IDF

➢ tf-idf = $tf \ * idf(t)$, where tf is term frequency

➢ idf(t) = $\log\left(\frac{nd}{nd(t)}\right)$, where nd = number of documents, nd(t) = number of documents that contain the term t

The Design and Implementation of XiaoIce, an Empathetic Social Chatbot

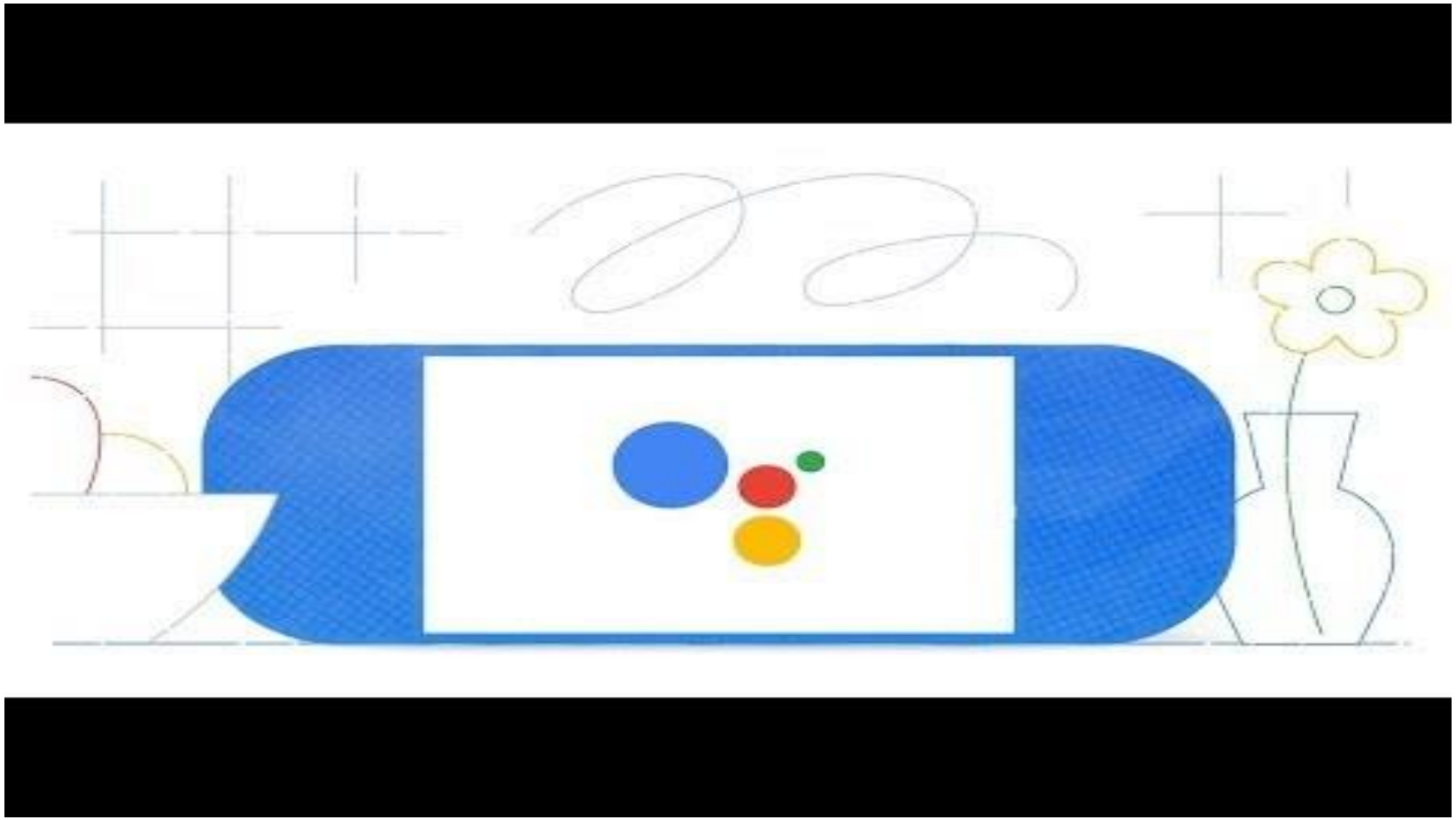
Zhou et al., Computational Linguistics, 2020.

Example: Singing

Example: Chat

Google Assistance

Demonstration



Google Duplex

Recipes for building an open-domain chatbot

Roller et al., Arxiv, 2020.

Demonstration

- https://ai.facebook.com/blog/state-of-the-art-open-source-chatbot/

Applications of the GPT-3 Model

- HTML Layout Generation
  - https://twitter.com/i/status/1282676454690451457
- Web Generation
  - https://twitter.com/jsngr/status/1287026808429383680?s=20
  - https://stripe.com/
- Generate & Update the Graph
  - https://twitter.com/plotlygraphs/status/1286688715167936512
- Others
  - https://github.com/elyase/awesome-gpt3

Introduction to Dialog System

Why Dialog System?

- 有許多人可能不會操作電腦或是對於操作電腦非常不熟悉。再好的圖形化介面都很難克服這問題，但幾乎人人都會講話，只是語言不同。
- Digital Data of July 2019

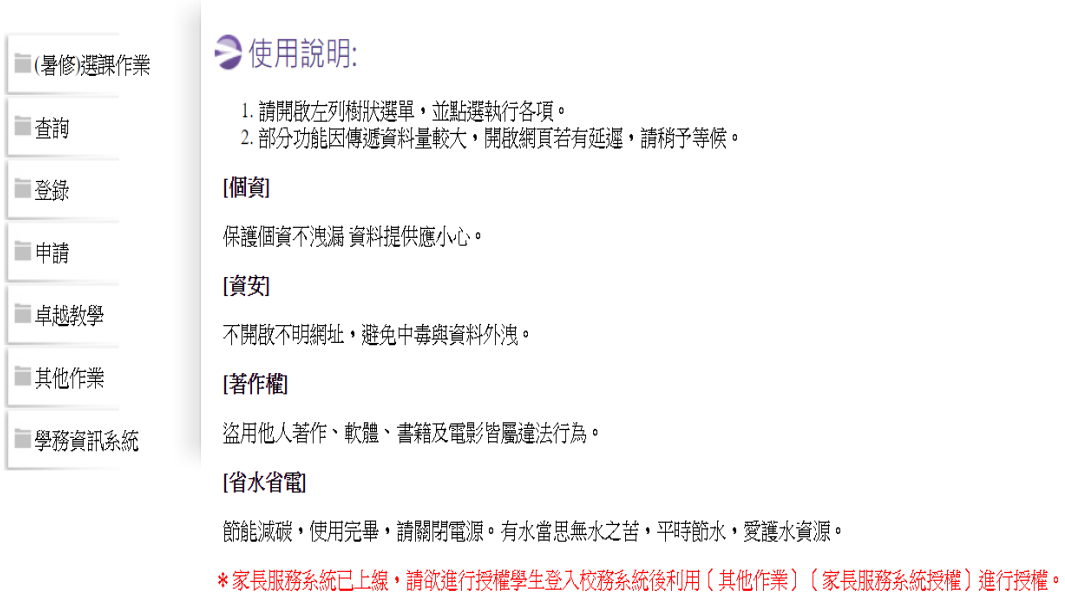
https://datareportal.com/global-digital-overview

Two Types of Intelligence Assistance

- Types of Intelligence Assistance
  - Reactive Assistance
  - Proactive Assistance

GUI v.s. CUI (Conversational UI)

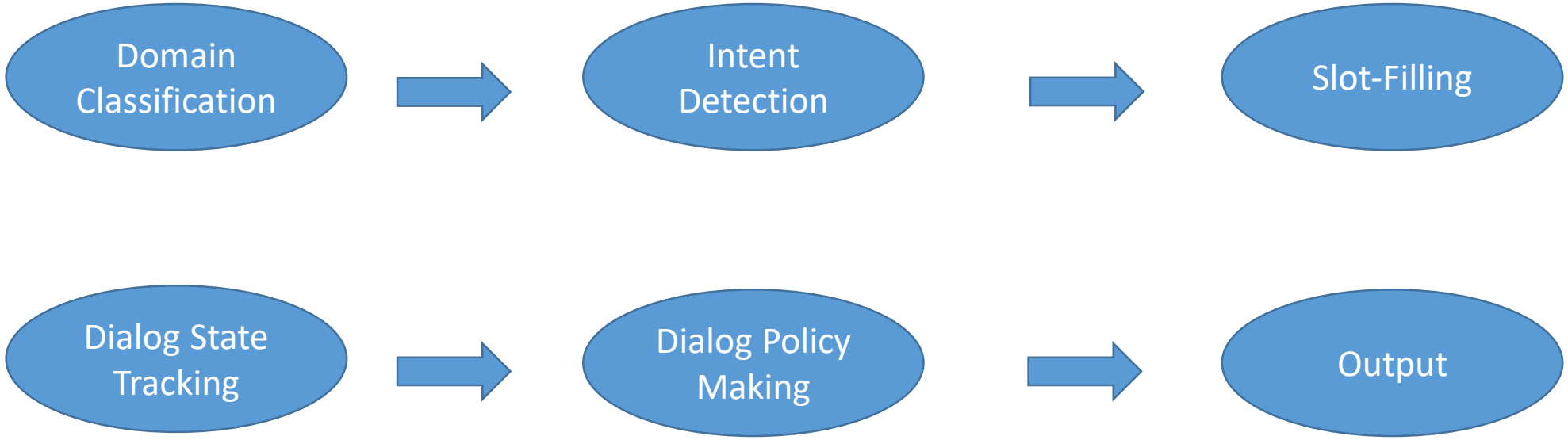
https://irs.thsrc.com.tw/IMINT/



http://webap.nkust.edu.tw/nkust/f_index.html

Introduction to Dialog System

- Types of Dialog System
  - Chit-Chat Dialog System
  - Task-Oriented Dialog System
- Flow of Dialog System



Dialog System Example

- I would like to book a HSR ticket from Kaohsiung to Nangang.
- Domain
  - HSR
- Intent
  - Book a Ticket
- Slot-Filling

o	o	o	o	o	o	o	o	B-Depart	I-Depart	B-Destination	I-Destination
I	would	like	to	book	a	HSR	ticket	from	Kaohsiung	to	Nangang

- Dialog State Tracking
  - Place
- Dialog Policy for Agent Action
  - We have place but we don't know the preference of seat, departure time etc. The system may return a question which is "what time do you depart?"

Regular Expression

What is regular expression?

- Write a rule to match the words.
- There are many editors allowing to search the text by using regular expression.

基本正規表示式介紹

- [ ] 匹配在括號裡面的字母
- [^] 匹配不包含在[ ]裡面的文字
- | 有OR的概念
- ? 前面的字元式非必要的
- * 前面的字元可以出現0次也可以出現多次
- + 前面的字元至少要出現一次
- . 不管任何字元都可以匹配
- ^ 匹配句子的起始位置
- $ 匹配句子結尾位置
- 還有許多規則可以到網路上查

Python Package: RE

- Built-in Function of Python
- Mainly used to process Regular Expression
- It can be used to match and split the string etc.

RE Example

```
Original String:NLP is interesting
Patttern: [ ]
regularized string: ['NLP', 'is', 'interesting']

Original String:NLP is interesting
Patttern: [^is]
regularized string: ['', '', '', '', 'is', 'i', '', '', '', '', 's', 'i', '', ''
]

Original String: NLP interesting aaa
Patttern: [^NLP|interesting]
regularized string: ['NLP', 'interesting', '', '', '', '']

Original String: NL interesting
Patttern: NLP?
regularized string: ['', ' interesting']

Original String: NLP interesting
Patttern: NLP*
regularized string: ['', ' interesting']

Original String: NLPPP interesting
Patttern: NLP+
regularized string: ['', ' interesting']

Original String: NLPis interesting
Patttern: NLP.
regularized string: ['', 's interesting']

Original String: NLPis interesting
Patttern: ^NLP.
regularized string: ['', 's interesting']

Original String: NLPis interesting
Patttern: interesting$
regularized string: ['NLPis ', '']
```

Introduction to API

Introduction to API

- Scikit-Learn
  - https://scikit-learn.org/stable/
- Pytorch
  - https://pytorch.org/
- TensorFlow
  - https://www.tensorflow.org/
- Google Colab Intro
  - https://colab.research.google.com/drive/14a6xiBuMtRF8snFYM-33i8BDNfiXBxQD
- TensorFlow With GPU
  - https://colab.research.google.com/drive/1aL_lmD8apPu2YXBxPzGSq0tdrV5btf-b
- Numpy
  - https://numpy.org/
- Pandas
  - https://pandas.pydata.org/

References

- D.Mannin et al., Foundations Of Statistical Natural Language Processing
- 自然語言處理流程
- An easy Introduction To Natural Language Processing
- 中研院語料庫
- 中研院斷詞
- 陳縕儂, 對話機器人的腦子與靈魂 Bot's Brain and Soul, PyConTW2017
- Dan Jurafsky and Chris Manning, “Natural Language Processing”, Stanford Online
- Python RE Package Document