2. NLP Questions

a) Stop words คืออะไร ให้ยกตัวอย่างในภาษาไทย และ ภาษาอังกฤษ

Ans Stop words คือ คำฟุ่มเฟือย ซึ่งเป็นคำที่ตัดออกได้โดยที่ข้อความยังสื่อความหมายเดิม

Ex.ภาษาอังกฤษ link (Library: NLTK, SpaCy, Gensim)

```python
import spacy
from nltk.tokenize import word_tokenize
# loading english language model of spaCy
en_model = spacy.load('en_core_web_sm')
# gettign the list of default stop words in spaCy english model
stopwords = en_model.Defaults.stop_words

sample_text = "Oh man, this is pretty cool. We will do more such things."
text_tokens = word_tokenize(sample_text)
tokens_without_sw= [word for word in text_tokens if not word in stopwords]

print(text_tokens)
print(tokens_without_sw)
```

spaCy.py hosted with ❤ by GitHub view raw

using spaCy to remove stop words

```
Output:
Tokenized text with stop words :
['Oh', 'man', ',', 'this', 'is', 'pretty', 'cool', '.', 'We', 'will', 'do', 'more', 'such', 'things', '.']
Tokenized text with out stop words :
['Oh', 'man', ',', 'pretty', 'cool', '.', 'We', 'things', '.']
```

tokenized vector with and without stop words

Ex. ภาษาไทย link (Library: PyThaiNLP)

```python
from pythainlp.tokenize import word_tokenize
text = "มีการความทดสอบภาษาไทย"
list_word = word_tokenize(text)
print(list_word)

from pythainlp.corpus import thai_stopwords
stopwords = list(thai_stopwords())
list_word_not_stopwords = [i for i in list_word if i not in stopwords]
print(list_word_not_stopwords)
```

ผลลัพธ์

```
['มี', 'การ', 'ความ', 'ทดสอบ', 'ภาษาไทย'] #คำทั้งหมดในประโยค
['ทดสอบ', 'ภาษาไทย'] #คำที่สื่อความหมายของประโยค
```

ปกติในการใช้งานทั้งภาษาไทยและภาษาอังกฤษ เราจะลบคำในประโยคตามตามคำที่มีใน Stop word ของแต่ละ Library ดังนั้นเราสามารถที่จะสร้างเป็น list ของ Stop word ของเรา เพื่อลบคำนั้นๆได้เช่นกัน

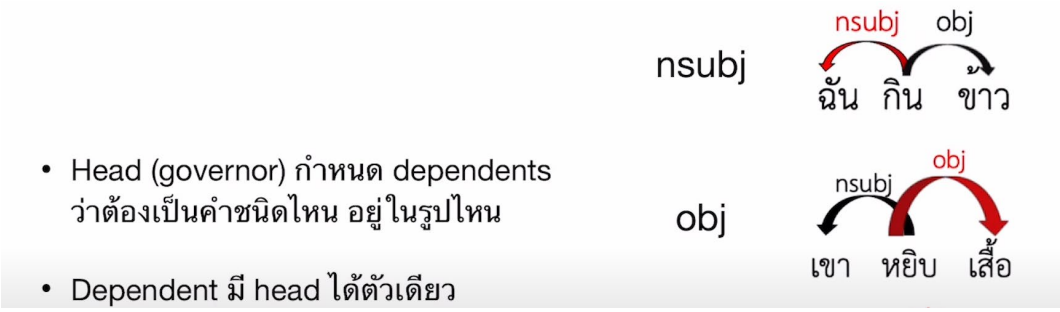
b) งาน Named Entity Recognition แนะนำให้ใช้จำนวนประโยคเท่าไหร่เป็นอย่างน้อย เพื่อให้สามารถได้ผลลัพธ์ที่น่าพอใจ

Ans ไม่แน่ใจ เพราะยิ่งใช้ข้อมูลเยอะมากๆ(10k++)ก็ดีกว่า จะยิ่งทำให้โมเดลเกิด loss น้อยลงได้ ดังนั้นผลลัพธ์ที่น่าพอใจก็ควรโมเดลที่ได้ loss น้อยที่สุด

c) Dependency parsing คืออะไร และมีประโยชน์อย่างไร

Ans Dependency parsing[link] คือการวิเคราะห์ความสัมพันธ์ของแต่ละคำในประโยคตามหลักภาษาไวยากรณ์ ซึ่งจะมีหลักการคือ ความสัมพันธ์แบบพึ่งพา (Dependency relation) จะกำกับระหว่าง Head โยงไปหา Dependent โดย Head จะเป็นตัวกำหนดว่า dependent ต้องเป็นคำชนิดไหน อยู่ในรูปไหน มีความสัมพันธ์กันอย่างไร และสำหรับแต่ละ Dependent นั้น ต้องมี Head ได้ตัวเดียวเท่านั้น

EX.



จากตัวอย่างข้างบน

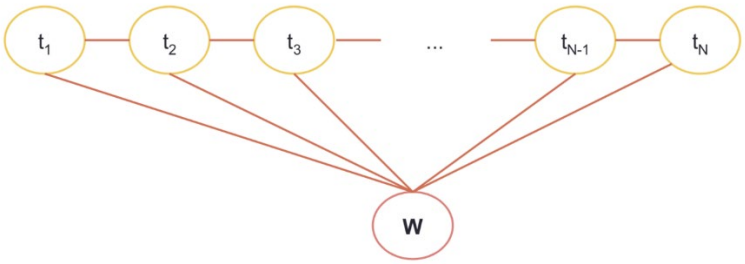
“ฉัน กิน ข้าว” -> “กิน” เป็น Head ดังนั้นคำว่า “กิน” จะกำหนดให้ Dependent อย่าง “ฉัน” ทำหน้าที่เป็น nsubj แล้ว Dependent อย่าง “ข้าว” ทำหน้าที่เป็น obj

Dependency Parser นำไปใช้ประโยชน์ได้ดังนี้

1. สามารถนำความสัมพันธ์เหล่านี้ ไปเขียนโค้ด และจัดทำข้อมูลการหาความสัมพันธ์ให้ AI เรียนรู้จดจำโครงสร้างประโยค ให้เข้าใจประโยคได้มากยิ่งขึ้น
2. ช่วยให้การทำ Text Generation หรือการสร้างคำให้ดูเป็นธรรมชาติมากขึ้น เพราะ AI เข้าใจโครงสร้างประโยค และไวยากรณ์
3. ใช้เป็นเครื่องมือตรวจสอบ Grammar ตรวจสอบไวยากรณ์ ของประโยคได้ จะทำให้รู้ว่าคำไหนในประโยคถูกหรือผิดหลักภาษา
4. Question Answering หรือ Chatbot นั่นเอง จะทำให้เรื่องในการถามตอบ ตรงประเด็นมากขึ้น รู้ความหมายของประโยคคำถาม และตอบได้ตรงกับสิ่งที่ถาม
5. Text Summarize หรือ การสรุปใจความสำคัญของประโยคหลาย ๆ ประโยคในบทความยาว ๆ ให้สามารถสั้นลงได้ เพราะการรู้ความสัมพันธ์ของประโยคสามารถ ละคำ โดยไม่สูญเสียความหมายได้

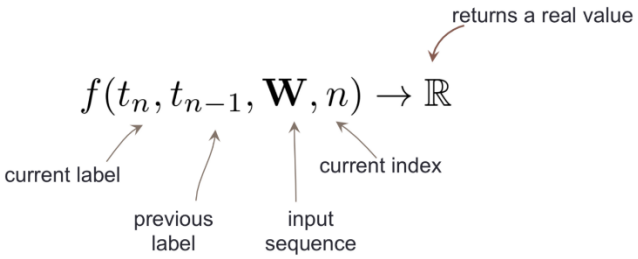
d) อธิบายโมเดลที่ใช้สำหรับงาน Named Entity Recognition และงาน Relation extraction

Ans จะใช้การทำงานของ Sequence labeling model ซึ่งเป็นโมเดลประเภทหนึ่งที่สามารถดึงส่วนข้อความที่สนใจออกจากข้อความหลัก โมเดลที่เป็นที่นิยมคือ Linear-chain CRF Model [link] ที่มีแนวคิดว่าการตัดสินใจทำนายประเภทคำ (tag) จะขึ้นอยู่กับความสัมพันธ์สองอย่างคือ



- Tag ของคำก่อนหน้า: จากภาพจะเห็นได้ว่า แต่ละ t หรือ tag นั้น มีเส้นเชื่อมต่อกันเป็นทอดๆ เช่น การตัดสินใจของ t2 จะขึ้นอยู่กับ t1 ด้วยว่าเป็นคำประเภทใด ซึ่งน่าจะตรงกับ การใช้งานจริง เพราะ ส่วนใหญ่คนไม่น่าพิมพ์คำประเภทเดิมติดกันหลายครั้ง เช่น ถ้าเรา predict เป็น Food ติดกันแล้วสองครั้ง ครั้งถัดไปก็ควรจะมีโอกาส predict ได้คำประเภทเดิมน้อยลง
- คำทั้งหมด : การทำนาย tag แต่ละครั้ง มีความสัมพันธ์กับทุกๆคำในประโยค (w) เช่น ถ้าประโยคขึ้นต้นด้วยคำว่าร้าน คำถัดไปก็มีโอกาสสูงที่จะเป็น business name

ประยุกต์ใช้เป็น Feature functions จากความสัมพันธ์แค่สองอย่าง เราสามารถนำมาคิด feature ย่อยๆ ได้เป็นจำนวนมากเท่าที่เราคิดว่ามีประโยชน์ต่อการทำนาย โดยแต่ละ feature จะถูกกำหนดเป็น feature function ที่รับ input เป็น tag ปัจจุบัน, tag ก่อนหน้า, คำทั้งหมด, และตำแหน่งปัจจุบัน ซึ่งเป็นไปตามกราฟความสัมพันธ์ ก่อนที่จะคำนวณออกมาเป็นคะแนน ซึ่งส่วนใหญ่มักเป็น ค่า 0, 1



Ex ร้าน |กาแฟ |อารีย์

โมเดลจะลอง tag ในแต่ละรูปแบบ เพื่อหารูปแบบที่ได้คะแนนรวมจากทุก feature functions สูงสุด

- Food | Business | Location = 1
- Stopword | Business | Location = 2

  …
- Stopword | Food | Location = 3

ซึ่งถ้าโมเดลนั้นทำนายได้ถูกต้อง **รูปแบบสุดท้าย จะต้องได้คะแนนเยอะที่สุด**

e) การใช้ Loss function และการปรับแต่ง Loss function อย่างไร เพื่อแก้ไขปัญหา imbalance dataset

Ans ในการแก้ปัญหา imbalance dataset ยกตัวอย่างเป็น classification แบบ 3 คลาสที่

Class 1: 900 elements Class 2: 15000 elements Class 3: 800 elements

ปกติเราก็ใช้ loss function อย่าง cross-entropy ที่ Yik = 1 เมื่อ k เป็น class ที่ถูกต้องของข้อมูลที่ i แล้วกรณีอื่นให้ Yik = 0 เขียนเป็นสมการดังนี้

$$H_y(y') = -\sum_i \sum_{k=1}^{K} y_{ik} log(y'_{ik})$$

ซึ่งเราจะปรับแต่ง loss function โดยเพิ่มเทอมของ W_k

$$H_y(y') = -\sum_i \sum_{k=1}^{K} w_k y_{ik} log(y'_{ik})$$

ตัวอย่างเช่น หาก class 1 มี 900 class 2 มี 15000 และclass 3 มี 800 ตัวอย่าง Weight ของแต่ละclass จะเท่ากับ 0.889, 0.053 และ 1.0 ตามลำดับ จะทำให้ Loss ของ Class นั้น ๆ มีสัดส่วนน้อยลง ส่งผลให้สัดส่วน Gradient ลดลงมาพอ ๆ กัน ทำให้โมเดลไปสนใจเรียนรู้ เทรนโมเดล ที่ให้ผลลัพธ์กับทุก Class เท่าเทียมกันมากขึ้น

Ref: link , link2

f) เจอปัญหาอะไรบ้างในการทำ NLP ภาษาไทย และมีแนวทางแก้ปัญหาอย่างไร (ถ้าไม่เคยมีประสบการณ์ ไม่ต้องตอบข้อนี้)

Ans “ ไม่เคยมีประสบการณ์ ”