การค้นหาสินทรัพย์ด้วยระบบแนะนำ

RECOMMENDATION Asset Search

กิตติทัต กาญจนพิบูลย์

KITTITUT GANCHANAPIBOON

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

RECOMMENDATION Asset search

KITTITUT GANCHANAPIBOON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INFIRMATION ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

KING MONGKUT ‘S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2020

หัวข้อปริญญานิพนธ์	การค้นหาสินทรัพย์ด้วยระบบแนะนำ
นักศึกษา	นายกิตติทัต กาญจนพิบูลย์ รหัสนักศึกษา 60010065
ระดับปริญญา	วิศวกรรมศาสตร์บัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2563
อาจารย์ที่ปรึกษา	อ.นิจจารีย์ สัตยารักษ์

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(อ.นิจจารีย์ สัตยารักษ์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	การค้นหาสินทรัพย์ด้วยระบบแนะนำ
นักศึกษา	นายกิตติทัต กาญจนพิบูลย์ รหัสนักศึกษา 60010065
ระดับปริญญา	วิศวกรรมศาสตร์บัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2563
อาจารย์ที่ปรึกษาปริญญานิพนธ์	อ. นิจจารีย์ สัตยารักษ์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอระบบการค้นหาแบบใหม่ขึ้น เพื่อแก้ไขปัญหาการค้นหาทรัพย์สินประเภทอสังหาริมทรัพย์ ที่เป็นการค้นหาในรูปแบบการค้นหาผ่านเลขประจำงานการประเมินทรัพย์สิน เป็นการค้นหาสินทรัพย์ด้วยการค้นหาด้วยชื่อสินทรัพย์ เพื่อเพิ่มความยืดหยุ่นในการค้นหา

โดย RECOMMENDATION Asset search จัดทำขึ้นด้วยภาษา Python และ ใช้ Angular ในส่วนของการแสดงผลบนเว็ปไซต์ ซึ่งในการพัฒนาได้มีการนำทักษะทางด้าน NLP มาประยุกต์ใช้ เพื่อเพิ่มประสิทธิภาพการค้นหา ซึ่งเมื่อผู้ใช้งานทำการป้อนข้อมูล เรียบร้อยแล้วระบบจึงนำข้อมูลที่มีมาประมวลผล เพื่อหาคำตอบที่เป็นไปได้ และนำข้อมูลเหล่านั้นมาแสดงผลให้แก่ผู้ใช้งาน ซึ่งผลลัพธ์ที่ได้ คือทำให้การค้นหานั้นสามารถดำเนินการได้อย่างรวดเร็วมากยิ่งขึ้น

Project Title	RECOMMENDATION Asset search
Student	Mr. KITTITUT GANCHANAPIBOON Student ID 60010065
Degree	Bachelor of Engineering
Program	Information Engineering
Academic Year	2020
Project Advisor	Miss Nitjaree Satayarak

ABSTRACT

RECOMMENDATION Asset search presents a new solution for asset searching in Commercial Banks. Since the legacy search system is searching by ID number, this project will replace legacy search system with asset name searching which is more flexible for discovery.

RECOMMENDATION Asset search is built on python programming language and develop front-end with Anugular. While developing this project, there need to be NLP subject to apply with searching for leverage searching power. To run this program, user must input asset name in searching field. So program will process input data with algorithm to find a matching or most possibly result in database then it can display the answer on the screen. This achievement makes searching progress less time consuming.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี ด้วยความอนุเคราะห์จากผู้เชี่ยวชาญในการสร้างโปรแกรมประยุกต์บนเว็บไซต์ที่ให้ความช่วยเหลือ และให้คำแนะนำ ผู้รายงานใคร่ขอกราบ ขอบพระคุณอาจารย์ที่ปรึกษาในการตรวจสอบความถูกต้องของเอกสารมา ณ โอกาสนี้

ขอขอบคุณ เพื่อนๆ และ พี่ๆ ทุกท่าน ที่คอยเป็นผู้ดูแล ช่วยเหลือ และคอยมอบคำแนะนำต่างๆ เพื่อให้สามารถรับมือปัญหาเหล่านั้นได้อย่างถูกต้อง

นาย กิตติทัต กาญจนพิบูลย์

สารบัญ

สารบัญรูป

บทที่1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันนี้ ทีมนักประเมินราคาสินทรัพย์ในองค์กร ธนาคารพาณิชย์ใช้เวลาในการค้นหาข้อมูลเพื่อที่จะใช้ในการประเมินราคาสินทรัพย์เป็นเวลา 10 – 15 นาที/งาน ซึ่งการทำ search engine ตรงนี้จะช่วยลดเวลาการทำงาน และยังใช้ข้อมูลนอกเหนือจากฐานข้อมูลเดิมที่มีมาประยุกต์ใช้งาน เพื่อให้ขอบเขตการค้นหามากยิ่งขึ้น

การประเมินราคาสินทรัพย์ที่ทีมนักประเมินราคาสินทรัพย์ทำอยู่นั้นมีอยู่ด้วยกัน 2 รูปแบบคือการประเมินโดยตรง กับการประเมินราคารโดยอ้างอิงจากราคาตลาด ซึ่งการประเมินราคาโดยตรงนั้นคือการประเมินราคาจากปัจจัยต่างๆที่เกี่ยวข้องซึ่งอาจส่งผลต่อราคาสินทรัพย์ได้ แต่การประเมินราคาโดยอ้างอิงจากราคาตลาดนั้นสามารถทำได้ด้วยการหาสินทรัพย์เปรียบเทียบในบริเวณใกล้เคียงที่มีลักษณะของสินทรัพย์ใกล้เคียงกัน โดยในการค้นหาสินทรัพย์เปรียบเทียบนั้นจำเป็นที่จะต้องทราบเลขรหัสงานเพื่อที่จะค้นหาสินทรัพย์ ดังนั้นนักประเมินสินทรัพย์ต้องย้อนกลับไปหาข้อมูลในฐานข้อมูลเดิมก่อนเพื่อที่จะหาเลขรหัสงานที่เกี่ยวข้องกับการประเมินราคาสินทรัพย์

ผู้จัดทำจึงได้คิดที่จะทำการสร้าง search engine ที่สามารถค้นหาสินทรัพย์เปรียบเทียบได้จากการระบุชื่อโครงการที่เกี่ยวข้อง ซึ่งคาดว่าจะสามารถลดหย่อนระยะเวลาการทำงานได้

RECOMMENDATION Asset search นั้นจึงถูกสร้างขึ้นมาเพื่อที่จะตอบโจทย์การลดเวลาการทำงานของทีมนักประเมินราคาสินทรัพย์ โดยการค้นหานั้นจะถูกออกแบบมาเพื่อให้การทำงานเป็นไปได้ด้วยความรวดเร็วและราบรื่นมากที่สุด

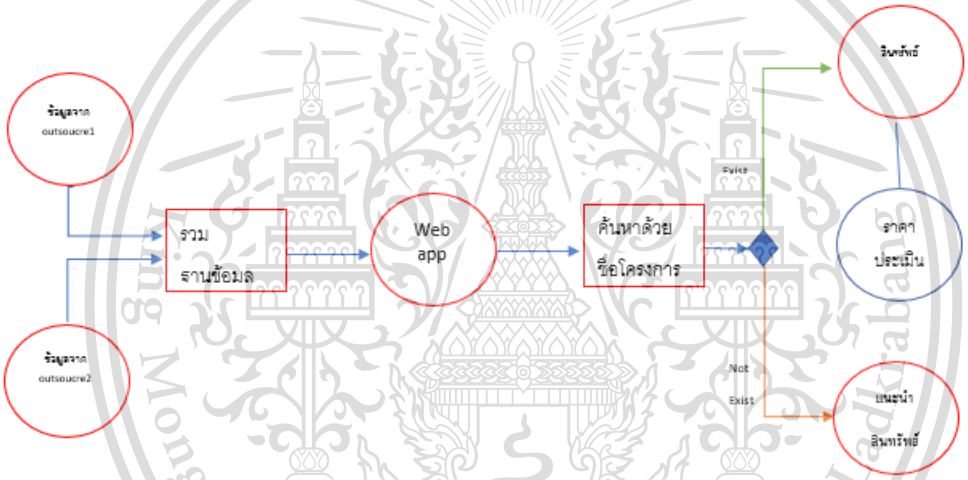
1.2 วัตถุประสงค์

1.2.1 เพื่อลดเวลาการทำงานของทีมนักประเมินราคาสินทรัพย์

1.2.2 เพื่อนำเอาข้อมูลที่อยู่นอกฐานข้อมูลเดิมมาผนวกรวมเข้ากับฐานข้อมูล เพื่อที่จะขยายขอบเขตการค้นหาให้กว้างยิ่งขึ้น

1.3 ขอบเขตการศึกษา

อสังหาริมทรัพย์ในพื้นที่รอบกรุงเทพมหานคร โดยสินทรัพย์เหล่านั้น จะเป็นข้อมูลที่เกิดจากการเก็บข้อมูลของธนาคารพาณิชย์โดยการทำงานมีลักษณะดังรูปข้างล่างนี้



รูปที่1.1 ออกแบบแผนผังการทำงานของโครงงานนี้

1.4 ผลที่คาดว่าจะได้รับ

หากโครงงานสามรถสำเร็จลุล่วงไปได้ด้วยดี คาดว่าจะส่งผลทำให้การค้นหาข้อมูลสามารถทำงานได้อย่างยืดหยุ่นมากยิ่งขึ้น และสามารถค้นหาอสังหาริมทรัพย์(บ้านจัดสรรค์) ได้ด้วยชื่อของโครงการ

1.5 อุปกรณ์ที่ต้องใช้

1.5.1 ฮาร์ดแวร์

-เครื่องคอมพิวเตอร์สำหรับพัฒนาโปรแกรม ที่มีการเชื่อมต่อกับเน็ตเวิร์ค จำนวน 1 เครื่อง

1.5.2 ซอฟท์แวร์

-โปรแกรมสำหรับการเขียนโค้ด (VS code)

-Ubuntu 18.04 LTS

-python 3.6.9

-MongoDB

1.6 ช่วงเวลาการดำเนินงาน

Task Name	2020							2021		
	June	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
1.ค้นหาหัวข้อ										
2.พัฒนาโปรแกรม										
3.Implement										
4.Test and Debug										
5.Documentation										

รูปที่1.2 ตารางเวลาช่วงดำเนินงานของโครงงานนี้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 Artificial Intelligent

Artificial Intelligent (AI) หรือ ปัญญาประดิษฐ์ คือหนึ่งในวิชาของงานในอาชีพวิทยาการข้อมูล (Data Science) เป็นศาสตร์ที่ผสมผสานวิชาสถิติ (Statistics) และ การเขียนโปรแกรม (Computer Programming) เข้าด้วยกัน เพื่อที่จะสร้างโปรแกรมที่มีความฉลาดมากพอที่จะสามารถทำการตัดสินใจในการทำสิ่งต่างๆ โดยการตัดสินใจนั้นจะตั้งอยู่บนพื้นฐานของข้อมูล

Machine Learning

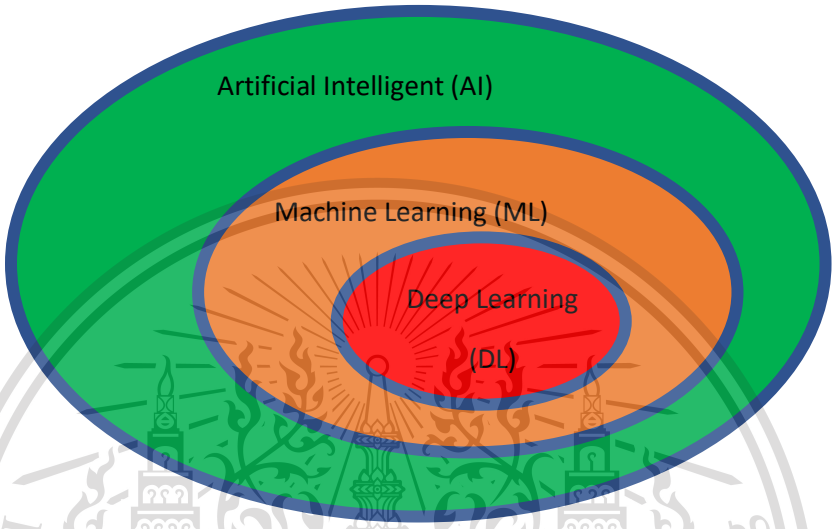
Machine Learning (ML) เป็นหนึ่งในสาขาย่อยของ AI โดย Machine Learning จะถูกออกแบบออกมาให้เป็นโปรแกรมที่มีความฉลาดมากพอที่จะสามารถทำนาย (prediction) ผลลัพธ์ของข้อมูลได้ โดยความฉลาดของโปรแกรมนี้จะเกิดมาจากการที่โปรแกรมนั้นเรียนรู้ข้อมูลต่างๆผ่าน ชุดข้อมูลที่ผู้พัฒนาได้ป้อนเข้าไป

โดย Machine Learning จะแบ่งการเรียนรู้ออกเป็น 3 รูปแบบด้วยกันได้แก่

1.) Supervised Learning: การเรียนรู้ของจักรกลผ่านการสอนด้วยข้อมูลและผลลัพธ์ของข้อมูล
2.) Unsupervised Learning: การเรียนรู้ของจักรกลโดยเรียนรู้จากการป้อนข้อมูลแค่ input
3.) Reinforcement Learning: การเรียนรู้ของจักรกลด้วยตัวเอง

Deep Learning

Deep Learning (DL) เป็นหนึ่งในสาขาย่อยของ ML โดย Deep Learning นั้นจะจำลองความฉลาดเหมือนโครงค่ายสมองของมนุษย์ (Neural Network)



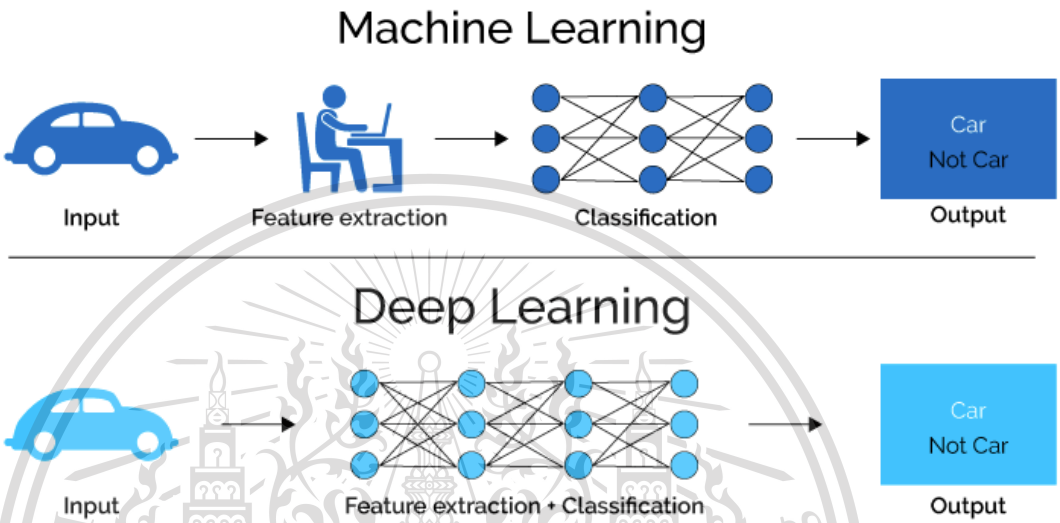
รูปที่2.1 ขอบเขตการศึกษาและความหมายของ AI, ML และ DL

2.1.1 Neural Network

Neural Network เป็นโมเดลทางคณิตศาสตร์ที่เป็นส่วนหนึ่งของ Deep learning โดยแนวคิดในการสร้าง Neural Network นั้นมีต้นแบบแนวคิดมาจากแบบจำลองโครงข่ายประสาทของมนุษย์ จึงทำให้ Neural Network เป็นโมเดลที่ใกล้เคียงกับคำว่า Artificial Intelligent มากที่สุด โดย Neural Network สามารถนำไปประยุกต์ใช้ได้กับหลายอย่างด้วยกัน โดย Artificial Intelligent ที่มีชื่อเสียง เช่น รถยนต์ขับเคลื่อนอัตโนมัติ (Self-Driving car), AlphaGo ต่างก็พัฒนามาจาก Neural Network

รูปที่2.2 AlphaGo ตัวอย่างของ Neural Network ที่มีชื่อเสียง

Neural Network เป็นโมเดลที่ได้รับความนิยมเป็นอย่างมากในปัจจุบัน เนื่องจากเป็นโมเดลที่ไม่จำเป็นที่จะต้องทำการคัดกรองลักษณะเด่นของข้อมูล (Feature Extraction) ซึ่งจะแตกต่างจาก Machine Leaning จึงทำให้ Neural Network เป็น model ที่สามารถเรียนรู้ลักษณะเด่นของข้อมูลผ่าน input data ได้โดยตรง



รูปที่2.3 ความแตกต่างระหว่าง Machine Learning และ Deep Learning

องค์ประกอบของ Neural Network

Neural Network คือโมเดลทางสถิติ ที่มีโครงสร้างลอกเลียนแบบมาจากโครงข่ายประสาทของมนุษย์ โดยองค์ประกอบพื้นฐานของ Neural Networks นั้นคือ

-Neurons (Nodes) เป็นหน่วยที่เล็กที่สุดใน Neural Network โดยจะเป็นหน่วยของข้อมูล ซึ่งจะมีสถานะแตกต่างกันไป เมื่อผ่านการกระทำทางคณิตศาสตร์ (Math Operation) โดยอาจจะเป็นข้อมูล input ที่เป็น matrix ตัวเลข, ข้อมูล output ที่สามารถทำนายผลลัพธ์ออกมา เป็นต้น โดย Nodes ในแถวเดียวกันจะถูกเรียกว่า Layer

-Links (Weights) เป็นองค์ประกอบในการเชื่อมต่อ Neurons ในแต่ละ layers เข้าด้วยกัน โดยการเชื่อมต่อรหว่าง node นั้นอาจจะประกอบไปด้วย Math Operation บางอย่าง เพื่อที่จะหาค่าข้อมูลใน Nodes ที่เป็นผลลัพธ์

โดยปรกติแล้ว Neural Network จะประกอบไปด้วย 3 Layers เป็นขั้นต่ำ นั้นคือ

1.Input Layer: เป็นชั้นของการรับข้อมูลจาก input เข้ามาเพื่อที่จะทำให้กลายเป็น Node ต่างๆ โดยทั่วไปแล้ว Layer นี้จะมีเพียงแค่ชั้นเดียว

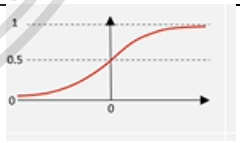
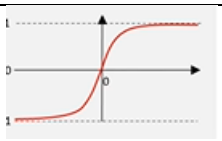
2.Hidden Layer: เป็นชั้นของการแปลงข้อมูลผ่าน Math Operation ที่แตกต่างกันไป เพื่อที่จะให้ได้ Node ผลลัพธ์ ซึ่งใน layer นี้ สามารถประกอบไปด้วย hidden layer หลายชั้นติดต่อกันได้

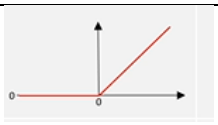
3.Output Layer: เป็นชั้นสุดท้าย ซึ่งมีหน้าที่ในการทำนาย (predict) ผลลัพธ์ว่าควรเป็นสิ่งใด

2.1.2 Activation Function

Activation Function คือสมการทางคณิตศาสตร์ที่จะมอบผลลัพธ์เพื่อทำการตัดสินใจว่าควรใช้งาน (Activate) หรือไหม โดย Activation Function สามารถมอบผลลัพธ์ออกมาเป็นค่าต่างๆ ตามแต่สมการที่กำหนด

Activation Function เข้ามามีส่วนร่วมสำคัญใน Neural Network โดยจะทำหน้าที่คัดกรองว่าผลลัพธ์ที่ได้จากสมการควรที่จะ activate หรือไหม โดยสมการที่ได้รับความนิยมในการใช้งานทางด้าน Neural Network จะมีดังนี้

ชื่อ function	สมการ	อนุพันธ์ของสมการ	ผลลัพธ์	ค่าผลลัพธ์
Sigmoid (logistic function)	$S(x) = \frac{1}{1+e^{-x}}$	$\frac{\partial S(x)}{\partial x} = S(x)(1 - S(x))$		(0,1)
TanH (Hyperbolic tangent)	$\tanh(x) = \frac{e^{x}+e^{-x}}{e^{x}-e^{-x}}$	$\frac{\partial tanh(x)}{\partial x} = \frac{4}{(e^{x}+e^{-a})^{2}}$		(-1,1)

ReLu (rectified linear unit)	$relu(x) = \max(0, x)$	$\frac{\partial relu(x)}{\partial x} = \begin{cases} 0, if\ a \leq 0 \\ 1, if\ a > 0 \end{cases}$		(0,∞)
softmax	$S(x) = \frac{1}{1 + e^{-x}}$	$\frac{\partial S(x)}{\partial x} = S(x)\left(\delta_{ij} - S(x)\right)$ Where δ_{ij} is 1 if i=j, 0 otherwise		(0,1)

รูปที่2.4 ตารางแสดงผลความแตกต่างของ Activation Function แต่ละแบบ

2.1.2 Cost Function/ Loss Function

Cost Function / Loss Function เป็นหนึ่งในวิธีการทางคณิตศาสตร์ที่ต้องการที่จะหาค่าความผิดพลาดที่เกิดขึ้น เพื่อที่จะได้นำค่าเหล่านั้นไปปรับปรุงค่าที่เกี่ยวข้อง เพื่อที่จะให้โครงข่ายมีความแม่นยำมากยิ่งขึ้น

Loss function ที่ได้รับความนิยม

1.L1-Norm,L2-Norm ซึ่งเหมาะสำหรับกับ model ที่มีจุดมุ่งหมายสำหรับการทำ regression

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} W_j^2$$

Loss function — Regularization Term

รูปที่2.5 สมการของ L1 Regularization และ L2 Regularization

2.Cross Entropy loss: ซึ่งเหมาะสำหรับ model ที่มีจุดมุ่งหมายสำหรับการทำ classification

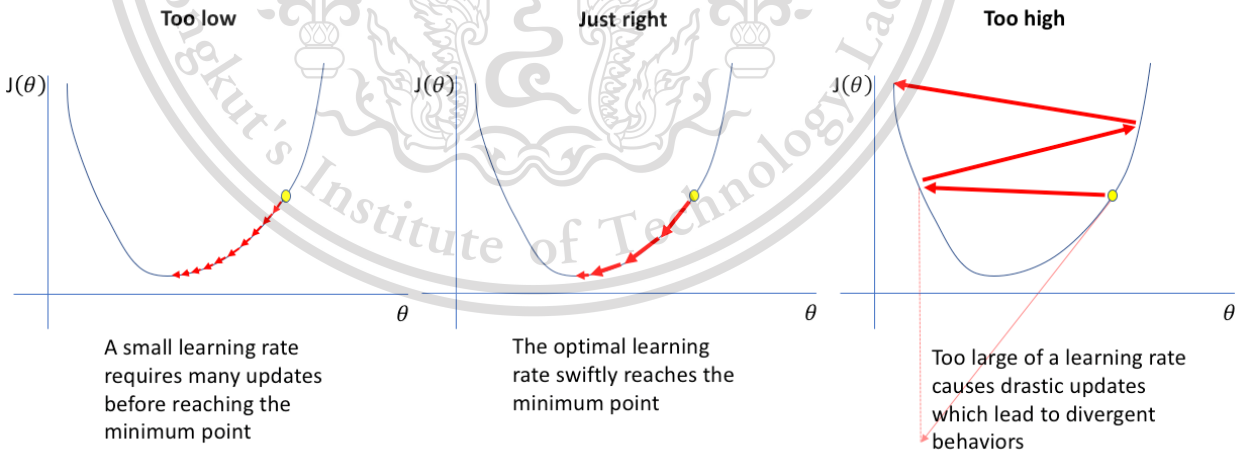
$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

รูปที่2.6 สมการ Entropy Loss

2.1.4 Optimization

การทำ optimization คือการทำให้ model สามารถเรียนรู้ที่จะเปลี่ยนแปลงค่า parameter ต่างๆ เช่น น้ำหนัก (weights) ค่า bias ซึ่งจะส่งผลให้ผลละธ์ที่ได้ออกมาจากการดำเนินการในครั้งถัดไปมีผลลัพธ์ที่จะแตกต่างกัน

Gradient Descent (GD) เป็นหนึ่งในวิธีการทำ optimization ที่ได้รับความนิยมเป็นอย่างมาก ซึ่ง gradient descent นั้นคือการหาค่าความชัน (slope) ซึ่งเราสามารถประยุกต์ใช้สิ่งนี้เข้ากับการหาค่า parameter ที่เหมาะสมได้ โดยการนำการหาค่าความชันของค่า loss ที่มีค่าน้อยสุด (ความชันมีค่าเข้าใกล้ 0) โดยจะมีอีกตัวแปรควบคุมอีกตัวหนึ่งที่สำคัญคือ Learning Rate ที่จะเป็นตัวบ่งบอกว่าเราถึงค่าความชันเป็น 0 หรือยัง



รูปที่2.7 ความสำคัญของ Learning rate เมื่อต้องการหาตำแหน่งที่มีคสามชันเข้าใกล้ 0

โดย optimization ที่นำเทคนิคการทำ gradient descent ไปประยุกต์นั้นมีมากมาย โดยจะยกตัวอย่างเทคนิคที่ได้รับความนิยมดังต่อไปนี้

1.Stochastic gradient descent(SGD): เป็นตัวที่พัฒนาต่อยอดมาจาก GD โดย SGD นั้นจะสุ่มใช้ข้อมูลเพียงแค่บางส่วน ซึ่งทำให้การทำงานในหนึ่งรอบการทำงาน (iteration) จะมีความรวดเร็วกว่า GD ที่ต้องนำข้อมูลทั้งหมดมาทำการหา slope ซึ่ง SGD จะมีบทบาทสำคัญในกรณีที่ต้องการปรับปรุง parameter จำนวนมากในแต่ละรอบการทำงาน

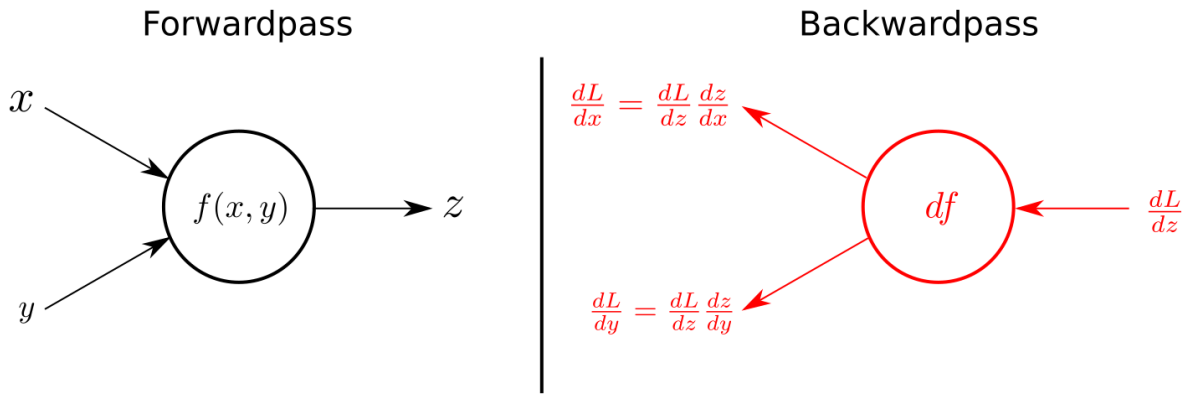
2.Adaptive Momentum Estimation (Adam): เป็นหนึ่งใน optimization ที่ได้รับความนิยมเป็นอย่างมากในปัจจุบัน เนื่องจากเป็นวิธีที่สามารถแก้ไขปัญหา local minimize ได้ เพราะการประยุกต์ momentum ซึ่งเป็นวิธีที่ให้ความสำคัญกับการเพิ่มความเร่ง เพื่อเข้าสู่จุดสูงกลางที่เป็นค่าที่มีค่าน้อยแบบสัมบูรณ์ (absolute minimize) และลดความสำคัญของทิศทางที่ไม่เกี่ยวข้อง จึงทำให้สามารถสร้างเส้นทางเพื่อไปหาความชันที่น้อยที่สุดได้

2.1.5 Forward Propagation และ Backward Propagation

ในการทำงานของ Neural Network นั้นจะแบ่งออกเป็น 2 ส่วน นั้นคือ Forward propagation และ Backward propagation

ในส่วนของ Forward propagation นั้นจะเป็นขั้นตอนของการส่งต่อ input ไปยัง layers ข้างหน้า เพื่อที่จะได้ออกมาใน output layer ในที่สุด ซึ่งในการที่จะดำเนินขั้นตอนในกระบวนการนี้นั้น ขั้นแรกเราจะมีชุดข้อมูลที่จะเป็นข้อมูล input ในลักษณะที่เป็น node ซึ่งแต่ละ node จะมีค่าน้ำหนัก (weights) ที่แตกต่างกัน โดยการคำนวนจะเกิดจากการที่เรานำค่าข้อมูลของแต่ละ node มาคูณกับค่า weight เพื่อให้ได้ค่าผลลัพธ์ออกมา โดยแต่ละ node จะส่งข้อมูลไปยัง hidden node ถัดไป โดยก่อนที่จะเข้ามายัง hidden layer จำเป็นที่จะต้องผ่าน summarization function ที่จะเป็นการรวมผลลัพธ์ที่มาจากแต่ละ node และ activation function ที่จะเป็นการตัดสินใจว่าผลลัพธ์ที่ได้จะถูกทำงาน (Activated) หรือไหม

Backward propagation เป็นขั้นตอนที่จะใช้ร่วมกับ cost function และ gradient descent เพื่อหา loss ที่จะสามารถนำไปคำนวนตัวแปรต่างๆ (weights, bias) ใหม่ เพื่อที่จะให้ค่า loss ลดลง



รูปที่2.8 การทำ Forward-pass และ Backward-pass ใน neural network

2.1.6 Train/validation/test

ในการนำ model ไปใช้งานจริงนั้น เราอาจไม่ทราบได้เลยว่า model เราจะมีความแม่นยำขนาดไหน เนื่องจากในการฝึก (Train) model ด้วยข้อมูลที่มีอยู่ทั้งหมด จะทำให้ไม่มีข้อมูลสำหรับการทดสอบว่า model นั้นมีความแม่นยำมากเพียงใด ซึ่งอาจจะส่งผลให้ overfitting model จึงทำให้ต้องมีการแบ่งชุดข้อมูลแยกออกมาเพื่อมีไว้สำหรับทดสอบโดยเฉพาะ ซึ่งการแบ่งนี้จะแบ่งชุดข้อมูลแตกย่อยออกมาเป็น 2 ชุดข้อมูลด้วยกัน ได้แก่ training set, validation และ test set

Training set เป็นชุดข้อมูลที่จะถูกแยกออกมาจากแหล่งชุดข้อมูล มักเป็นชุดข้อมูลที่มีอัตราส่วนมากกว่าชุดข้อมูลอื่น เนื่องจากชุดข้อมูลนี้จะเป็นชุดข้อมูลที่มีไว้เพื่อสำหรับการ Train model โดยเฉพาะ

Validation set เป็นชุดข้อมูลที่บางทีอาจจะถูกมองข้ามไปบ่อยครั้ง เนื่องจากเป็นชุดข้อมูลที่สร้างมาเพื่อสำหรับการปรับแก้ค่า hyperparameter ต่างๆ เพื่อที่จะให้ผลลัพธ์นั้นมีความแม่นยำมากยิ่งขึ้น

Test set เป็นชุดข้อมูลที่จะถูกแยกออกมาจากแหล่งชุดข้อมูล จึงทำให้ชุดข้อมูลนี้จะประกอบไปด้วยข้อมูลที่ไม่เคยปรากฏใน training set หรือ validation set มาก่อน โดยจะมีหน้าที่เป็นเหมือนกลุ่มตัวแทนของชุดข้อมูลในโลกความเป็นจริง (เนื่องจาก Test set คือชุดข้อมูลที่ model ยังไม่เคยพบมาก่อน ซึ่งในการเอา model ไปประยุกต์ใช้งานกับความเป็นจริง ข้อมูลบางอย่างอาจจะเป็นข้อมูลที่ model ไม่เคยพบมาก่อนเหมือนกัน)

โดยอัตราส่วนในการแบ่งชุดข้อมูลให้ออกมาเป็น 3 ส่วนนี้ จะไม่มีอัตราส่วนที่ตายตัว อย่างไรก็ตาม นักพัฒนาส่วนมากเลือกที่จะใช้อัตราส่วน 70:20:10 ตามลำดับ ในการแบ่งชุดข้อมูล

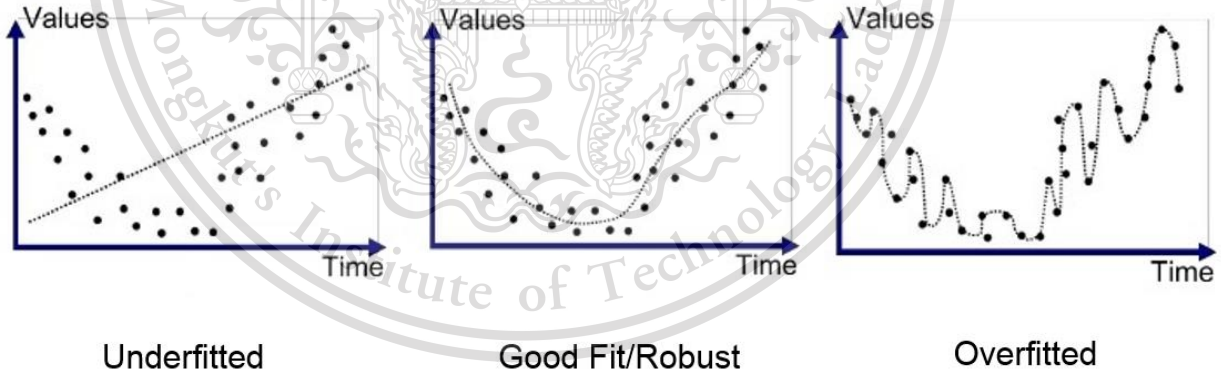
2.1.7 Underfitting/Robust/Overfitting

Neural Network เป็นโครงข่ายประสาทเทียมที่จะมอบผลลัพธ์ที่เกิดจากการทำนายมาได้ โดยการทำงานของ Neural Network นั้นเกิดมาจากสมการทางคณิตศาสตร์หลากหลายสมการเพื่อให้ได้มาซึ่งผลลัพธ์ โดยผลลัพธ์นี้ส่วนมากมาจากการเรียนรู้ผ่านข้อมูล ส่งผลให้ผลลัพธ์จะแบ่งออกเป็น 3 กลุ่มด้วยกัน

1.Under fitting model คือการที่ model ทำนายผลลัพธ์ออกมาได้โดยมีความแม่นยำค่อนข้างต่ำ เนื่องจากข้อมูลที่มีไม่มากพอที่จะคลอบคลุมการทำนายผล

2.Robust / Good fit คือการที่ model สามารถทำนายผลลัพธ์ออกมาได้ถูกต้อง แม้ว่าข้อมูลนั้นจะเป็นข้อมูลที่ไม่เคยปรากฏมาก่อนในชุดข้อมูล

3.Overfitting คือการที่ model สามารถทำนายผลลัพธ์ได้ถูกต้องเฉพาะกับข้อมูลที่เป็น training set แต่ในกรณีที่เป็ยข้อมูลที่ไม่เคยพบมาก่อน จะไม่สามารถคำนวนได้อย่างถูกต้อง



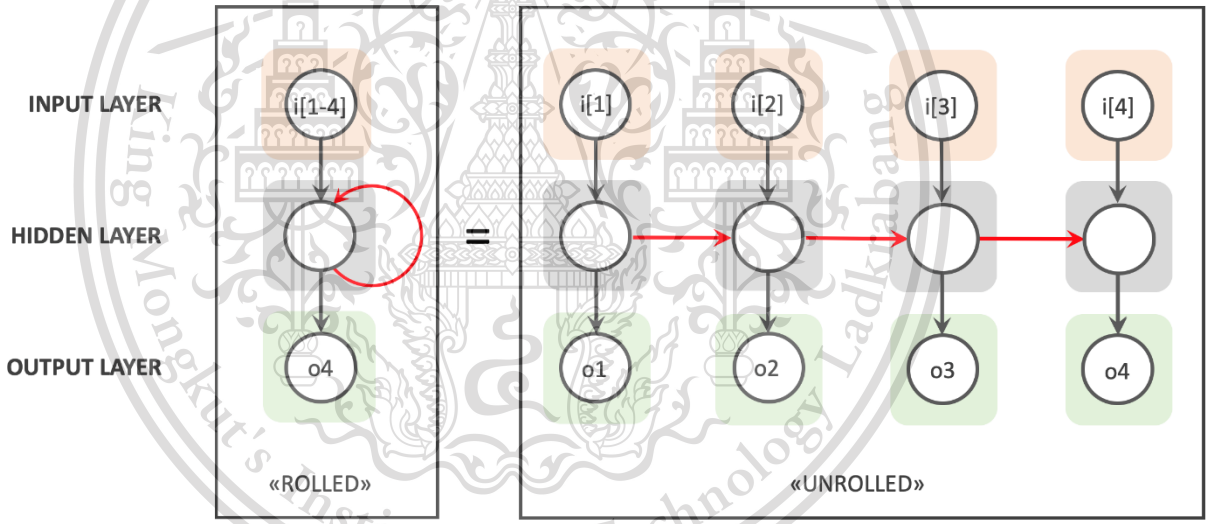
รูปที่2.9 กราฟแสดงความแตกต่างระหว่างโมเดลที่ underfitting/robust/overfitting

2.1.8 Recurrent Neural Network

Recurrent Neural Network (RNN) เป็นหนึ่งใน model ที่พัฒนาต่อยอดมาจาก ANN (Artificial Neural Network) โดยที่จุดเด่นของ model นี้คือ model นี้สามารถจัดการข้อมูลที่เป็น sequence data ได้ โดยที่ sequence data เหล่านี้คือ ข้อมูลที่มีลำดับช่วงเวลามาเกี่ยวข้อง ตัวอย่างเช่น หากเราต้องการจะทราบว่า ‘บ’ อยู่ลำดับที่เท่าไรของพยัญชนะไทย เราอาจจะต้องท่อง ก-ฮ เพื่อที่จะหาว่า บ อยู่ที่ลำดับที่เท่าใด

เช่นเดียวกันกับงาน NLP ที่ลำดับของเวลาเป็นสิ่งที่สำคัญเนื่องจาก การที่เราสามารถจดจำสถานะก่อนหน้าได้ จะทำให้เราสามารถทำนายคำที่จะเกิดขึ้นในลำดับถัดไปได้

โดยการที่ RNN สามารถจดจำข้อมูลจากสถานะก่อนหน้านี้ได้นั้นเกิดมาจากการที่ model นำ output ที่ได้ออกมาจาก hidden layers มาแปลงเป็น input สำหรับ epoch ในรอบถัดไป

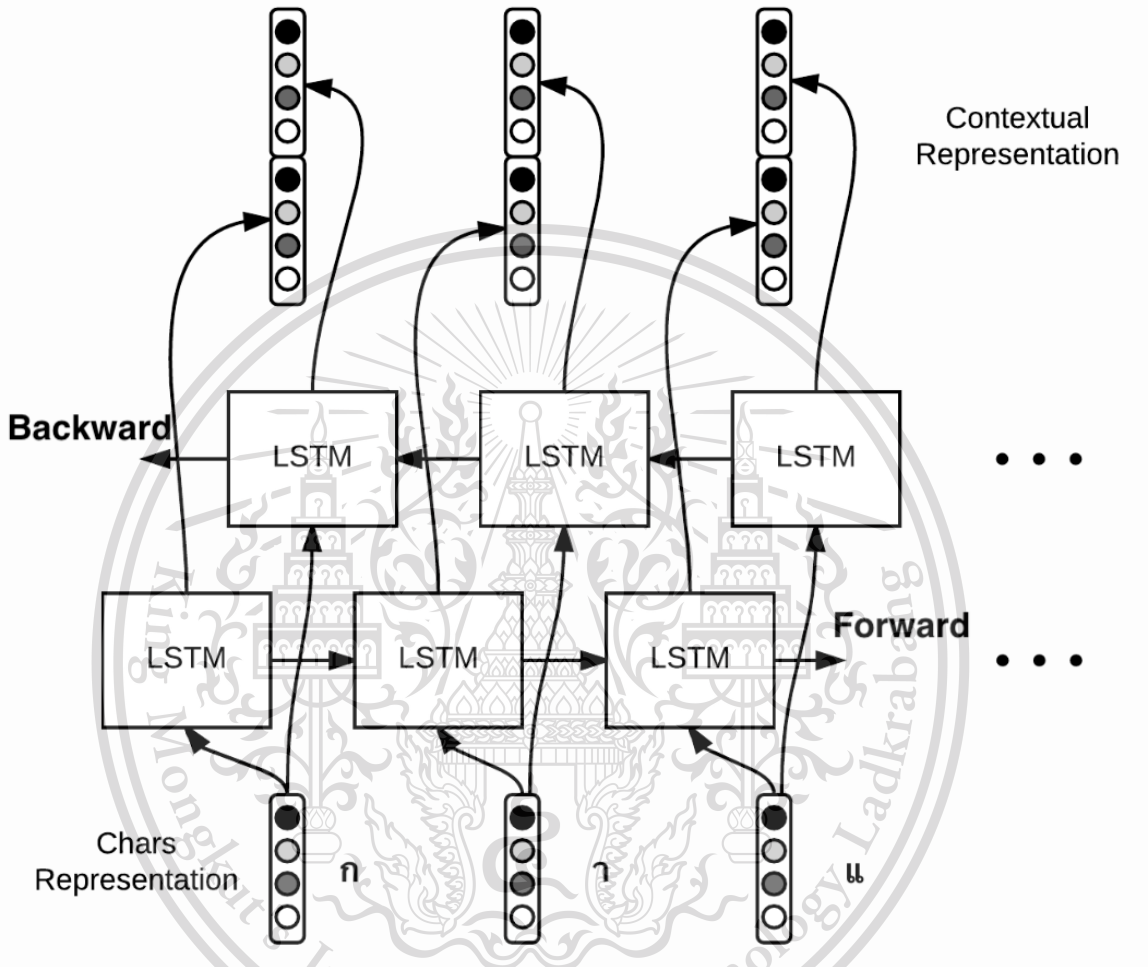


รูปที่2.10 รูปแสดงการทำงานของ Recurrent Neural Network

2.1.9 Bi-RNN

Bi direction Recurrent Neural Network เป็นตัวที่พัฒนาต่อยอดมาจาก RNN แบบดั้งเดิม โดยสิ่งที่ Bi-RNN แตกต่างจาก RNN คือ Bi-RNN นั้นสามารถอ่านเนื้อหาจากข้างหน้าไปข้างหลังได้ และสามารถอ่านเนื้อหาย้อนกลับจากข้างหลังมาข้างหน้า ซึ่งการทำแบบนี้จะสามารถคลอบคลุมเนื้อหาได้ดีมากยิ่งขึ้น และข้อแตกต่างอีกข้อคือ Bi-RNN จะเปลี่ยนจากการใช้ hidden unit เป็นการใช้ GRU หรือ

LSTM แทน โดยที่ LSTM และ GRU มีสิ่งที่คล้ายๆกันคือการเลือกที่จะจดจำเฉพาะเนื้อหาที่เกี่ยวข้องหรือสำคัญเท่านั้น เนื้อหาใดที่ไม่มความสำคัญจะถูกทิ้งออกไป เพื่อไม่ให้การทำงานในแต่ละ iteration มความล่าช้า จากการที่ต้องมาจดจำข้อมูล sequence จำนวนยาวมากเกินไป



รูปที่2.11 รูปแสดงการทำงานของ Bi-RNN

2.2 Natural Language Processing

Natural Language Processing (NLP) หรือที่จะเรียกในภาษาไทยว่า กระบวนการประมวลผลทางภาษา NLP เป็นหนึ่งในแขนงของเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligent, AI) โดยที่ NLP จะมุ่งเน้นไปที่กระบวนการการแปลงภาษาของมนุษย์ (Natural Language) ให้คอมพิวเตอร์ (Machine) เข้าใจ

คอมพิวเตอร์นั้นจะอ่านข้อมูลเป็นเลขฐานสอง(Binary Number) และทำการอ่านข้อมูลเลขฐานสองนั้นเป็นชุด Bytes เพื่อที่จะนำข้อมูล Byte เหล่านี้มาแปลงเป็นคำสั่ง (Operation) ในการทำงานต่างๆ

และด้วยเหตุผลข้างต้นจึงเป็นที่มาของ NLP ที่จะเข้ามาทำการแปลงข้อมูลที่เป็นภาษามนุษย์ ให้คอมพิวเตอร์เข้าใจถึงเนื้อหาที่มนุษย์ต้องการจะสื่อ

Tokenization

Tokenization เป็นหนึ่งในงาน (Task) ที่เป็นรากฐาน (Fundamental) ของ NLP โดยหน้าที่ของ Tokenization คือการแบ่งชุดข้อมูลออกมาเป็นหน่วย (unit) ย่อยๆ เพื่อที่จะสามารถนำ unit เหล่านี้ไปใช้งานต่อได้ ซึ่งความยากของการทำ Tokenization ในแต่ละภาษาจะมีความแตกต่างกันไป อย่างเช่น ภาษาอังกฤษจะมี stop word ที่เห็นชัดเจน คือการเว้นช่องว่าง ในขณะที่การทำ Tokenization ภาษาไทยไม่มี stop word อย่างชัดเจน จึงต้องใช้กระบวนการอื่นๆมาเกี่ยวข้องเพื่อทำการหา Tokenization ในประโยค

Corpus

Corpus/Corpora คือเอกสารที่รวบรวมคลังคำศัพท์ที่นำมาถูกใช้งานในด้าน NLP โดยส่วนมากนั้น Corpus นั้นยิ่งมีขนาดใหญ่นั้นยิ่งดี เนื่องจากยิ่งมีคลังคำศัพท์ใหญ่มากเท่าใด นั้นหมายถึงประสิทธิภาพของระบบที่จะรู้จักคำนั้นๆ

Bigram

Bigram คือการจดจำคำศัพท์สิ่งของรูปแบบหนึ่ง โดยที่ในปกติเราจะจดสิ่งของเพียงสิ่งๆเดียว แต่การใช้ Bigram จะเป็นการจดจำสิ่งของใดๆที่อยู่ใกล้เคียงกับสิ่งของที่เราต้องการที่จะจดจำ

Bigram คือหนึ่งในเทคนิคที่สามารถนำมาประยุกต์ใช้กับงานทางด้าน nlp ได้ นั้นคือเราสามารถจดจำคำศัพท์ถัดไปต่อจากคำศัพท์ที่เราต้องการจะจดจำ ซึ่งการนำ Bigram มาใช้งานจะทำให้เราสามารถทำนายผลลัพธ์ของคำได้

ตัวอย่างการบันทึกของ bigram

เมื่อให้ประโยคที่จะจดจำคือประโยค : “ผมขายน้ำเปล่า”

Tokenize ของประโยค: “ผม” “ขาย” “น้ำ” “เปล่า”

Tokenize ของประโยคที่ใช้ bigram: “ผม,ขาย” “ขาย,น้ำ” “น้ำ,เปล่า”

ซึ่งจากตัวอย่างข้างต้น หากมีการป้อนข้อมูลที่ผิดพลาด อย่าเช่น “น้ำเผล่า” ซึ่งเราจะเห็นว่ามีการป้อนข้อมูบผิด จากคำว่า “น้ำเปล่า” เป็น “น้ำเผล่า” ซึ่งคำว่า “น้ำเผล่า” ไม่ได้มีการบันทึกไว้ใน bigram ของเรา จึงทำให้เราสามรถอนุมานได้ว่า คำที่ถูกป้อนนั้นควรจะเป็นคำว่า “น้ำเปล่า” นั้นเอง

Term frequency – Inverse Document Frequent

Term frequency – Inverse Document Frequent (TF-IDF) คือ กระบวนการการวัดข้อมูลทางสถิติว่าเอกสารทั้งสองเอกสารมีความคล้ายหรือแตกต่างกันมากเพียงใด โดยที่ TF-IDF นั้นมาจากการนำคำสองคำมาผสมกันนั้นคือ Term Frequency และ Inverse Document Frequent

Tern Frequency คือการนับคำซ้ำของคำศัพท์ในเอกสารนั้นๆ เพื่อแปลงค่าออกมาเป็นตัวเลข ซึ่งยิ่งคำ (words) นั้นเกิดขึ้นซ้ำมากเท่าใดในเอกสาร คำนั้นย่อมมีค่ามากยิ่งขึ้น

$$tf(word, document) = \frac{frequncy\ of\ word\ in\ document}{total\ number\ of\ word\ in\ document}$$

Inverse Document Frequent คือการวัดความสำคัญของคำใดๆ ในเอกสาร (texts) ทั้งหมด เพื่อแปลงค่าออกมาเป็นตัวเลข ซึ่งยิ่งคำใดเกิดขึ้นบ่อยในหลายเอกสารนั้นหมายความว่า คำนั้นมีความสำคัญน้อย เนื่องจากคำนั้นมีโอกาสที่จะเป็น คำศัพท์ทั่วไป (common word)

$$idf(word, document) = log \frac{total\ doument}{document\ that\ contain\ word}$$

TF-IDF คือการนำผลลัพธ์ที่ได้จาก Term Frequency และ Inverse Document Frequent มาคูณกัน ซึ่งผลลัพธ์จากการคูณกัน จะหมายความว่าเราจะได้ผลลัพธ์ออกมาเป็นตัวเลข ซึ่งตัวเลขนั้นจะเรียกว่า TF-IDF weight

คำศัพท์ที่มีค่า TF-IDF weight มากจะหมายความว่าเกิดขึ้นบ่อยในเอกสารนั้น (Term Frequency) และไม่ได้เจอในหลายเอกสาร (Inverse document Frequent) จึงทำให้คำศัพท์นั้นเป็นคำศัพท์ที่สำคัญ

จากการทำ TF-IDF จะทำให้เราสามารถกรอง common word ออกไปจากเอกสารเราได้ และทำให้เราเข้าใจเนื้อหา (context) ที่สำคัญของเอกสารนั้นๆ

2.3 Python

Python เป็นภาษาโปรแกรมมิ่งระดับสูง (High-level Programming Language) ที่มีลักษณะการทำงานแบบ Object Oriented Programming ซึ่งเป็นภาษาที่ได้รับความนิยมเป็นอย่างมากในปัจจุบัน เนื่องจาก Python เป็นภาษาโปรแกรมมิ่งระดับสูง จึงทำให้สามารถเข้าใจความหมายได้โดยไม่ยากเนื่องจาก Python ดัดแปลงเรื่องไวยากรณ์ (Syntax) ให้เข้าใจได้ง่ายๆ

Python มี library มากมายสำหรับงานด้านต่างๆ ตั้งแต่งานทางด้าน แอพพลิเคชันบนเว็ปไซต์ (Website-based Application), วิทยการข้อมูล (Data science), การใช้ภาพเป็นตัวประสานกับผู้ใช้ (Graphical User Interface) และงานด้านอื่นๆอีกมากมาย นั้นจึงเป็นอีกหนึ่งสาเหตุที่ Python ได้รับความนิยมจากโปรแกรมเมอร์ทั่วโลก

Textdistance

Textdistance เป็น library ของ python ที่สามารถเข้ามาช่วยการทำงานทางด้าน NLP ได้ส่วนหนึ่ง โดย textdistance จะสามารถเปรียบเทียบความคล้ายคลึงระหว่างคำ (string similarity) ระหว่างคำ ผ่านสมการทางคณิตศาสตร์ โดยเมื่อหลังจากที่คำนวนเสร็จเรียบร้อยแล้ว จะได้ค่าคะแนน (score) ออกมา

Text similarity สามารถแบ่งออกไดเป็นหลายรูปแบบ โดยผู้จัดทำจะขอยกตัวอย่างเพียงแค่ 2 รูปแบบที่ได้ทำการทดลองเท่านั้น นั้นคือ Edit based similarity และ Tokenize similarity

1.Edit based Similarity: การเปรียบเทียบแบบนี้จะเน้นไปที่การเปนรยบเทียบความแตกต่างของอักขระแต่ละตัวระหว่างที่ต้องการจะเปรียบเทียบ โดยการเปรียบเทียบนั้นจะเป็นการเปรียบเทียบผ่านกระบวนการ (Operation) เพื่อที่จะเปลี่ยนแปลงจาก คำหนึ่งเป็นอีกคำ ซึ่งการเปรียบเทียบแบบนี้จะเป็นเพียงการสร้างความเข้าใจแค่ระดับอักขระ (Atomic Understanding) มิใช่ความเข้าใจในระดับความหมาย (Semantic Meaning)

ตัวอย่างของ Edit based Similarity ใน textdistance

1.1 Levenshtein distance: เป็นการเปรียบเทียบระหว่างคำสองคำ โดยเปรียบเทียบผ่าน 3 Operation ได้แก่ การเพิ่ม (Added), การลบ (Deleted), การแทนที่ (Replaced) เพื่อที่จะเปลี่ยนคำต้นฉบับเป็นคำที่ต้องการเปลี่ยนแปลง ซึ่งยิ่งมีค่าคะแนนน้อยจะหมายความว่า คำต้นฉบับและคำที่ต้องการมีควาคล้ายคลึงกันมาก

1.2 Hamming distance: เป็นการเปรียบเทียบระหว่างคำสองคำ โดยเปรียบเทียบจากการนำคำต้นฉบับและคำที่ต้องการจะเปลี่ยนแปลง แล้วทำการเปรียบเทียบตัวอักษรของทั้งสองคำทีละตำแหน่ง โดยถ้าหากตัวอักษรในตำแหน่งที่เปรียบเทียบไม่เหมือนกัน จะมีค่าคะแนนเพิ่มขึ้นทีละ 1 ซึ่งยิ่งค่าคะแนนน้อย จะหมายความว่าคำทั้งสองคำมีความคล้ายคลึงกันมาก

2.Token based similarity: เป็นการเปรียบเทียบระหว่างชุดของ tokenize ซึ่งจะแตกต่างกับ Edit based similarity ซึ่งจะเปรียบเทียบระหว่าง string กับ string โดย Token based similarity จะเป็นการหาความคล้ายคลึงของ token ระหว่างเอกสาร (text) ผ่านสมการทางคณิตศาสตร์เพื่อที่จะได้คะแนนที่สามารถเปรียบเทียบความคล้ายคลึงระหว่างเอกสารทั้งสอง ซึ่งความคล้ายคลึงนี้จะสามารถพัฒนาต่อยอดเพื่อไปทำเป็น vector เพื่อที่จะเข้าใจความคล้ายคลึงทางความหมาย (Semantic Meaning) ได้

ตัวอย่างของ Token based Similarity ใน textdistance

2.1 Cosine Similarity: เป็นการเปรียบเทียบเอกสารสองเอกสารว่ามีความคล้ายคลึงหรือใกล้เคียงกันมากเพียงใด โดยการเปรียบเทียบความซ้ำซ้อนของ token ของทั้งสองเอกสาร ซึ่งผลลัพธ์ที่ออกมาจะมีค่าตั้งแต่ 0 ถึง 1 โดย 0นั้นหมายถึงเอกสารทั้งสองไม่มีความคล้ายคลึงกันเลย และ 1 หมายถึงเอกสารทั้งสองมีความคล้ายคลึงกันอย่างสมบูรณ์ โดย Cosine Similarity จะมีสมการดังนี้

$$similarity(A,B) = \frac{A \cdot B}{||A||\ x\ ||B||} = \frac{\sum_{i=1}^{n} A_i\ x\ B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ x\ \sqrt{\sum_{i=1}^{n} B_i^2}}$$

2.2 Jaccard Similarity: เป็นการเปรียบเทียบเอกสารสองเอกสารว่ามีความคล้ายคลึงหรือใกล้เคียงกันมากเพียงใด โดยการเปรียบเทียบความซ้ำซ้อนของ token ของทั้งสองเอกสาร ซึ่ง

ผลลัพธ์ที่ออกมาจะมีค่าตั้งแต่ 0 ถึง 1 โดย 0นั้นหมายถึงเอกสารทั้งสองไม่มีความคล้ายคลึงกันเลย และ 1 หมายถึงเอกสารทั้งสองมีความคล้ายคลึงกันอย่างสมบูรณ์ โดย Jaccard Similarity จะมีสมการดังนี้

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Scikit Learn

Scikit Learn เป็นหนึ่งใน open-source library ที่เกี่ยวกับการทำงานทางด้านวิทยาการข้อมูล ที่ได้รับความนิยมเป็นอย่างมาก เนื่องจากใน Scikit Learn มีฟังก์ชันที่หลากหลายที่สามารถให้ใช้งานได้ อย่างเช่น random forest, regression, classification, DBScan รวมทั้งการแปลงข้อมูลให้เป็น vector

Scikit Learn ยังถูกออกแบบมาเพื่อให้สามารถใช้งานร่วมกับ Numpy ซึ่งเป็นอีกหนึ่ง library ที่สำคัญสำหรับการทำงานทางด้านวิทยาการข้อมูล

ReGex

Regular Expression (ReGex) คือหนึ่งในรูปแบบการกำหนดรูปแบบของคำ/กลุ่มคำ ที่ต้องการ โดยสามารถเขียนกำหนดรูปแบบได้ทั้งกำหนดอักขระที่ต้องการ, กำหนดตัวเลขที่ต้องการ, จำนวนของอักจระในข้อความ โดยที่ ReGex นั้นสามารถนำไปประยุกต์ใช้ได้กับหลายกรณีด้วยกัน ไม่ว่าจะเป็นการกำหนดรูปแบบของการป้อนข้อมูลเพื่อให้เป็นไปตามที่ admin ต้องการ, การค้นหาคัดกรองข้อมูลโดยขั้นต้น (มีคำที่ match กับคำที่ค้นหาไหม)

Pandas

Pandas เป็น library หนึ่งของ python ซึ่งได้รับความนิยมอย่างมากในกลุ่มที่ทำงานเกี่ยวกับวิทยาการข้อมูล (Data Science) เนื่องจาก pandas สามารถใช้งานได้กับข้อมูลหลากหลายรูปแบบ (Format) ซึ่งหนึ่งในนั้นคือไฟล์ประเภท excel

ซึ่งสอดคล้องกับข้อมูลที่ต้องการจะใช้งาน เนื่องจากข้อมูลที่มีอยู่นั้นเป็นข้อมูลไฟล์ประเภท excel ผู้จัดทำจึงเลือกใช้ pandas ในการอ่านข้อมูลที่ต้องการจากข้อมูลทั้งหมดในไฟล์ และทำการบันทึกข้อมูลนั้นในรูปแบบของ Excel

2.5 Database, Relational database, NoSQL

ฐานข้อมูล (Database) คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ด้วยกัน ซึ่งข้อมูลจะต้องมีความสัมพันธ์ระหว่างข้อมูลด้วยกันเอง ซึ่งการรวบรวมข้อมูลเข้าสู่ฐานข้อมูล จะส่งผลให้การจัดการข้อมูลสามารถเป็นไปได้อย่างไม่ยุ่งยาก ทำให้ผู้ใช้งานสามารถสร้างฐานข้อมูล, เรียกใช้ข้อมูล, ปรับปรุงข้อมูล และ ลบข้อมูล

ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) คือ การจัดเก็บข้อมูลในเชิงความสัมพันธ์ของข้อมูล โดยส่วนมากมักจะแสดงผลออกมาในรูปแบบตาราง (table) ซึ่งจะมีลักษณธโครงสร้างไม่ซับซ้อนซึ่งจะมีเทคโนโลยีที่เกี่ยวข้องคือ SQL

Structured Query Language (SQL) เป็นฐานข้อมูลที่มีรูปแบบการสร้างโดยอิงตามการสร้าง Relational Database

No-SQL database เป็นฐานข้อมูลที่จะมีความยืดหยุ่นมากกว่า SQL เนื่องจากจะไม่มีรูปแบบในการจะดเก็บที่ตายตัว การจัดเก็บจะอยู่ในรูปแบบของ key-value โดยที่ key จะเป็นเหมือน index ที่สามารถแจ้งได้ว่าต้องการที่จะค้นหาสิ่งใด และ value คือค่าของ key ที่จะแตกต่างกันไปตามวัตถุ (object) ที่บันทึก

2.6 Angular

Angular นั้นเป็น framework สำหรับฝั่ง front end ซึ่งได้รับความนิยมเป็นอย่างมากในการพัฒนา web app ซึ่งจะประกอบไปด้วย javascript+css+html โดยที่ angular จะมีการออกแบบด้วยสถาปัตยกรรม Model View Controller (MVC) โดยที่ MVC จะประกอบไปด้วย model(m) view(v) controller(c)

Model (m) คือ object ที่ทำหน้าที่เป็นตัวแทนของข้อมูล ซึ่งการมองข้อมูลเป็นวัตถุนั้นจะไปสอดคล้องกับหลักการเขียน program เชิงวัตถุ (Object Oriented Programming /OOP) โดยที่ object นั้นจะเป็นวัตถุใดวัตถุหนึ่ง ที่มีคุณสมบัติ (attributes) และ พฤติกรรม (Method) ของ object นั้น ตัวอย่าง เช่น บ้านพักที่มี attributes สีแดง และมี method คือ สามารถเป็นที่พักพิงได้

View (v) คือ object สำหรับการแสดงผลต่างๆ ซึ่ง object เหล่านี้ สามารถเปรียบเทียบให้คล้ายตคลึงกับ interface ก็เป็นได้ ตัวอย่างเช่น แบบฟอร์มถามตอบคำถาม, การแสดงผลของหน้า web

Controller (c) คือ object สำหรับคอยควบคุม model และ view ให้ทำงานร่วมกันได้ ตลอดจนสิ้นเรื่องการทำงานของคำสั่งใดๆ ตัวอย่างเช่น การรับคำสั่งจาก interface เพื่อนำคำสั่งนั้นไป ทำงานเพื่อให้สอดคล้องกับ model

บทที่ 3

การออกแบบและพัฒนา

ในการดำเนินงานสร้างสรรค์ RECOMMENDATION Asset Search นั้นถูกออกแบบมาเพื่อเป็นตัวช่วยในการค้นหาข้อมูลสินทรัพย์ประเภทอสังหาริมทรัพย์ให้กับพนักงานฝ่ายประเมินราคาสินทรัพย์ประเภทอสังหาริมทรัพย์ ในบริษัท ธนาคารพาณิชย์ซึ่งการพัฒนาระบบนี้ขึ้นมานั้นเพื่อที่จะทดแทนการค้นหาข้อมูลแบบเดิมด้วยเลขรหัสงาน ซึ่งมีความซับซ้อน และกินเวลา ผู้จัดทำจึงเล็งเห็นปัญหานี้ และได้ทำการพัฒนาระบบ RECOMMENDATION Asset search เพื่อให้การทำงานนั้นเป็นไปได้อย่างยืดหยุ่นมากขึ้น ซึ่งมีการดำเนินงานดังต่อไปนี้

3.1 ปัญหาที่พบ

RECOMMENDATION Asset search นั้นเกิดมาได้จากการที่ผู้จัดทำได้มีโอกาสเข้าไปฝึกงานที่บริษัท ธนาคารพาณิชย์ซึ่งได้เข้าไปฝึกในตำแหน่ง IT and Operation support ซึ่งเป็นแผนกที่มีหน้าที่ในการช่วยเหลือและให้การสนับสนุนแก่แผนกอื่นๆในองค์กร ซึ่งผู้จัดทำได้มีโอกาสเข้าไปเยี่ยมชมที่แผนกประเมินราคาทรัพย์สินฝ่ายอสังหาริมทรัพย์ (Mortgage Team) โดยการประเมินราคาทรัพย์สินในบางครั้ง จำเป็นที่จะต้องประเมินราคาอิงตามราคาตลาด ซึ่งข้อมูลการประเมินราคทรัพย์สินนั้นจะถูกบันทึกไว้ในระบบของธนาคาร ดังนั้นพนักงานฝ่ายการประเมินทรัพย์สินจึงจำเป็นที่จะต้องค้นหาข้อมูลทรัพย์สินเปรียบเทียบเพื่อที่จะหาราคาและปัจจัยที่เกี่ยวข้องในการประเมินราคาทรัพย์สิน ซึ่งการค้นหาทรัพย์สินที่ต้องการจะเปรียบเทียบนั้นจำเป็นที่จะต้องค้นหาด้วยเลขรหัสทรัพย์สิน ซึ่งเป็นการค้นหาที่กินเวลาโดยไม่จำเป็น เนื่องจากพนักงานฝ่ายประเมินราคาทรัพย์สิน จำเป็นที่จะต้องเข้าไปค้นหาเลขรหัสงานที่ต้องการจะเปรียบเทียบในระบบฐานข้อมูล ซึ่งมีขนาดใหญ่ ดังนั้นผู้จัดทำจึงเล็งเห็นปัญหาตรงนี้ และได้คิดค้นหาวิธีการที่จะสามารถทำการค้นหาข้อมูลได้อย่างมีความยืดหยุ่นมากยิ่งขึ้น อย่างเช่นการค้นหาด้วยชื่อทรัพย์สิน, ที่อยู่ของทรัพย์สิน ซึ่งจะสามารถลดเวลาการทำงานของพนักงานลงได้

3.2 การออกแบบ data pipeline

การออกแบบการทำงานของ RECOMMENDATION Asset search มีพื้นฐานมาจากการออกแบบ data pipeline ซึ่งการออกแบบ data pipeline นั้นมีจุดมุ่งหวังเพื่อที่จะให้เห็นการไหลของ

กระแสข้อมูลจากตั้งแต่ต้นน้ำถึงปลายน้ำ เปรียบเสมือนกับการออกแบบโครงสร้างท่อลำเลียงน้ำประปา (pipeline) ที่จะเป็นการลำเลียงน้ำประปาจากแหล่งกำเนิดน้ำไปจนถึงการนำน้ำประปาไปใช้งาน

พื้นฐานของการออกแบบ data pipeline จะประกอบไปด้วยการทำงาน 4 ตอน (phase) ได้แก่

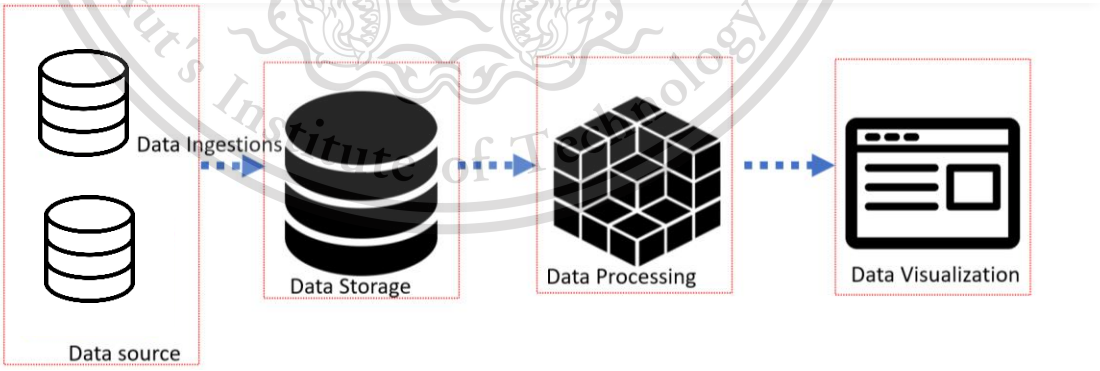
-การนำเข้าข้อมูล (data Ingestion) คือการแปลงข้อมูลข้อมูลดิบ (raw data) ให้เป็นรูปแบบ (format) ที่สามารถนำไปใช้งานใน phase ต่อไปได้

-การจัดเก็บข้อมูล (data Storage) คือกระบวนการในการจัดเก็บข้อมูล เพื่อที่เรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพใน phase ถัดไป

-การจัดการข้อมูล (Data processing) คือกระบวนการการนำข้อมูลที่มีมาประมวลผล เพื่อให้ได้ผลลัพธ์ที่จะสามารถนำไปใช้งานได้ใน phase ถัดไป

-การนำข้อมูลไปใช้ (Data Consumption) คือการนำข้อมูลที่ได้มาจาก phase ก่อนหน้ามาใช้งาน ซึ่งอาจจะเป็นการทำรายงาน (Report) หรือการแสดงผล (Visualization)

โดยในการออกแบบ RECOMMENDATION Asset search จะยึดหลักการออกแบบ data pipeline ด้านต้นเป็นพื้นฐานการดำเนินงาน และจะมีการประยุกต์การออกแบบเล็กน้อยเพื่อให้เหมาะสมกับโครงงาน โดยที่ RECOMMENDATION Asset search จะมีการออกแบบ data pipeline ดังต่อไปนี้



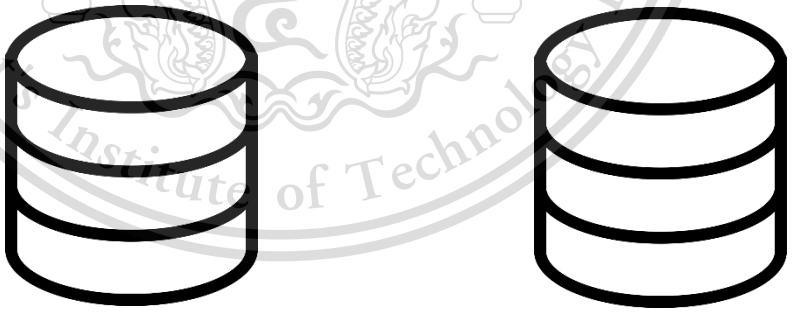
รูปที่3.1 แผนผัง Data Pipeline ของโครงงานนี้

3.2.1 Data Source

แหล่งข้อมูล (Data Source) คือแหล่งข้อมูลของการทำงาน โดยที่ data source นั้นสามารถมาได้จากหลากหลายแห่ง, มีเนื้อหาข้างในแตกต่างกันไป หรือแม้กระมั่งเป็นข้อมูลที่ไม่เป็นรูปแบบ (Unstructured Data)

โดยแหล่งข้อมูลใน RECOMMENDATION Asset search นั้นเป็นข้อมูลมาจากบริษัท outsource ที่ช่วยทำงานประเมินนราคาทรัพย์สินให้กับบริษัท ธนาคารพาณิชย์ซึ่งสาเหตุที่ผู้จัดทำเลือกใช้ข้อมูลจาก outsource นั้นเป็นเพราะข้อมูลของ outsource นั้นเป็นข้อมูลที่ยังไม่ได้จัดเก็บไว้ในบริษัท ซึ่งทำให้ไม่สามารถดึงประโยชน์การทำงานแบบขับเคลื่อนในข้อมูล (data driven) ได้อย่างเต็มที่ ผู้จัดทำจึงเล็งเห็นปัญหาตรงนี้ โดยเลือกที่จะนำข้อมูล outsource บางส่วนมาใช้งาน เพื่อที่จะสร้าง RECOMMENDATION Asset search ซึ่งจะเป็นระบบค้นหาสินทรัพย์ประเภทอสังหาริมทรัพย์ โดยหวังผลว่า RECOMMENDATION Asset search จะเป็นผลงานทดลอง (prototype) ที่สามารถจะนำไปประยุกต์ใช้กับระบบฐานข้อมูลจริงได้

ซึ่ง data source ที่มาจาก outsource นั้นมาจากบริษัทที่ให้บริการประเมินราคาสินทรัพย์ประเภทอสังหาริมทรัพย์ แต่สิ่งที่แตกต่างกันคือรายละเอียดข้อมูล (Feature) ภายใน ซึ่งไม่เหมือนกัน จึงต้องทำการสร้างมาตรฐาน (Standardization) ให้ข้อมูลมีมาตรฐานเดียวกัน



รูปที่3.2 data source ในการทำงานของ RECOMMENDATION Asset search

3.2.2 Data Ingestion

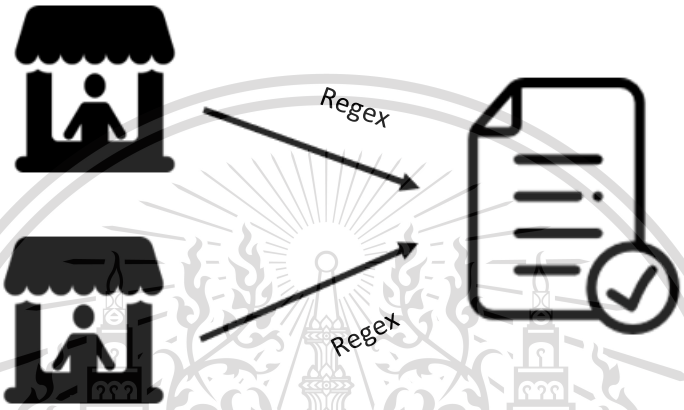
การนำเข้าข้อมูล (Data Ingestion) คือการแปลงข้อมูลจากแหล่งข้อมูลที่มีความหลากหลายให้มีมาตรฐานเดียวกัน เพื่อที่จะสามารถนำไปใช้งนได้จริง เนื่องจากข้อมูลของ Data source บางครั้งจะเป็นข้อมูลมหัต (Big data) ซึ่งเป็นชุดข้อมูล (Data set) ที่มีขนาดใหญ่ ซึ่งมีความเป็นไปได้ที่จะเป็น Unstructured Data จึงต้องทำการแปลงข้อมูลให้เป็นข้อมูลในรูปแบบที่ต้องการ

โดยใน RECOMMENDATION Asset search จะมีการนำเข้าข้อมูลจาก data source ซึ่งจะมาเป็นข้อมูลในรูปแบบของ spreadsheet ในโปรแกรม excel (.xls) ซึ่งมีข้อมูลที่มากมายหลากหลายและมีความแตกต่างกันจากทั้งสองบริษัท ทางผู้จัดทำจึงทำการคัดแยกข้อมูล (extracted data) โดยเลือกเฉพาะ Features ที่คิดว่าสำคัญต่อการค้นหา

ซึ่งหลังจากเลือก features ที่สนใจมาได้แล้ว เราจำเป็นที่จะต้องคัดแยกข้อมูลจากทั้งสองออกมา โดยเมื่อทำการคัดเลือกข้อมูลทั้งสองออกมาเสร็จทำให้พบว่าการคัดเลือกข้อมูลออกมานั้นมีข้อมูลบาง Feature ที่มีปัญหา อย่างเช่นข้อมูลมีค่า NaN,ข้อมูลไม่ครบ (missing data) จึงทำให้ต้องทำการทำความสะอาดข้อมูล (Cleaning Data) เพื่อกำจัดข้อมูลที่ไม่สามารถใช้งานได้ออกไปก่อน ซึ่งการทำแบบนี้นั้น จะทำให้ขั้นตอนในอนาคตอย่างการทำ Processing Data สามารถทำงานได้มีประสิทธิภาพมากยิ่งขึ้น เนื่องจากข้อมูลที่เป็นข้อมูลขยะ (Garbage Data) จะถูกจำกัดออกไปแล้ว และเมื่อทำการทำความสะอาดเสร็จแล้วจึงแปลงข้อมูลกลับออกมาเป็นในรูปแบบของไฟล์ประเภท excel (.xlsx) ซึ่งจะถูกแปลงเพื่อไปใช้งานในอนาคต

โดยขั้นตอนการนำข้อมูลเข้าสู่ระบบและการทำความสะอาดข้อมูลนั้นจะทำงานด้วย python และ pandas เนื่องจาก pandas มีความสามารถในการจัดการข้อมูลประเภท excel ได้เป็นอย่างดี ไม่ว่าจะเป็นการเปิดไฟล์ประเภท excel (.xls,.xlsx) รวมไปถึงการอ่านและเขียนข้อมูลลงไปเพื่อที่จะจัดการข้อมูล (data manipulation) ให้ออกมาในรูปแบบที่ต้องการ ซึ่งนั้นหมายความว่าเราเลือกที่จะเลือกเฉพาะข้อมูลที่เราต้องการได้ และทำการส่งออก(export) ออกมาเพื่อที่จะมาตรวจสอบข้อมูล (Data Investigation) เพื่อที่จะตรวจหาข้อผิดพลาด และทำการแก้ไขเพื่อให้ข้อมูลมีความสะอาดได้ แต่อย่างไรก็ตาม แม้ pandas จะสามารถจัดการข้อมูลได้เป็นอย่างดี ข้อมูลบางอย่างยังจำเป็นที่จะต้องอาศัยการแก้ไขด้วยมือ (manual)

ในการทำความสะอาดข้อมูลนั้น นอกจากจะใช้ pandas ซึ่งเป็น library ที่สำคัญในการทำ data manipulation ผู้จัดทำยังใช้ re ซึ่งเป็น library หรับการทำ Regular Expression (Regex) โดยจะนำมาช่วยสร้าง standard ให้กับข้อมูล อาทิเช่นข้อมูลรหัสทางภูมิศาสตร์ (Geocode) ซึ่งในที่นี้หมายถึง latitude และ longitude ซึ่งข้อมูลบางฉบับอาจจะมีเครื่องหมายองศา (Degree sign) ซึ่งเป็นข้อมูลที่ไม่จำเป็น ทำให้เราเก็บแค่ตัวเลขก็เพียงพอ



รูปที่3.3 การสร้าง standard ให้ข้อมูลให้มีมาตรฐานเดียวกัน

3.2.3 Data Storage

การจัดเก็บข้อมูล (Data Storage) คือกระบวนการจัดเก็บข้อฒุลเพื่อให้สามารถเรียกใช้งานได้อย่างง่ายดาย โดยการจัดเก็บข้อมูลนั้นอาจจะหมายถึงการบันทึกข้อมูลลงฐานข้อมูล

ฐานข้อมูล (Data Base) นั้นหมายถึงการรวบรวมข้อมูลเข้าด้วยกัน เพื่อที่จะสามารถสร้างฐานข้อมูล (Create) เรียกใช้ข้อมูล (Retrieve) แก้ไขข้อมูล (Update) และการลบข้อมูล (Delete) โดยที่ผู้จัดทำได้มีตัวเลือก database สองตัวนั้นคือ mongoDB และ Elasticsearch

MongoDB เป็นฐานข้อมูลที่ได้รับความนิยมเป็นอย่างมาก โดย MongoDB มีการจัดเก็บข้อมูลแบบ NoSQL ซึ่ง MongoDB นั้นมีข้อดีคือข้อมูลจะอยู่ในรูปแบบที่คล้ายคลึงกับ JSON ซึ่งข้อมูลแบบ JSON นี้สามารถนำไปใช้งานได้กับงานที่เกี่ยวข้องกับข้อมูลได้สะดวก

Elasticsearch เป็นฐานข้อมูลที่ได้รับความนิยมเป็นอย่างมากเช่นกัน เนื่องจาก Elasticssearch นั้นมีความสามารถในการค้นหาเอกสารได้อย่างรวดเร็ว

ซึ่งทั้ง MongoDB และ Elasticsearch เป็นสองตัวเลือกที่น่าสนใจเป็นอย่างมาก เนื่องจากเป็นฐานข้อมูลที่มีความสามารถที่มีความสามารถที่ดีทั้งคู่ และนั้นรวมไปถึงมีชุมชน (Community) ขนาดใหญ่ ที่จะสามารถให้คำปรึกษาเมื่อพบปัญหาได้

อย่างไรก็ตามผู้จัดทำเลือกที่จะใช้ mongoDB เนื่องจากเมื่อเทียบประสิทธิภาพ (performance) แล้ว MongoDB นั้นสามารถ Retrieve ได้ดีกว่า Elasticsearch และถึงแม้ว่า Elasticsearch จะสามารถค้นหาเอกสารได้ แต่นั้นไม่จำเป็นสำหรับ RECOMMENDATION Asset search เนื่องจากจะเป็นการทำหน้าที่ซ้ำซ้อนกับการทำ Enterprise Search ที่ผู้จัดทำได้ออกแบบ

ในการเรียกใช้งาน mongoDB ผ่านภาษา python นั้นจำเป็นที่จะต้องใช้ library ในการเชื่อมต่อเข้าสู่ฐานข้อมูล ซึ่งผู้จัดทำเลือกช้ pymongo ซึ่งมี API สำหรับการติดต่อเข้ากับฐานข้อมูล mongoDB และนอกจากจะสามารถเชื่อมต่อเข้าสู่ฐานข้อมูลได้แล้ว ยังสามารถที่จะสร้างฐานข้อมูล เพิ่มข้อมูล ผ่าน Application Programing Interface (API)

รูปที่3.4 mongoDB เป็นฐานข้อมูลในการใช้งานในโครงการนี้

3.2.4 Data Processing

Data Processing / Data Transformation คือการนำข้อมูลมาประมวลผลเพื่อให้ได้ผลลัพธ์ที่ต้องการตามวัตถุประสงค์ของผู้พัฒนาว่าต้องการจะนำข้อมูลไปทำอะไร

โดยใน RECOMMENDATION Asset search มีการทำ data processing คือการสร้าง Enterprise Search ที่จะเข้ามาทดแทนการทำงานของการค้นหาแบบเดิมที่ค้นหาผ่านเลขรหัสงานซึ่งเป็นปัญหาดั่งที่กล่าวไว้ข้างต้น

Enterprise Search คือหนึ่งในรูปแบบการค้นหาข้อมูลที่ถูกออกแบบมาเพื่อช่วยเหลือพนักงานที่ต้องการจะค้นหาข้อมูลจากหนึ่งแหล่งฐานข้อมูลหรือหลากหลายแหล่งฐานข้อมูลในการค้นหาเพียงการค้นหาเดียว ซึ่งข้อมูลในการค้นนหานั้นสามารถอยู่ในรูปแบบใดก็ได้และจากฐานข้อมูลใดก็ได้ในฐานข้อมูลของบริษัท, เอกสารการบันทึก, email-server ของบริษัท และอื่นๆที่เป็นข้อมูลของบริษัท

RECOMMENDATION Asset search จะเข้าข่ายการทำงานของ Enterprise Search เนื่องจากจะเป็นการค้นหาโดยการใช้งานข้อมูลของ outsource ซึ่งถือว่าเป็นหนึ่งในข้อมูลของทางบริษัท

ซึ่งในการค้นหาที่ผู้จัดทำได้ออกแบบมานั้น คือการค้นหาด้วยชื่อทรัพย์สินได้โดยตรงเลย แต่ปัญหาของการค้นหาด้วยชื่อทรัพย์สินคือปัญหาทางด้านการใช้ภาษา เนื่องจากภาษามนุษย์มีความยืดหยุ่นทางภาษา รวมไปถึงความผิดพลาดของมนุษย์ (human error) ที่สามารถเกิดขึ้นได้ ซึ่งแตกต่างจากภาษาคอมพิวเตอร์ที่เป็นภาษาที่มีความแม่นยำและเสถียร ผู้จัดทำจึงจำเป็นที่จะต้องทำให้คอมพิวเตอร์เข้าใจภาษาของมนุษย์ผ่านการทำงานของ Natural Language Processing (NLP)

โดยงาน NLP นั้นมีทางหลากหลายด้านมากมายที่น่าสนใจ ซึ่งงาน NLP ที่นำมาประยุกต์ใช้ใน RECOMMENDATION Asset search นั้นจะประกอบไปด้วยสองส่วนใหญ่ๆ ได้แก่ Tokenization และ Text comparison

3.2.4.1 Tokenization

การทำ Tokenization เป็นงานที่น่าสนใจในภาษาไทย เนื่องจากภาษาไทยเป็นภาษาที่ไม่มี stop word ไม่ชัดเจนเหมือนภาษาอังกฤษที่มีการใช้ stop word ที่ชัดเจนด้วยการใช้ช่องว่าง (white space) ดังนั้นภาษาไทยจึงต้องมีรูปแบบการทำที่แตกต่างออกไปจากภาษาสากลอย่างเช่นภาษาอังกฤษอย่างเห็นได้ชัด

ซึ่งผู้จัดทำได้ทำการทดลองด้วยวิธีการ 3 รูปแบบด้วยกันได้แก่ pythainlp, attacut, sertis model

PyThaiNLP เป็นหนึ่งใน library ของ python ที่เข้ามาช่วยในการจัดการการทำงานทางด้าน NLP สำหรับภาษาไทยโดยเฉพาะ ซึ่ง PyThaiNLP มีฟังก์ชันที่เกี่ยวข้องกับการทำงาน NLP มากมาย อาทิ เช่น การทำ tokenize, การเช็คคำผิด, การอ่านออกเสียง และฟังก์ชันอื่นๆอีกมากมาย

โดยในการทดลองของผู้จัดทำในการทำ Tokenization กับ library ของ PyThaiNLP ซึ่งมีฟังก์ชัน word_tokenize ที่จะสามารถทำการ Tokenize ให้กับ text ที่ป้อนเข้าไปได้ ซึ่ง word_tokenize เองก็จะมี parameter ที่ชื่อว่า engine ที่จะเป็นเหมือนการเลือกทฤฎีในการตัดคำ ซึ่งการใช้ engine แต่ละ engine ก็จะได้ผลลัพธ์ที่มีความแตกต่างกันไป

Attacut เป็นอีกหนึ่งใน library ในการทำงานทางด้าน NLP ของภาษาไทย และเป็นอีกหนึ่งใน engine การทำงานของ word_tokenize ของ PyThaiNLP

Thai Word Segmentation เป็น model สำหรับการสร้างการตัดคำภาษาไทย (Thai word segmentation) ที่ถูกพัฒนาโดย Sertis Co.,Ltd ซึ่งตัว model การตัดคำภาษาไทยนี้ถูกพัฒนาต่อยอดมาจาก Artificial Neural Network (ANN) ซึ่งพัฒนากลายมาเป็น model ที่มี layers ที่จะประกอบไปด้วย 4 layers ด้วยกันได้แก่

1.Input Layers

Input layers ในชั้นนี้นั้นคือการป้อนข้อมูลเข้าสู่ Bi-RNN ซึ่งก่อนที่จะป้อนข้อมูลเข้าสู่กระบวนการนี้ได้นั้นจำเป็นที่จะต้องทำการเตรียมความพร้อมของข้อมูลเสียก่อน (Preprocessing Data)

Preprocessing Data คือการเตรียมความพร้อมข้อมูลก่อนที่จะทำการป้อนข้อมูลเพื่อเป็น input เข้าสู่โครงข่ายการทำงาน โดยก่อนที่จะทำ Preprocessing Data เราจำเป็นที่จะต้องเข้าใจปัญหาการทำ Tokenization ในภาษาไทยเสียก่อน เพื่อให้สามารถเตรียมความพร้อมให้เหมาะสมกับโครงข่ายที่จะใช้งาน

เข้าใจการทำ Tokenization ของภาษาไทย

ดังที่เขียนไว้ข้างต้น ภาษาไทยนั้นไม่สามารถแบ่งคำด้วยเทคนิคการหา Stop word ได้อย่างชัดเจน แต่ภาษาไทยนั้นจะตัดคำออกจากประโยคได้นั้น เราต้องรู้ว่า word ที่เราต้องการจะทำการ Tokenize นั้นคืออะไร ซึ่งองค์ประกอบของคำในภาษาไทยนั้นสามารถมองเป็นอักขระเริ่มต้น และอักขระสิ้นสุดของคำได้ดังตัวอย่างต่อไปนี้

[“สวัสดี”] จะแบ่งออกได้เป็น [‘ส’, ‘ว’, ‘ไม้หันอากาศ’, ‘ส’, ‘ด’, ‘สระอี’ ,] โดยที่ ‘ส’ ที่ตำแหน่งแรกจะเป็นอักจระเริ่มต้น ส่วน ‘สระอี’ จะเป็นอักขระสิ้นสุดของคำ และเมื่อเราแทนอักขระเริ่มต้นด้วย 1 และอักขระสิ้นสุด และอักขระระหว่างอักขระเริ่มต้นและอักขระสิ้นสุดจะแทนค่าด้วย 0 เราจะได้ค่าดังนี้

[“สวัสดี”] == [100000]

ซึ่งจากการใช้เทคนิคนี้ จะสามารถเข้ามาแก้ปัญหา Tokenization ได้ในกรณีคำนั้นเป็นคำพ้องรูป โดยถ้าอ้างอิงจากตัวอย่างข้างต้นจะมีการตัดคำดังนี้

[“เพลา”] == [“เพ”, “ลา”] == [1010]

[“เพลา”] == [“เพลา”] == [1000]

หลังจากที่เราสามารถเปลี่ยนข้อมูลที่เป็นอักขระให้เป็นตัวเลขได้แล้ว อีกสิ่งหนึ่งที่เราต้องทำคือการสร้าง dictionary ที่สามารถจดจำอักขระให้เป็นตัวเลขได้ อาทิเช่น ส = 15 ว = 20 ไม้หันอากาศ =25 ด =30 สระอี = 45 ดังนั้นคำว่าสวัสดีจะมีค่าเป็น [15,20,25,15,30,45]

Batching

การทำ Batching คือการจับกลุ่มข้อมูล เพื่อแบ่งข้อมูลออกเป็นชุดๆ โดย Batch Size ในที่นี้จะยึดจาก word ที่มีอักระมากสุดในประโยคนั้นๆ และหาก word ที่อยู่ในประโยคเดียวกันมี size ที่เล็กกว่าให้อาศัยการเติม padding เข้าไป

ตัวอย่างเช่น [“สวัสดีครับผมชื่อกิตติทัต”] จะแบ่งออกมาได้ ดังนี้

[“สวัสดี”] = size:6char+2padding = [100000,pad,pad]

[“ครับ”] = size:4 char+4padding= [1000,pad,pad,pad,pad]

[“ผม”] = size:2 char+6padding= [10,pad,pad,pad,pad,pad,pad]

[“ชื่อ”] = size:4 char+4padding = [1000,pad,pad,pad,pad]

[“กิตติทัต”] = size: 8 char(max batch size)= [10000000]

โดยการเตรียมข้อมูลก่อนที่จะเข้า model สำหรับการ train นั้น จะแบ่งข้อมูลออกเป็น 90% สำหรับ train set และ 10% สำหรับ validate set ซึ่งในแต่ละ batch เองก็จะประกอบไปด้วยหลายประโยค

ซึ่งเมื่อเตรียมความพร้อมเสร็จสิ้นแล้ว เราจะได้ข้อมูลออกมาเป็น batch แต่ละชุด ที่จะสามารถป้อนเข้าสู่ขั้นตอนต่อไปในการทำงานของโครงข่ายการทำงานได้

2.Character Embedding

ในขั้นตอนต่อมาหลังจากเตรียมความพร้อมข้อมูลเสร็จสิ้นและได้แบ่งชุดข้อมูลออกมาเป็นแต่ละ batch เพื่อที่จะไปสู่ layers ถัดไปคือ Character Embedding ที่จะแปลงข้อมูลจาก label เป็น vector เพื่อที่จะสามารถนำไป process ต่อได้ในอนาคต

3.Recurrent Layer

มีการใช้ Bi-RNN มาประยุกต์ใช้ เพื่อให้สามารถครอบคลุมเนื้อหาทั้ง forward และ backward และเอาผลลัพธ์ที่ได้จากฝั่ง forward และฝั่ง backward มารวมกัน เพื่อที่จะส่งไปยัง layers ต่อไป

4.Output layer

หลังจากได้ผลลัพธ์จากการคำนวนจาก LSTM unit แล้วจึงจะนำผลลัพธ์นั้นเข้าสู่ activation function เพื่อที่จะตัดสินใจว่าคำๆนั้นควรจะมี token เป็นอย่างไร

3.2.4.2 Text Comparison

Text Comparison ใน RECOMMENDATION Asset search นั้นเป็นงานที่สำคัญมาก เช่นเดียวกันกับ Tokenization เพราะเป็นหัวใจหลักในการค้นหาชื่อสินทรัพย์แทนการค้นหาด้วย ID เพราะ Text comparison ใน RECOMMENDATION Asset search เพราะจะเอาเทคนิคนี้มาทำการเปรียบเทียบคำที่ User ป้อนเข้ามา เพื่อเปรียบเทียบกับข้อมูลในฐานข้อมูล โดยผู้จัดทำได้ทำการทดลองผ่านกระบวนการทั้งหมด 3 วิธีการด้วยกัน ได้แก่ Textdistance, PyThaiNLP, TFIDF+Bigram

Textdistance เป็นหนึ่งใน library ของ python ที่มีฟังก์ชันของการหาความแตกต่างของคำศัพท์ โดยมีฟังก์ชันหลากหลายสมการ ซึ่งในโครงงานนี้มีการใช้ฟังก์ชันจาก library ทั้งหมด 4 ฟังก์ชันด้วยกันได้แก่ Leveshtien distance, Hamming distance, Cosine similarity, Jaccard Index

PyThaiNLP มีฟังก์ชันการแก้ไขคำผิด (ที่คาดว่าเกิดจากการพิมพ์ผิด, สะกดผิด) นั้นคือฟังก์ชัน spell ที่จะ return ค่าออกมาเป็นชุดคำตอบที่เป็นไปได้ 10 อันดับ

TFIDF+Bigram การเปรียบเทียบคำโดยวิธีนี้นั้นประกอบไปด้วยวิธีการ 2 วิธีการเข้าด้วนกัน นั้นคือ TF-IDF และ Bigram

TF-IDF คือหนึ่งในการ weight น้ำหนักของคำแต่ละคำใน text

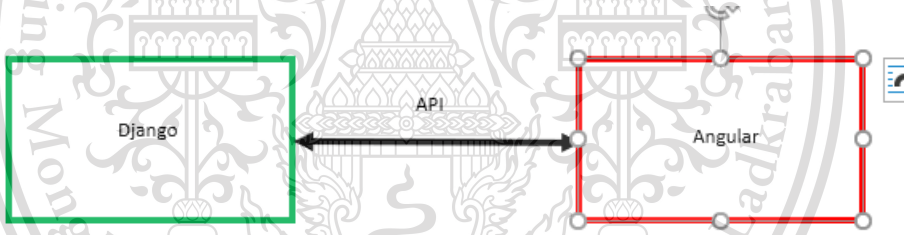
Bigram คือหนึ่งในรูปแบบของการจัดเก็บคำศัพท์

โดยการสร้าง TF-IDF นั้นสามารถทำได้ง่ายๆ ด้วยการเรียกใช้งานผ่าน scikit-learn ซึ่งเป็นหนึ่งใน library ของ python ที่มีความสามารถในการแปลง text ให้เป็น TF-IDF vector ได้

3.2.5 Data Visualization

Data Visualization คือการแสดงผลลัพธ์ออกมาเพื่อถ่ายทอดข้อมูลออกมาให้ได้อย่างถูกต้องและกระชับ เพื่อที่ผู้ใช้งาน (User) สามารถเข้าใจถึงตัวข้อมูลได้อย่างรวดเร็วและแม่นยำ

โดยใน RECOMMENDATION Asset search จะทำการแสดงผล (visualize) ผ่านเว็ปไซต์ (web-based visualization) ซึ่งการพัฒนาฝั่ง front-end นั้นจะใช้ anugular ในการแสดงผล โดยในการออกแบบนั้น ผู้จัดทำได้รงบันดาลใจมาจาก google ซึ่งเป็น search engine ยอดนิยม และมี interface ที่ผู้ใช้งานคุ้นชินแล้ว จึงเป็นสาเหตุที่ว่า UI ของโครงงานนี้จึงคล้ายคลึงกับ google นั้นเอง



รูปที่3.5 การติดต่อระหว่าง 2 Framework เพื่อการแสดงผลบน web app

3.3 กระบวนการดำเนินงานการพัฒนา

ในการนำการออกแบบ Data pipeline มาประยุกต์เพื่อให้สามารถใช้งาน (Implementation) ได้ ซึ่งการทำงานส่วนใหญ่จะทำอยู่บน laptop ของผู้จัดทำ โดยการทำงานเพื่อที่จะแปลง Data pipeline ออกมาเป็นในรูปแบบของ coding โดยส่วนมากโครงการรี้จะถูกพัฒนาบนภาษา python และการทำงานเพื่อทดลองและพัฒนาระบบจะถูกทำบนระบบปฏิบัติการ(Operation system) window 10 และ Ubuntu 18.04 LTS ซึ่งเป็น software จำลองการทำงานของระบบปฏิบัติการ Ubuntu

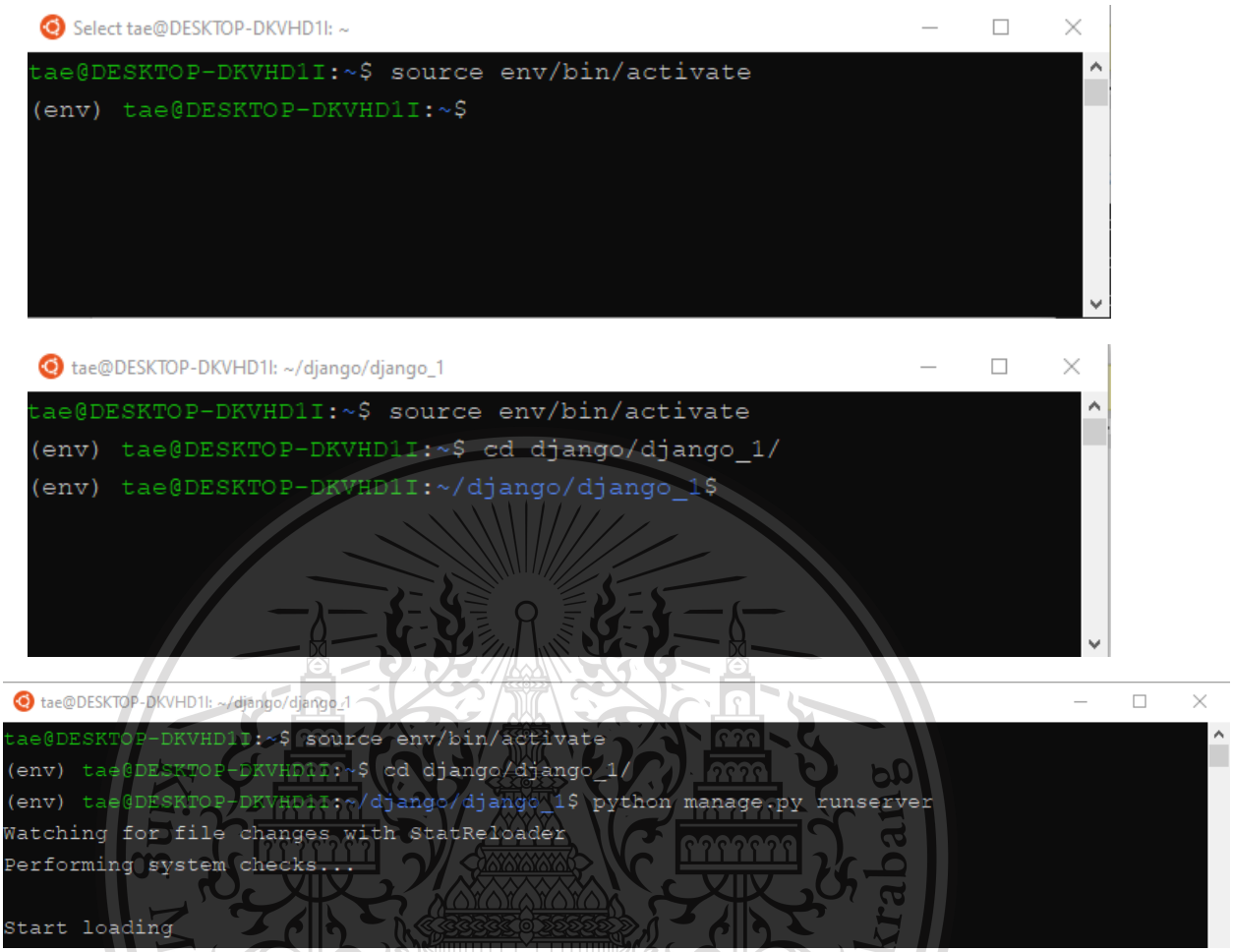
การทดลองบางส่วนนั้นจำเป็นที่จะต้องทำงานบนระบบปฏิบัติการ Ubuntu เนื่องจาก Ubuntu เป็นหนึ่งในระบบพัฒนามาจากระบบปฏิบัติการ Linux ซึ่งระบบปฏิบัติการ Linux จะมีความยืดหยุ่นในการลง library ต่างๆ มากกว่า window จึงทำให้การทดลองต้องทดสอบอยู่บนระบบปฏิบัติการนี้

ซึ่งการพัฒนาการเขียนโปรแกรมนั้นจะถูกเขียนอยู่บน Visual Studio code ซึ่งเป็น Integrated development environment (IDE) ที่ใช้งานได้ไม่ซับซ้อน และได้รับความยอมรับและถูกใช้งานโดยเหล่า developer ทั่วโลก

Data source ในทางปฏิบัตินั้นไม่สามารถนำข้อมูลจาก outsource ซึ่งเป็นข้อมูลของทางบริษัทอันมีความลับของลูกค้า ผู้จัดทำจึงจำเป็นที่จะต้องหาชุดข้อมูลอื่นมาเพื่อทำการทดแทน ซึ่งผู้จัดทำได้ทำการ web scraping ซึ่งเป็นการเก็บข้อมูลมาจากเว็บไซต์ โดยเว็บไซต์ที่ผู้จัดทำได้ยกมาใช้งานคือ zmyhome โดยมีขอบเขตการดึงข้อมูลคือข้อมูลประเภทคอนโด ในจังหวัดกรุงเทพมหานคร

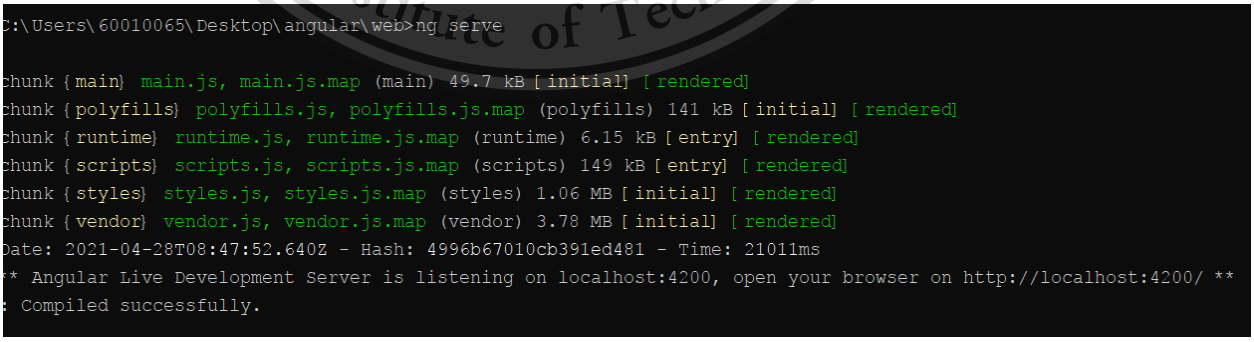
3.4 กระบบวนการใช้งาน

การใช้งาน RECOMMENDATION Assset search นั้นจำเป็นที่จะต้องดำเนินการเปิดเซิฟเวอร์จากทางฝั่งของ back end และ front end เพื่อใช้งาน โดยในฝั่งของ back end นั้นจำเป็นที่จะต้องเข้าไปใน Ubuntu 18.04 LTS จากนั้นต้องทำการ activate virtual environment ที่สร้างมาเพื่อใช้งานกับ RECOMMENDATION Asset search โดยให้สั่งการด้วยคำสั่ง *source env\bin\activate* จากนั้นเข้าไปที่ django_1/django_1 ซึ่งจะมีไฟล์ที่มีชื่อว่า manage.py จากนั้นจึงสั่งการด้วยคำสั่ง *python manage.py runserver*



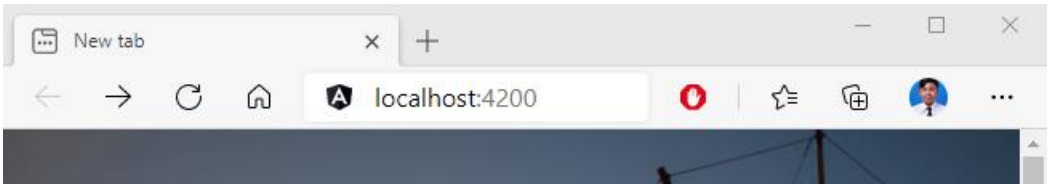
รูปที่3.6 การสั่งงานเพื่อใช้งานส่วนของ Back-end

ในส่วนของฝั่ง front end นั้น ให้เข้าไปที่ angular\web จากนั้นจึงสั่งการด้วยคำสั่ง ng serve เพื่อเปิดใช้งาน



รูปที่3.7 การสั่งงานเพื่อใช้งานส่วนของ Front-end

เมื่อเปิดการทำงานแล้ว เราสามารถเข้าไปที่ browser และทำการป้อน http://localhost:4200/ เพื่อเริ่มใช้งาน

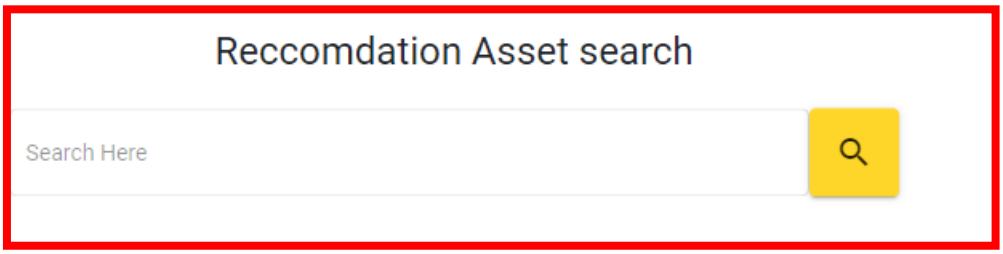


รูปที่3.8 browse ไปยัง port ที่กำหนดเพื่อเข้าใช้งาน

โดยผลลัพธ์ของ website เราจะเป็นดังนี้

รูปที่3.9 หน้าแสดงผลหลัก (Home Page)

เมื่อเข้ามาที่หน้าเริ่มต้น (Home Page) จะสามารถเห็นได้ชัดว่าจะมีช่องให้กรอกชื่อสินทรัพย์ที่ต้องการที่จะค้นหา

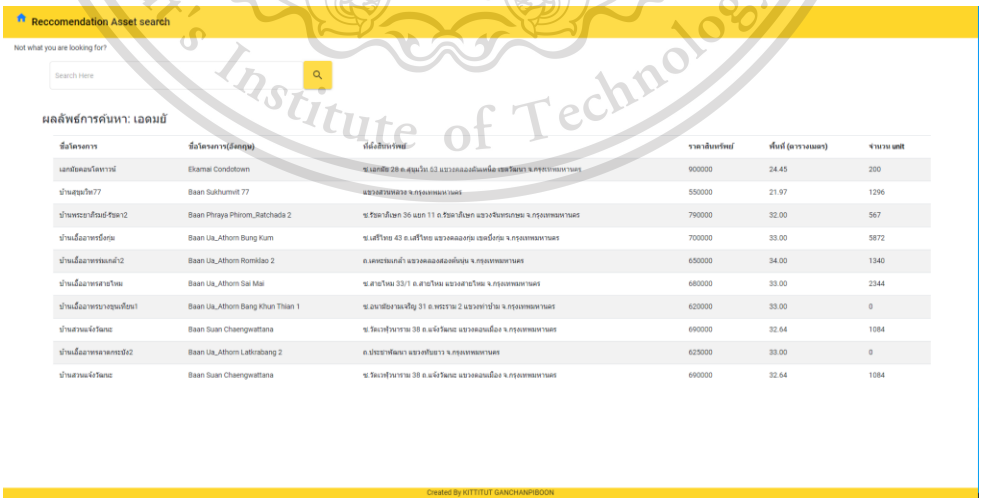


รูปที่3.10 ช่องกรอกข้อมูลที่ต้องการค้นหา

ซึ่ง textbox ตรงนี้จะเป็นช่องสำหรับกรอกชื่อสินทรัพย์ที่ต้องการจะค้นหา ซึ่งเมื่อกรอกเสร็จสิ้นแล้วสามารถที่จะกดปุ่มแว่นขยาย เพื่อทำการค้นหาสินทรัพย์ได้

รูปที่3.11 ตัวอย่างการกรอกข้อมูลที่ผิดพลาดจากการพิมพ์ผิด

โดยในตัวอย่างการใช้งานนี้ ผู้จัดทำจงใจที่จะกรอกข้อมูลผิด จาก “เอกมัย” เป็น “เอดมยี” เพื่อรอดูผลลัพธ์ว่าจะตรงตามการค้นหาที่ต้องการหรือไหม ซึ่งรูปข้างล่างคือผลลัพธ์หลังจากการกดค้นหา



รูปที่3.12 หน้าแสดงผลลัพธ์

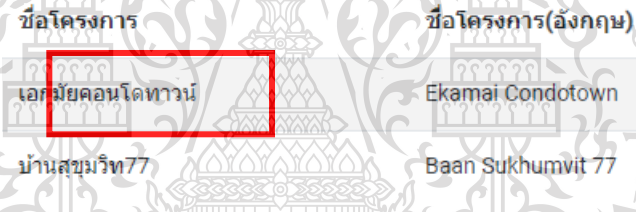
โดยผลลัพธ์จากการค้นหานั้นจะมีทั้งหมด 10 ข้อมูลที่มีความใกล้เคียงกับสิ่งที่เราทำการกรอกข้อมูลลงไป

ผลลัพธ์การค้นหา: เอดมย้

ชื่อโครงการ	ชื่อโครงการ(อังกฤษ)	ที่ตั้งสินทรัพย์	ราคาสินทรัพย์	พื้นที่ (ตารางเมตร)	จำนวน unit
เอกมัยคอนโดทาวน์	Ekamai Condotown	ซ.เอกมัย 28 ถ.สุขุมวิท 63 แขวงคลองตันเหนือ เขตวัฒนา จ.กรุงเทพมหานคร	900000	24.45	200
บ้านสุขุมวิท77	Baan Sukhumvit 77	แขวงสวนหลวง จ.กรุงเทพมหานคร	550000	21.97	1296
บ้านพระยาภิรมย์-รัชดา2	Baan Phraya Phirom_Ratchada 2	ซ.รัชดาภิเษก 36 แยก 11 ถ.รัชดาภิเษก แขวงจันทรเกษม จ.กรุงเทพมหานคร	790000	32.00	567
บ้านเอื้ออาทรบึงกุ่ม	Baan Ua_Athorn Bung Kum	ซ.เสรีไทย 43 ถ.เสรีไทย แขวงคลองกุ่ม เขตบึงกุ่ม จ.กรุงเทพมหานคร	700000	33.00	5872
บ้านเอื้ออาทรร่มเกล้า2	Baan Ua_Athorn Romklao 2	ถ.เคหะร่มเกล้า แขวงคลองสองต้นนุ่น จ.กรุงเทพมหานคร	650000	34.00	1340
บ้านเอื้ออาทรสายไหม	Baan Ua_Athorn Sai Mai	ซ.สายไหม 33/1 ถ.สายไหม แขวงสายไหม จ.กรุงเทพมหานคร	680000	33.00	2344
บ้านเอื้ออาทรบางขุนเทียน1	Baan Ua_Athorn Bang Khun Thian 1	ซ.อนามัยงามเจริญ 31 ถ.พระราม 2 แขวงท่าข้าม จ.กรุงเทพมหานคร	620000	33.00	0
บ้านสวนแจ้งวัฒนะ	Baan Suan Chaengwattana	ซ.วัดเวฬุวนาราม 38 ถ.แจ้งวัฒนะ แขวงตลาดเมือง จ.กรุงเทพมหานคร	690000	32.64	1084
บ้านเอื้ออาทรลาดกระบัง2	Baan Ua_Athorn Latkrabang 2	ถ.ประชาพัฒนา แขวงทับยาว จ.กรุงเทพมหานคร	625000	33.00	0
บ้านสวนแจ้งวัฒนะ	Baan Suan Chaengwattana	ซ.วัดเวฬุวนาราม 38 ถ.แจ้งวัฒนะ แขวงตลาดเมือง จ.กรุงเทพมหานคร	690000	32.64	1084

รูปที่3.13 ผลลัพธ์ที่ได้จากการค้นหาจากรูป 3.8

โดยจากผลลัพธ์ จะเห็นได้ว่าผลลัพธ์ที่ 1 มีชื่อสอดคล้องกับสิ่งที่ผู้จัดทำต้องการจะค้นหา



ชื่อโครงการ	ชื่อโครงการ(อังกฤษ)
เอกมัยคอนโดทาวน์	Ekamai Condotown
บ้านสุขุมวิท77	Baan Sukhumvit 77

รูปที่3.14 ผลลัพธ์ที่ได้ ที่สอดคล้องกับการค้นหา

และถ้าหากผู้ใช้งานยังต้องที่จะค้นหาสินทรัพย์อื่นๆ สามารถที่จะกรอกข้อมูลที่ช่องกรอกข้อมูลทางด้านซ้ายบนได้

รูปที่3.15 ช่องค้นหาเพิ่มเติม

หรือถ้าหากต้องการที่จะกลับไปที่หน้า Home Page ผู้ใช้งานสามารถที่จะกด Icon รูปบ้านทางด้านซ้ายบนได้

รูปที่3.16 นำทางกลับไปหน้าหลัก

บทที่ 4

ผลการดำเนินงาน

4.1 ผลการทดลอง tokenization

ในโครงงานนี้ผู้จัดทำได้ทำการทดลองการทำ tokenization ด้วยเทคนิค 3 รูปแบบด้วยกัน ได้แก่ pythainlp.tokenize(), Attacut() และ Sertis Model

ทั้ง 3 เทคนิคนี้ มีข้อดีและข้อด้อยแตกต่างกันไป โดยที่ pythainlp.tokenize() และ Attacut() นั้นเป็น model สำหรับการทำ tokenize ในกรณีทั่วไป เนื่องจากเป็นโมเดลที่ได้รับการ trained มาก่อนหน้า(transfer learning) ด้วย corpus ที่มีขนาดใหญ่ แต่อย่างไรก็ตาม เมื่อเปรียบเทียบกับ Thai-word-segmentation ซึ่งเป็นโมเดลที่เป็น model ที่ต้องอาศัยการ train จากเอกสารที่สามารถเลือก input ข้อมูลได้เอง กลับพบว่า Thai-word-segmentation นั้นมีความสามารถในการทำ tokenize ได้เหมาะสมกว่า model ทั้งสองข้างต้น โดยการทดสอบนี้จะทดสอบด้วย Extrinsic evaluation ซึ่งเป็นการทดสอบจากการใช้งาน เพื่อหาความสอดคล้องของการ tokenize กับความต้องการของ user

โดยรูปข้างล่างนี้จะเป็นผลการดำเนินงานของการทำ tokenize ด้วยเทคนิคข้างต้น

```
from pythainlp.tokenize import word_tokenize

text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
print('original word :',text)
token = word_tokenize(text)
print('tokenization list :',token)
```

```
original word : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
tokenization list : ['เดอะ', 'เฟิร์ส', 'โฮม', ' ', 'วงแหวน', ' ', 'ลำ', 'ลูก', 'กา', ' ', 'คลอง', ' ', '3']
```

รูปที่4.1 ผลลัพธ์ของ pythainlp.tokenize

```
from attacut import tokenize
text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
print('original word :',text)
token = tokenize(text)
print('tokenization list :',token)
```

```
original word : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
tokenization list : ['เดอะ', 'เฟิร์ส', 'โฮม', ' ', 'วงแหวน', ' ', 'ลำลูกกา', ' ', 'คลอง', ' ', '3']
```

รูปที่4.2 ผลลัพธ์ของ Attaccut()

```
texts = ['เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3']
```

```
[ 'เดอะเฟ ิ ร์ สโฮม ', 'วง', 'แหวน ', 'ลำลู กกา', ' คลอง', ' 3']
```

รูปที่4.3 ผลลัพธ์ของ Sertis Model (Thai word Segmentation)

4.2 ผลการทดลอง textdistance

รูปแบบของ textdistance นั้นมีหลากหลายรูปแบบด้วยกัน ซึ่งในโครงงานนี้ ผู้จัดทำได้หยิบยกการทำ textdistance มาด้วยกันทั้งหมด 2 รูปแบบนั้นคือ Edit base similarity (Hamming distance และ Levenshtein distance) และ Tokenize base similarity (Cosine similarity และ Jaccard's similarity)

```
import textdistance


lst = ['เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3','เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง','เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3']


for i in lst:
    text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
    distance = textdistance.levenshtein(text,i)
    tmp = (distance,i)
    print('ข้อมูลที่ถูกต้อง :',text)
    print('ข้อมูลเปรียบเทียบ :',i)
    print('distance :',distance)
    print('**************************')
```

```
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3
distance : 12
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง
distance : 5
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3
distance : 5
**************************
```

รูปที่4.4 ผลลัพธ์ของ Levenshtein distance

```python
import textdistance


lst = ['เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3','เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง','เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3']


for i in lst:
    text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
    distance = textdistance.hamming(text,i)
    tmp = (distance,i)
    print('ข้อมูลที่ถูกต้อง :',text)
    print('ข้อมูลเปรียบเทียบ :',i)
    print('distance :',distance)
    print('**************************')
```

```
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3
distance : 20
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง
distance : 5
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3
distance : 27
**************************
```

รูปที่4.5 ผลลัพธ์ของ Hamming distance

```python
import textdistance


lst = ['เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3','เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง','เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3']


for i in lst:
    text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
    distance = textdistance.cosine(text,i)
    tmp = (distance,i)
    print('ข้อมูลที่ถูกต้อง :',text)
    print('ข้อมูลเปรียบเทียบ :',i)
    print('distance :',distance)
    print('**************************')
```

```
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3
distance : 0.7919455160226819
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง
distance : 0.8892320230027655
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3
distance : 0.9048740202266848
**************************
```

รูปที่4.6 ผลลัพธ์ของ Cosine Similarity

```
import textdistance


lst = ['เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3','เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง','เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3']


for i in lst:
    text = 'เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3'
    distance = textdistance.jaccard(text,i)
    tmp = (distance,i)
    print('ข้อมูลที่ถูกต้อง :',text)
    print('ข้อมูลเปรียบเทียบ :',i)
    print('distance :',distance)
    print('**************************')
```

```
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน พุทธมณฑลสาย 2 คลอง 3
distance : 0.6521739130434783
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลองขวาง
distance : 0.8
**************************
ข้อมูลที่ถูกต้อง : เดอะเฟิร์สโฮม วงแหวน ลำลูกกา คลอง 3
ข้อมูลเปรียบเทียบ : เดอะเฟิร์สเฮ้าส์ วงแหวน ลำลูกกา คลอง 3
distance : 0.825
**************************
```

รูปที่4.7 ผลลัพธ์ของ Jaccard’s Index

โดยเมื่ออ้างอิงจากการทดลองและการสืบค้นแล้ว จึงทำให้ทราบว่า edit base similarity นั้นจะเป็นการเปรียบเทียบข้อมูลด้วยความต่างของอักขระ (atomic understanding) ซึ่งเป็นการหาความแกต่างที่ไม่สามารถสื่อถึงสารในข้อมูลได้ ในขณะที่ Tokenize base similarity จะเป็นการหาความแตกต่างของเนื้อหาภายในของข้อมูลได้ (Semantic meaning) โดยที่การทำ textdistance ด้วย Semantic meaning นั้นสามารถนำไปประยุกต์ใช้งานกับ word vector ที่จะมอง document ของเราเป็น metrics

ซึ่งการมองเป็น metrics นั้นมีข้อได้เปรียบที่เราสามารถที่จะนำข้อมูลของเราไปค้นหาความคล้ายคลึงของ document ใน model สำหรับการค้นหาได้

ในส่วนของ Jaccard similarity และ Cosine Similarity นั้นก็มีข้อดีข้อเสียต่างกันไป โดยที่ Jaccard similarity จะเหมาะสมกับกรณีที่ข้อมูลซ้ำ (duplicate) ไม่มีผลในการค้นหา และ Cosine similarity จะเหมาะสมกับข้อมูลซ้ำมีผลต่อการค้นหา ซึ่งทางผู้จัดทำเลือกที่จะใช้ Cosine Similarity เนื่องจากเป็นวิธีที่สอดคล้องกับการค้นหาข้อมูลที่ผู้จัดทำต้องการ

4.3 ผลการทดลองการค้นหา

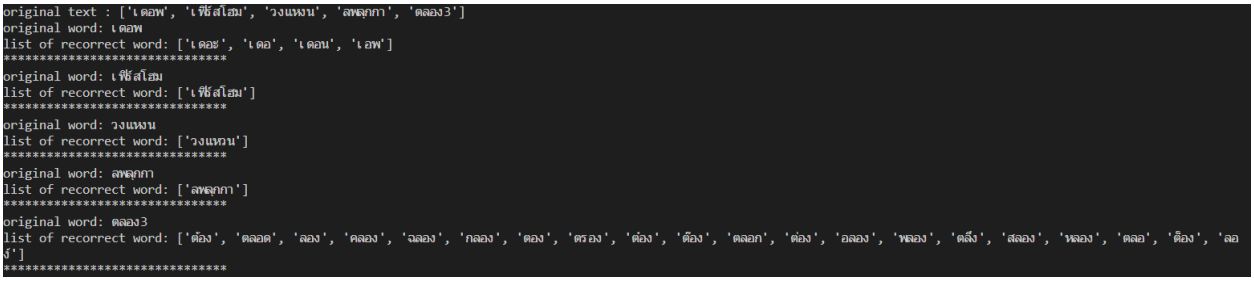
จากสองการทดลองข้างต้น ทำให้ผู้จัดทำเลือกที่จะใช้เทคโนโลยีของ Thai-word-segmentation ในการทำ tokenization และเลือกใช้ Cosine similarity ในการเปรียบเทียบความคล้ายคลึงของข้อมูล แต่ว่าการทดลองยังไม่จบเพียงเท่านั้น ผู้จัดทำยังเลือกที่จะหาวิธีสืบค้นข้อมูลแบบอื่นมา เพื่อให้การค้นหามีประสิทธิภาพมากยิ่งขึ้น

“การค้นหา” อาจจะสามารถสื่อได้ถึงการแปลงข้อมูลที่คิดว่าใกล้เคียงที่สุดในกรณีที่ไม่พบ keyword ที่ต้องการค้นหา ผู้จัดทำจึงทดลองใช้ pythainlp.spell() ที่เป็น function ในการแก้ไขคำผิด ซึ่งจะแก้ไขคำผิดนั้นให้เป็นคำที่ใกล้เคียงที่สุด ซึ่งวิธีนี้เป็นวิธีที่ดีสำหรับข้อมูลเรื่องทั่วไป

แต่หากเพิ่มความสามารถให้กับการค้นหาของเราได้มีประสิทธิภาพมากยิ่งขึ้น จึงจำเป็นที่จะสร้างขอบเขตการค้นหาของตัวเองขึ้นมา โดยผู้จัดทำเลือกที่จะนำการทำ TF-IDF มาผสมผสานกับ Bigram เนื่องจาก TF-IDF นั้นทำงานสอดคล้องกับ Semantic meaning ซึ่งจะกรองเฉพาะคำศัพท์ที่สำคัญในประโยคที่ต้องการจะค้นหา และ Bigram จะมีหน้าที่ในการขยายขอบขตของการค้นหามากยิ่งขึ้น เพื่อที่จะทำให้การค้นหานั้นคลอบคลุมยิ่งกว่าเดิม

```
from pythainlp import spell

text = ['เดอพ','เฟิร์สโฮม','วงแหงน','ลพลุกกา','ตลอง3']
print('original text :',text)
for i in text:
    token = spell(i)
    print('original word:',i)
    print('list of recorrect word:',token)
    print('******************************')
```



รูปที่4.8 ผลลัพธ์ของ pythainlp.spell()

```
def main_search(string):
    res = []
    query_words_kum = cut_kum(string,sub_space=False)
    print("input term",string)
    print("tokenize ", query_words_kum)
    a, pair_dist, query_words_kum_out = tfidf_search(
        tfidf, query_words_kum, tfidf_features)

    print("possibly search results ", query_words_kum_out)
    for index in a:
        most_sim = "".join(Raw_data[index])
        print(most_sim)
        res.append(most_sim)


    return res,query_words_kum_out
```

```
input term เฟิ ร์ ส
tokenize  ['เฟิ ร์ ส']
word not in corpus  เฟิ ร์ ส
recorrect word would be  เฟิ ร์ สโฮม
possibly search results  ['เฟิ ร์ สโฮม']
เดอะเฟิ ร์ สโฮม วงแหวน ลำลู กกา คลอง 3
เดอะเฟิ ร์ สโฮม วงแหวน ลำลู กกา คลอง 3
เดอะเฟิ ร์ สโฮม วงแหวน ลำลู กกา คลอง 3
```

รูปที่4. 9 ผลลัพธ์ของ Bigram + TFIDF

บทที่ 5

สรุปผลและการวิจารณ์

5.1 สรุปผลการดำเนินงาน

RECOMMENDATION Asset Search นั้นถูกออกแบบมาเพื่อให้พนักงานทีม mortgage สามารถค้นหาสินทรัพย์ได้ว่องไวมากยิ่งขึ้น เพื่อลดเวลาการทำงานในส่วนของการค้นหาข้อมูล ซึ่ง RECOMMENDATION Asset search จะถูกพัฒนาด้วย Django rest framework ในส่วนของ Back-end และถูกพัฒนาด้วย Angular ในฝั่งของ Front-end ถึงแม้การค้นหาข้อมูลนั้นอาจจะดูผิวเพินเหมือนเป็นการ query ข้อมูลจาก mongoDB ซึ่งเป็นฐานข้อมูลหลักในโครงานนี้ แต่เบื้องลึกนั้นได้มีการนำเทคโนโลยีต่างๆเข้ามาใช้งาน เพื่อเพิ่มประสิทธิภาพของการค้นหา เพื่อให้สามารถทำการค้นหาสินทรัพย์จากชื่อโครงการของสินทรัพย์นั้นได้ ซึ่งจะสามารถลดเวลาในการค้นหาสินทรัพย์ลง โดยการค้นหานั้นจะประกอบไปด้วยเทคโนโลยีดังต่อไปนี้

Thai-word-segmentation ที่จะทำหน้าที่การตัดคำ (Word Segmentation) ซึ่งเบื้องหลังของ model นี้นั้นก็คือ Bi-Recurrent Neural Network ที่เป็นหนึ่งใน Deep Learning ตัวหนึ่งที่มีประสิทธิภาพในการจดจำข้อมูลเพื่อเป็นปัจจัยในการ predict

Cosine Similarity ที่จะทำหน้าที่เปรียบเทียบความคล้ายคลึงกันของข้อมูล ว่ามีความคล้ายคลึงมากมายเพียงใด และการเปรียบเทียบนี้นั้นยังจำเป็นที่จะต้องอาศัยการทำ tokenize/word segmentation เพื่อที่จะดึงประสิทธิภาพออกมาให้ได้ดีมากที่สุด

TF-IDF และ Bigram สองเทคนิคที่จะเข้ามาช่วยเพิ่มประสิทธิภาพการทำงานให้มีประสิทธิภาพการค้นหาที่มีขอบเขตมากยิ่งขึ้น

5.2 ปัญหาที่เกิดขึ้น และแนวทางแก้ไขปัญหา

5.2.1 ปัญหาเรื่องชุดข้อมูล

เนื่องจากข้อมูลสำหรับใช้อ้างอิงในการค้นหานั้น แรกเริ่มเดิมทีนั้นเป็นข้อมูลของบริษัทธนาคารพาณิชย์จึงทำให้ไม่สามารถนำข้อมูลเหล่านั้นออกมาแสดงได้ เนื่องจากอาจจะเป็นการไปละเมิดข้อมูลของทางลูกค้าของทาง บริษัท ซึ่งอาจจะนำมาซึ่งการเสื่อมเสียชื่อเสียงได้

-แนวทางการแก้ไข เมื่อผู้จัดทำได้รับทราบปัญหาในข้อนี้แล้ว ผู้จัดทำจึงต้องหาแหล่งข้อมูลใหม่ ซึ่งผู้จัดทำเลือกที่จะใช้ข้อมูล สินทรัพย์ประเภท condo ในจังหวัดกรุงเทพ ของ www.zmyhome.com แทน

5.2.2 จำนวนของข้อมูล

การที่จะทำให้การสืบค้นมีประสิทธิภาพมากยิ่งขึ้นนั้นจำเป็นที่จะต้องมีชุดข้อมูลจำนวนมาก ซึ่งยิ่งจำนวนมากเท่าไร ยิ่งส่งผลดีต่อการ train model ให้ดีขึ้นไปอีก

-แนวทางการแก้ไข ทำการเก็บข้อมูลให้มากที่สุดเท่าที่จะทำได้ เพราะแม้ว่าข้อมูลที่ตอนแรกแม้จะดูเหมือนมีจำนวนมาก แต่เมื่อทำการ cleaning data แล้ว จำนวนข้อมูลอาจจะลดลงอย่างเห็นได้ชัด จึงเป็นทางเลือกที่ดีกว่า ที่จะรวบรวมข้อมูลให้มากที่สุดเท่าที่จะเป็นไปได้

5.2.3 การแสดงผล

เดิมทีนั้นผู้จัดทำเลือกที่จะทำงานบน Django เพียงอย่างเดียว เนื่องจาก Django เองนั้นมี template สำหรับการแสดงผลออกมาทาง html อยู่แล้ว แต่ทว่าการแสดงผลนั้น อาจจะมีข้อจำกัดมากมาย เนื่องจาก Django นั้นไม่ได้ถูกออกแบบให้เป็น full stack website เสียทีเดียว

-แนวทางการแก้ไข ผู้จัดทำจึงเลือกที่จะให้ Django เป็น back-end หรับการจัดการในการค้นหาข้อมูล และส่ง API มายัง Angular ซึ่งเป็น front-end framework ที่สามารถทำ feature ได้มากกว่า Django เช่น angular material, one page loading เป็นต้น

5.3 แนวทางในการพัฒนาต่อไปในอนาคต

5.3.1 เพิ่มฟังก์ชันในการแสดงผลพื้นที่ ที่มีสินทรัพย์โดยเฉลี่ยสูงที่สุด เพื่อที่จะสามารถนำข้อมูลตรงนี้มาหา insight เพิ่มเติมสำหรับการ predict สินทรัพย์ในอนาคต

5.3.2 เพิ่มฟังก์ชันในการ predict ราคาสินทรัพย์ เพื่อที่จะสร้าง standard สำหรับการประเมิณให้เป็น standard เดียวกันได้

5.3.3 เปลี่ยน stack เทคโนโลยีใหม่ เพราะว่าในโลกปัจจุบันนั้นถูก disrupt ไปด้วยความรวดเร็ว ดังนั้นย่อมเป็นเรื่องธรรมดาที่เราอาจจะต้องสร้าง stack ใหม่อยู่เสมอ อาทิเช่น การค้นหาอาจจะเปลี่ยนมาใช้ BERT ซึ่งเป็น Transformer ที่ได้รับความนิยมเป็นอย่างมากในหมู่ของ NLP Data Science หรือ

อาจจะเปลี่ยน Django เป็น Fast-API ซึ่งเป็น framework ที่เป็น framework สำหรับการสร้าง API ที่กำลังได้รับความนิยมในขณะนี้เช่นเดียวกัน

บรรณานุกรม

1.Deep Learning แบบฉบับคนสามัญ, 2 มีนาคม 2562 [ออนไลน์], Available:

Deep Learning แบบฉบับคนสามัญชน EP 1 : Neural Network History | by Mr.P L | mmp-li | Medium

2.Alpha Go logo

aplpha go logo - Bing images

3.เริ่มต้น Neural Networks กับ python, 20 เมษายน 2559 [ออนไลน์], Available:

เริ่มต้น Neural Networks กับ Python ~ Python 3 (wannaphong.com)

4.loss function, 24 ตุลาคม 2562 [ออนไลน์], Available:

Loss Function คืออะไร Cost Function, Error Function คืออะไร ทำงานอย่างไร ใน Machine Learning - Loss Function ep.1 - BUA Labs

5.optimization และ activation function, 21 พฤษภาคม 2562 [ออนไลน์], Available:

Deep Learning แบบฉบับสามัญชน EP 2 Optimization & Activation Function เรียนกันสบายๆ สไตส์ชิลๆ | by Mr.P L | mmp-li | Medium

6.พื้นฐาน deep learning (ทฤษฎี), 14 เมษายน 2563 [ออนไลน์], Available:

พื้นฐาน Deep Learning (ทฤษฎี): The Neural Network | by lLoLi | Medium

7.Regular Expression, 26 กรกฎาคม 2560 [ออนไลน์], Available:

Regular Expressions คืออะไร ?. หลายคนอาจจะเคยได้ยินเกี่ยวกับ Regular... | by Thirawat T. | Medium

8.Natural Language Processing เทคโนโลยีเชื่อมโยงปัญญาประดิษฐ์กับมนุษย์ด้วย “ภาษา”:

http://www.dv.co.th/blog-th/get-to-know-natural-language-processing-nlp/

9.TF-IDF, 29 พฤศจิกายน 2560 [ออนไลน์], Available:

TF-IDF คำไหนสำคัญนะ?. คำเตือน บทความนี้ อาจจะยาวนิดนึง... | by lukkiddd | lukkiddd

10.python, Available:

What is Python? Executive Summary | Python.org

11.Build Data pipeline on Google Cloud, Available:

Cloud Ace Thailand - Data pipeline on Google Cloud (cloud-ace.com)

12.Thai word segmentation with bi-directional RNN, 20 พฤษจิกายน 2560[ออนไลน์], Available:

Thai Word segmentation with bi-directional RNN (sertiscorp.com)