รายงานสหกิจศึกษาฉบับสมบูรณ์

การวิเคราะห์ Sentiment ภาษาอื่นด้วยภาษาจีน
Multi-Languages sentiment analysis based on Chinese sentiment

นายอมรภัทร จำปา

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

รายงานสหกิจศึกษาฉบับสมบูรณ์

การวิเคราะห์ Sentiment ภาษาอื่นด้วยภาษาจีน

Multi-Languages sentiment analysis based on Chinese sentiment

นายอมรภัทร จำปา

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ชื่อโครงงานสหกิจศึกษา	การวิเคราะห์ Sentiment ภาษาอื่นด้วยภาษาจีน
ชื่อ-สกุล นักศึกษา	นายอมรภัทร จำปา
คณะ วิศวกรรมศาสตร์	**ภาควิชา** วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ	อาจารย์บัณฑิต พัสยา
	อาจารย์จิระศักดิ์ สิทธิกร
ชื่อ-สกุล ผู้นิเทศงาน	นายวิภาส สุตันตยาวลี
สถานประกอบการ	บริษัท แบ็คยาร์ด จำกัด ประเทศไทย

บทคัดย่อ

สำหรับงานวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาวิธีการวิเคราะห์ความรู้สึกของข้อความด้วย Machine learning โดยใช้ Self-Attention เป็นโครงสร้างของโมเดล และนำไปต่อยอดกับการวัด sentiment score ของภาษาอื่นได้ด้วยวิธีการนำข้อความ ไปผ่าน Google translation API เพื่อแปลมาเป็นข้อความภาษาจีนและนำข้อความภาษาจีนที่ได้มาไปวัด sentiment score และงานวิจัยนี้ถูกนำไปเขียนเป็นเว็บให้ทดลองใช้ โดยเบื้องหลังของเว็บนั้นจะถูกเขียนด้วย Flask ซึ่งเป็น Python framework ตัวนึงที่ใช้พัฒนาเว็บ

คำสำคัญ : Machine learning, Self-Attention, Sentiment Analysis, Google Translate API, Flask

Co-operative Title: Multi-Languages sentiment analysis

(based on Chinese sentiment)

Student Intern Name : Mr.Amornpat Champa

Faculty : Engineering **Department :** Computer Engineering

Advisor name : Mr.Bundit Pasaya

Mr.Jirasak Sittigorn

Mentor name : Mr.Vipas Sutantayawalee

Company : Backyard Co, Ltd.

Abstract

The purpose of this research is for study methodology of Sentiment Analysis by Machine learning which is using Self-Attention as model structure. Moreover, we can measure sentiment other language by translate other language sentence into Chinese sentence with Google Translate API, and then we pass it to model for measure a sentiment score. This project is also applied as a website and the website is written by Flask, which is a Python Framework used to develop a website.

Keywords : Machine learning, Self-Attention, Sentiment Analysis, Google Translate API, Flask

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้เสร็จสมบูรณ์ได้ด้วยความช่วยเหลือจากหลายท่านทั้งทางตรง และทางอ้อม ซึ่งจะสำเร็จไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

ขอขอบคุณ อาจารย์ผู้นิเทศ อาจารย์บัณฑิต พัสยา ซึ่งเป็นผู้ที่มานิเทศงาน และช่วยให้คำแนะนำในการทำงาน การแก้ไขปัญหา และจุดบกพร่องของโครงงาน ซึ่งทำให้โครงงานสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ นายวิภาส สุตันตยาวลี หัวหน้างานที่ให้คำปรึกษา แนะนำหลักการในการทำงาน และรูปแบบของงานที่เป็นมาตราฐานตามหลักการที่ถูกต้องอย่างสม่ำเสมอ ซึ่งทำให้ตัวผลงานสำเร็จลุล่วงอย่างมีคุณภาพ

ขอขอบคุณ ทีม Data science ที่คอยสนับสนุน รวมทั้งให้คำปรึกษาเกี่ยวกับเรื่องต่างๆ ไม่ว่าจะเป็นตัวโครงงานหรือแม้แต่ตัวงาน ทำให้ตัวผลงานที่ทำออกไปนั้นมีคุณภาพและสามารถสื่อสารให้คนอื่นเข้าใจได้

สุดท้ายนี้ขอขอบคุณ บิดา มารดา ครอบครัว เพื่อนๆ ตลอดจนผู้เกี่ยวข้องที่ไม่ได้กล่าวนามทุกท่าน ที่เป็นกำลังใจ ให้การสนับสนุน และความช่วยเหลือในการทำโครงงานครั้งนี้จนสำเร็จลุล่วงไปได้

อมรภัทร จำปา

สารบัญ

บทที่ **หน้า**

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

ภาพที่ **หน้า**

บทที่ 1
บทนำ

1.1. ความเป็นมาและความสำคัญ

บริษัทแบ็คยาร์ดเป็นบริษัทที่ทำเกี่ยวกับข้อมูลตัวหนังสือ โดยข้อมูลเหล่านั้น ถูกเก็บมาจาก Social ต่างๆ ไม่ว่าจะเป็น Facebook, Instagram, twitter หรือเว็บไซต์ข่าวต่างๆ ทางบริษัทจะนำข้อมูลที่ได้มามาวิเคราะห์โดยหัวข้อต่างๆที่จะนำมาวิเคราะห์ก็มีทั้ง ปริมาณข้อมูลในแต่ละช่วงเวลาและความรู้สึกของข้อความ (Sentiment) และอื่นๆอีกมากมาย ในส่วนของความรู้สึกหรือ Sentiment นั้นจะไม่ได้ตัดสินจากตัวบุคคลโดยตรง แต่ตัดสินด้วยวิธีการที่กำหนดไว้ในโปรแกรม เนื่องจากข้อมูลมีปริมาณมากเกินไป จึงทำให้ไม่สามารถวิเคราะห์โดยใช้บุคคลเป็นหลักได้ ในส่วนวิธีการวิเคราะห์ Sentiment นั้น จะมีวิธีการคำนวณแบบการให้คะแนนคำ ซึ่งจะนำมารวมในตอนสุดท้าย โดยผลรวมสุดท้ายสามารถบอกได้ว่า ข้อความทั้งหมดนี้โดยรวมเป็น ข้อความเชิงบวกหรือข้อความเชิงลบ รวมไปถึงข้อความที่เป็นกลางด้วย (หรือสามารถข้อความประเภทคำถามก็ได้) และปัจจุบันทางบริษัทได้ทดลองเกี่ยวกับการทำ Sentiment โดยใช้ Machine learning เข้ามาช่วย ณ ตอนนั้นจึงเป็นโอกาสที่ผมได้มีส่วนร่วมในการเข้าไป research ร่วมกับบริษัท โดยทีม Data science จะแบ่งงาน Project กัน ในส่วนของผมจะได้รับ Project เป็น การทำ Sentiment ในภาษาต่างประเทศ โดยโจทย์ในตอนนั้นจะมีอยู่ 4 ภาษา ได้แก่ ภาษาพม่า ภาษากัมพูชา ภาษาญี่ปุ่นและภาษาจีน โดยภาษาที่เลือกมาทำการทดลองนั้นคือภาษาพม่าและภาษาจีน ในตอนเริ่มต้นของ Project ผมเลือกที่จะทำ Sentiment ของภาษาพม่าก่อน และได้พบว่า ภาษาพม่าเป็นภาษาที่มีแหล่งข้อมูลต่ำ (low resource) ซึ่งทำให้จำเป็นจะต้องหยุดพักโปรเจคภาษาพม่าและเริ่มต้นที่ภาษาจีนก่อน เมื่อได้ศึกษาค้นขวาโปรเจคภาษาจีน จะพบได้ว่า จีนเป็นประเทศที่มีเทคโนโลยีสูง การทำ Sentiment ในภาษาจีนนั้นสามารถศึกษาจาก Project ที่มีมาก่อนได้ รวมถึงมีการทำ Public dataset ไว้อยู่แล้ว แต่วิธีการที่จะหา Project ที่ทำไว้แล้วนั้นรวมถึงเทคโนโลยีที่จีนใช้ จะง่ายขึ้นหากเราค้นหาด้วยคำภาษาจีน ทั้งนี้การค้นหา Project ที่สามารถหาได้รวมถึง Dataset ที่สามารถหาได้มาทดลองใช้งาน เพื่อการทำ Research ของบริษัทและสุดท้ายจึงได้ออกมาเป็น Sentiment Analysis model ออกมา หลังจากนั้นจึงมีการทดลองต่อว่าหาก เรามี Model ที่สามารถบ่งบอกได้ถึง Sentiment แล้ว ถ้าเรานำ Model ตัวนี้ไปประยุกต์กับภาษาอื่น จะมีผลลัพธ์เป็นอย่างไร จึงเป็นที่มาของโครงการนี้

1.2. วัตถุประสงค์ของการวิจัย

1. วิเคราะห์ Sentiment ภาษาจีนด้วย Machine learning
2. วิเคราะห์ Sentiment ในภาษาต่างๆด้วย Machine learning โดยมีภาษาจีนเป็นพื้นฐาน

1.3. ขอบเขตการวิจัย

เครื่องมือสำหรับการวิเคราะห์ Sentiment แบ่งระบบออกเป็น 2 ส่วนหลักๆ ดังนี้

1. โมเดลวิเคราะห์ Sentiment (Sentiment analysis Model)
2. เว็บไซต์สำหรับแสดงผลการใช้งาน (Website)

1.4. วิธีดำเนินการวิจัย

1. รวบรวมข้อมูลที่ต้องใช้ในการทำ Model
2. ศึกษา Machine learning เพื่อนำมาทำระบบ
3. ออกแบบระบบ
4. เขียนโปรแกรม
5. วัดประสิทธิภาพของ Model
6. สาธิตระแบบให้ผู้นิเทศงาน
7. จัดทำรายงาน

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1. การนำเทคโนโลยี Machine learning มาแก้ปัญหาในเชิงทางธุรกิจ
2. รู้จุดแข็งและจุดอ่อนของการทำ Sentiment ข้ามภาษา

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1. โปรแกรมและ Library ที่ใช้ในการพัฒนา

2.1.1 Python3

ภาพที่ 2.1.1 Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถทำงานภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครบคุมกับทุกลักษณะงาน

ภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจว่าจะให้เป็นภาษาที่อ่านง่าย จึงถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าภาษา C และ Pascal

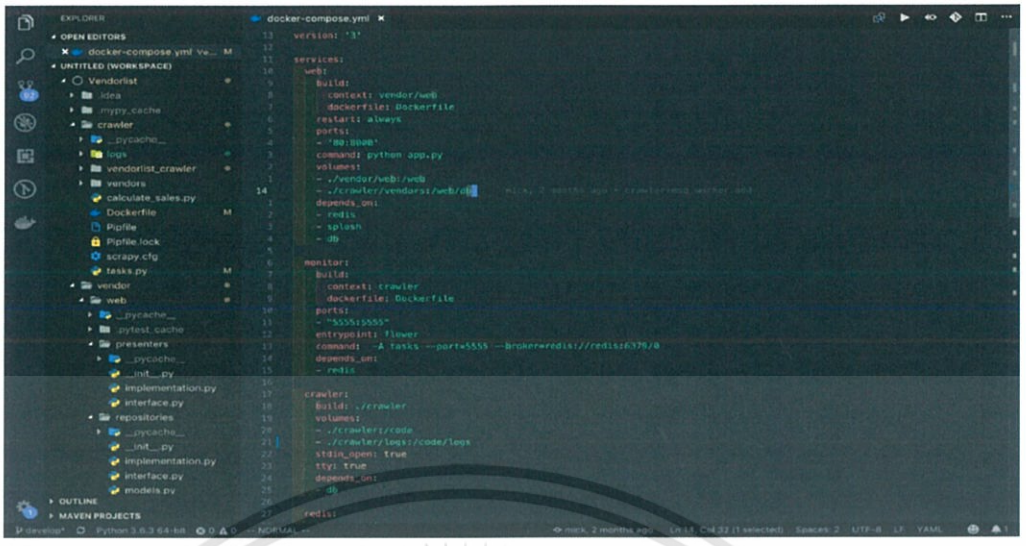
Python interpreter นั้นเป็นตัวแปรภาษาของภาษา Python เพื่อให้สามารถทำงานโค้ด Python ได้ ซึ่งได้มากับไลบรารี่มาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ https://www.python.org/ ซึ่งเป็นโปรแกรมแบบ source และ binary สำหรับแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ Interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดของภาษา Python ลงไปและเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที Python Interpreter นั้นยังสามารถนำเพิ่มความสามารถกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ Python นั้นเหมาะสำหรับเป็นภาษาในการสร้าง Extension และแอพพลิเคชั่นที่ปรับแต่งได้

2.1.2 Visual Studio Code

ภาพที่ 2.1.2 Visual studio Code

Visual Studio Code หรืออีกที่ชื่อที่เรียกกันว่า VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขหรือปรับเปลี่ยนโค้ด จากค่ายไมโครซอฟท์ ที่มีการสร้างออกมาในรูปแบบของ Open Source จึงสามารถนำมาใช้งานได้แบบฟรีๆ โดยไม่มีการเสียค่าใช้จ่ายใดๆทั้งสิ้น เป็นโปรแกรมที่นำมาใช้เพื่องานที่ต้องการความเป็นมืออาชีพ

โดย Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาก ไม่ว่าจะเป็น 1.การเปิดใช้รองรับมากกว่า 30 โปรแกรมภาษาอะไรบ้าง เช่น C++, C#, CSS, Dockerfile, HTML, JavaScript, JSON, Less, Markdown, PHP, Python, Sass, TypeScript ที่สำคัญรองรับภาษา Java อีกด้วย2.Themes 3.Debugger 4.Commands เป็นต้น โดยสิ่งที่แตกต่างกันของ Visual Studio Code กับ Microsoft Visual Studio ทั้งๆที่เป็น Code Editor เหมือนกันนั้นก็คือ Microsoft Visual Studio นั้นจะมี .net framework ส่วน Visual Studio Code นั้นไม่มีทำให้เหมาะกับองค์การ์ที่ไม่ต้องการพัฒนาโปรแกรมโดยใช้เทคโนโลยี .net framework หรือจะมองได้ว่า Visual Studio Code เอาไว้สำหรับนักพัฒนาโปรแกรม ที่ไม่ใช้ Microsoft Windows นำเอาไปใช้งาน โดยสัญลักษณ์ของโปรแกรม Visual Studio Code จะอยู่ใน ภาพที่ 2.2.1 และหน้าตาของโปรแกรม Visual Studio Code จะอยู่ในภาพที่ 2.2.2 ตามลำดับ



ภาพที่ 2.1.3 หน้าตาโปรแกรม Visual Studio Code

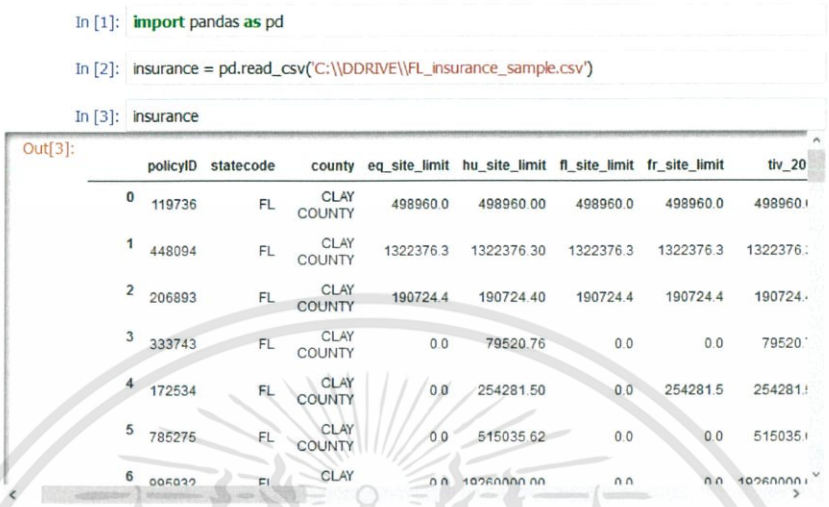
2.1.3 Jupyter Notebook

ภาพที่ 2.1.4 Jupyter Notebook

Jupyter Notebook นั้นสามารถทำ Data Science ได้เป็นอย่างดีเนื่องด้วยการแสดงผลลัพธ์ที่เป็นกราฟ หรือ การแสดงผลลัพธ์ที่เป็นตารางก็ย่อมได้ อีกทั้ง Jupyter Notebook นั้นสามารถทำ Markdown ได้ในแต่ละ บรรทัด เพื่อเขียนคำอธิบายโค้ดในแต่ละบรรทัดทำให้คนที่เข้ามาอ่านโค้ดสามารถเข้าใจโค้ดได้อย่างง่ายดายเพราะตัว Jupyter จะไม่ส่งคำอธิบายเหล่านั้นไปให้ Python ประมวลผลการทำงาน

ข้อดีของ Jupyter Notebook

1. สามารถอ่านไฟล์ CSV,EXCEL หรือไฟล์อื่นๆได้



```
In [1]: import pandas as pd
In [2]: insurance = pd.read_csv('C:\\DDRIVE\\FL_insurance_sample.csv')
In [3]: insurance
```

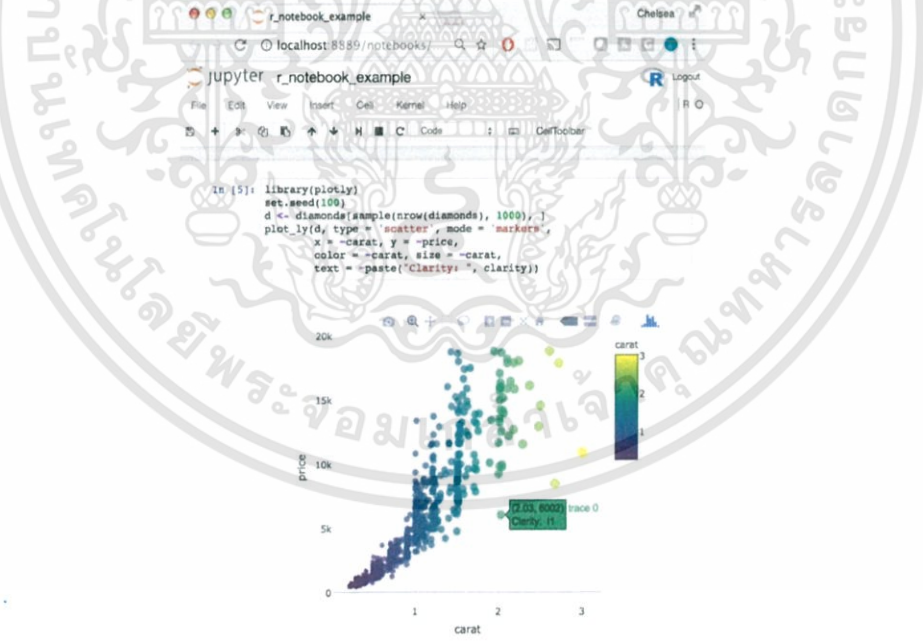
Out[3]:

	policyID	statecode	county	eq_site_limit	hu_site_limit	fl_site_limit	fr_site_limit	tiv_20
0	119736	FL	CLAY COUNTY	498960.0	498960.00	498960.0	498960.0	498960.
1	448094	FL	CLAY COUNTY	1322376.3	1322376.30	1322376.3	1322376.3	1322376.
2	206893	FL	CLAY COUNTY	190724.4	190724.40	190724.4	190724.4	190724.
3	333743	FL	CLAY COUNTY	0.0	79520.76	0.0	0.0	79520.
4	172534	FL	CLAY COUNTY	0.0	254281.50	0.0	254281.5	254281.
5	785275	FL	CLAY COUNTY	0.0	515035.62	0.0	0.0	515035.
6	995932	FL	CLAY	0.0	19260000.00	0.0	0.0	19260000.

ภาพที่ 2.1.5 Read File

ที่มา: http://www.mindphp.com/forums/viewtopic.php?f=144&t=47184

2. สามารถแสดงผลลัพธ์ของโค้ดให้เป็นรูปแบบของ กราฟได้

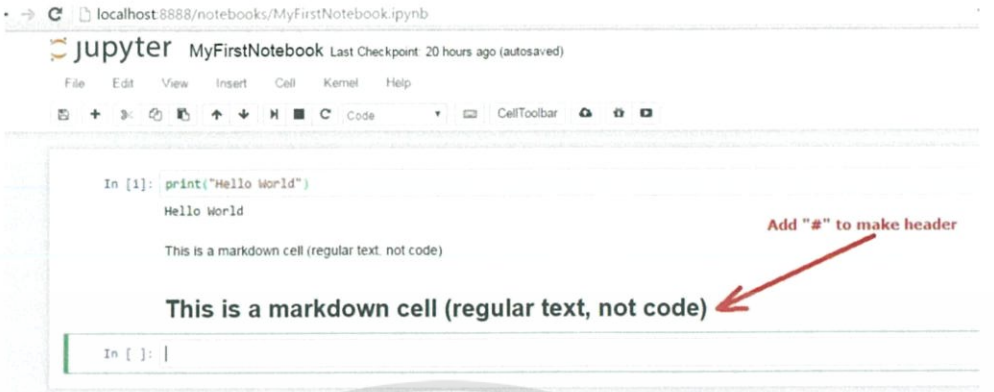


ภาพที่ 2.1.6 Plot graph

ที่มา:

http://www.mindphp.com/forums/viewtopic.php?

3. สามารถทำ Markdown เพื่อทำการชี้แจงโค้ดแต่ละบรรทัดได้อย่างชัดเจน



ภาพที่ 2.1.7 Markdown

ที่มา: http://www.mindphp.com/forums/viewtopic.php?f=144&t=47184

2.1.4 Git

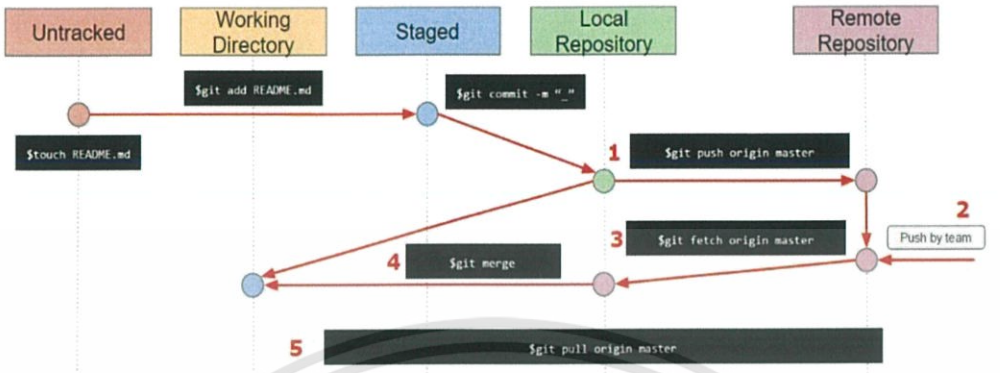
Version Control System(VCS) แบบ Distributed ตัวหนึ่ง เป็นระบบที่ใช้จัดเก็บและควบคุมการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ชนิดใดก็ได้ เพื่อที่คุณสามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งกลับมาดูเมื่อไรก็ได้ ไม่ว่าจะเป็น Text File หรือ Binary File ก็ตาม นอกจากนั้นระบบ VCS ยังจะช่วยให้คุณเปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต ดูว่าใครเป็นคนแก้ไขคนสุดท้ายที่อาจทำให้เกิดปัญหา แก้ไขเมื่อไร ฯลฯ และยังช่วยให้คุณสามารถกู้คืนไฟล์ที่คุณลบหรือทำเสียโดยไม่ตั้งใจได้อย่างง่ายดาย

Git Status

สถานะของ Source Code ที่เก็บอยู่ในระบบของ Git นั้นมีดั่งนี้

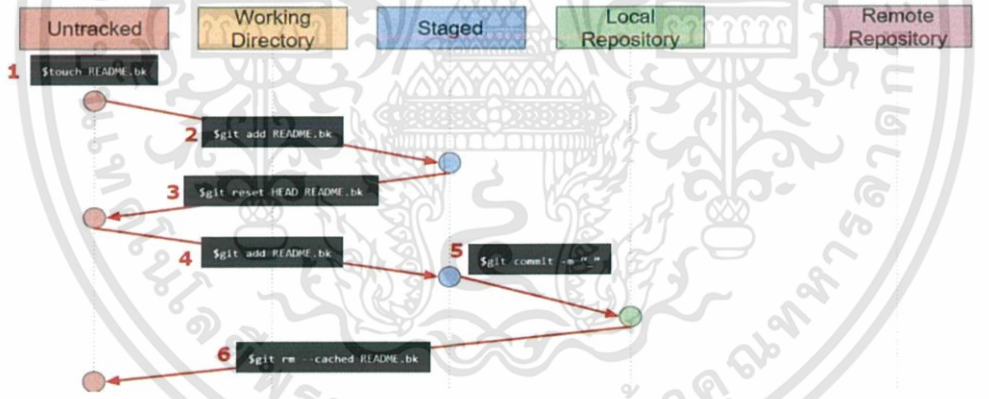
- Untracked เป็นสถานะที่ Source Code ถูกเพิ่มเข้ามาใหม่และยังไม่ได้ถูกเก็บไว้ในระบบของ Git
- Working Directory เป็นสถานะที่กำลังมีการเปลี่ยนแปลงหรือแก้ไข Source Code หรืออาจจะเรียกสถานะนี้ว่า Modified
- Staged เป็นสถานะที่ Source Code กำลังเตรียมที่จะ Commit เพื่อยืนยันการเปลี่ยนแปลงก่อนที่จะเก็บลงในสถานะ Local Repository
- Local Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Local (ที่เครื่องตัวเอง)

Remote Repository เป็นสถานะที่มีการเก็บบันทึกข้อมูลการเปลี่ยนแปลงของ Source Code ลงไปที่ Git Repository ที่เป็น Hosting (ที่เครื่องเซิร์ฟเวอร์)



ภาพที่ 2.1.8 Git Forward

ที่มา: https://medium.com/@pakin/git-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-git-is-your-friend-c609c5f8efea



ภาพที่ 2.1.9 Git Backward

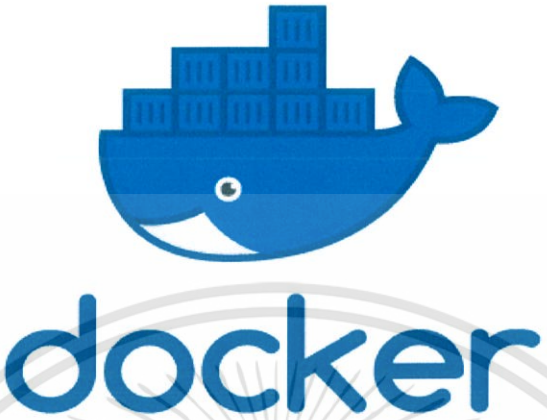
ที่มา: https://medium.com/@pakin/git-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-git-is-your-friend-c609c5f8efea

2.1.5 Flask

ภาพที่ 2.1.10 Flask Logo

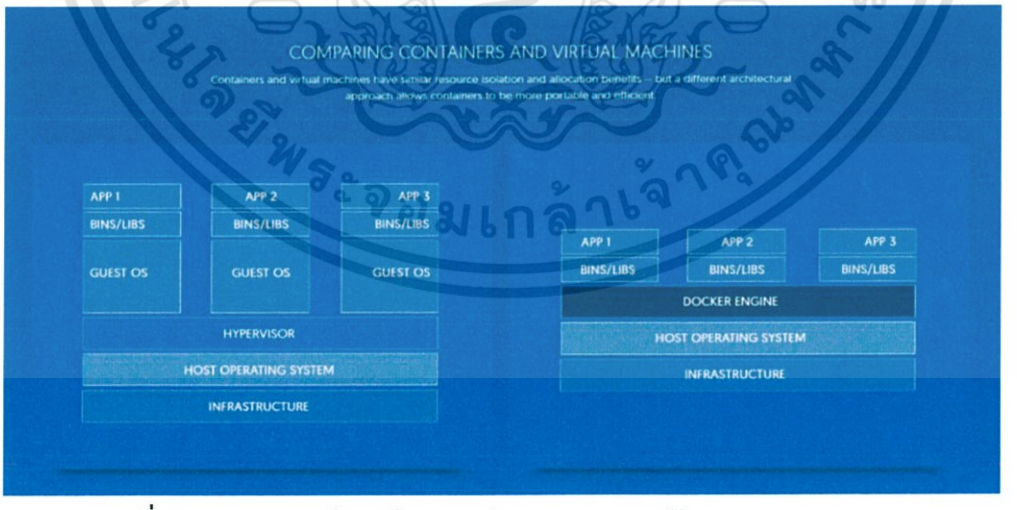
Flask คือ web framework ที่เขียนขึ้นมาสำหรับ Python เพื่อใช้ร่วมกัน webserver เช่น Apache และได้รับการยอมรับจาก community wepages ชั้นนำเช่น Pinterest, LinkedIn เป็นต้น โดย Flask ถูกเรียกว่า micro framework เพราะว่า มันไม่ต้องการเครื่องมือ หรือ library อะไรมาก อีกทั้ง ไม่จำเป็นต้องมี database ด้วย แต่อย่างไรก็ตาม Flask ก็ยังรองรับการเพิ่ม extensions พิเศษได้ ถ้ามันรองรับ Flask

2.1.6 Docker



ภาพที่ 2.1.11 Docker Logo

Docker คือ engine รูปแบบหนึ่งที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server เพื่อใช้ในการ run service ที่ต้องการ มีการทำงานคล้ายคลึงกับ Virtual Machine เช่น VMWare, VirtualBox, XEN, KVM แต่ข้อแตกต่างที่ชัดเจนคือ Virtual Machine ที่รู้จักกันก่อนหน้านี้นั้น เป็นการจำลองทั้ง OS เพื่อใช้งานและหากต้องการใช้งาน service ใดๆ จึงทำการติดตั้งเพิ่มเติมบน OS นั้นๆ แต่สำหรับ docker แล้วจะใช้ container ในการจำลองสภาพแวดล้อมขึ้นมา เพื่อใช้งานสำหรับ 1 service ที่ต้องการใช้งานเท่านั้น โดยไม่ต้องมีส่วนของ OS เข้าไปเกี่ยวข้องเหมือน Virtual Machines อื่นๆ



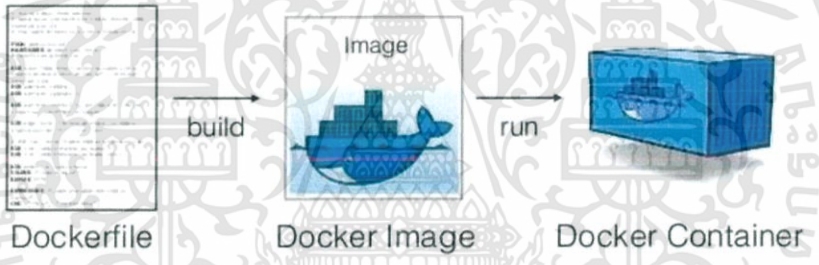
ภาพที่ 2.1.12 การเปรียบเทียบระหว่าง Container กับ Virtual Machines

ที่มา: https://saixiii.com/python-flask-web-application/

Docker นั้น เป็นที่รู้จักกันอย่างแพร่หลายในช่วง 1-2 ปีที่ผ่านมานี้ เนื่องจากสามารถใช้งานได้อย่างสะดวก และตอบสนองความต้องการของ ผู้พัฒนาโปรแกรม (Developer) หรือ ผู้ดูแลระบบ (System admin)

Docker image เรียกได้ว่าเป็นพิมพ์เขียว หรือเป็นต้นแบบของการสร้าง container ขึ้นมาใช้งาน ถ้าเปรียบเทียบกับการเขียนโปรแกรมเชิงวัตถุ(OOP) เจ้าตัว docker image นี้ก็เปรียบเสมือน class ซึ่งเป็นต้นแบบในการสร้าง object (docker container) อีกทีครับ Docker image สามารถเรียกใช้ได้สองวิธี นั่นก็คือการสร้างจาก Dockerfile ที่เป็นสคริปต์ที่อธิบายว่าเราจะติดตั้งอะไรลงไปใน image บ้าง หรือดาวน์โหลดและเรียกใช้งาน docker image ที่มีอยู่แล้วบนอินเตอร์เน็ต

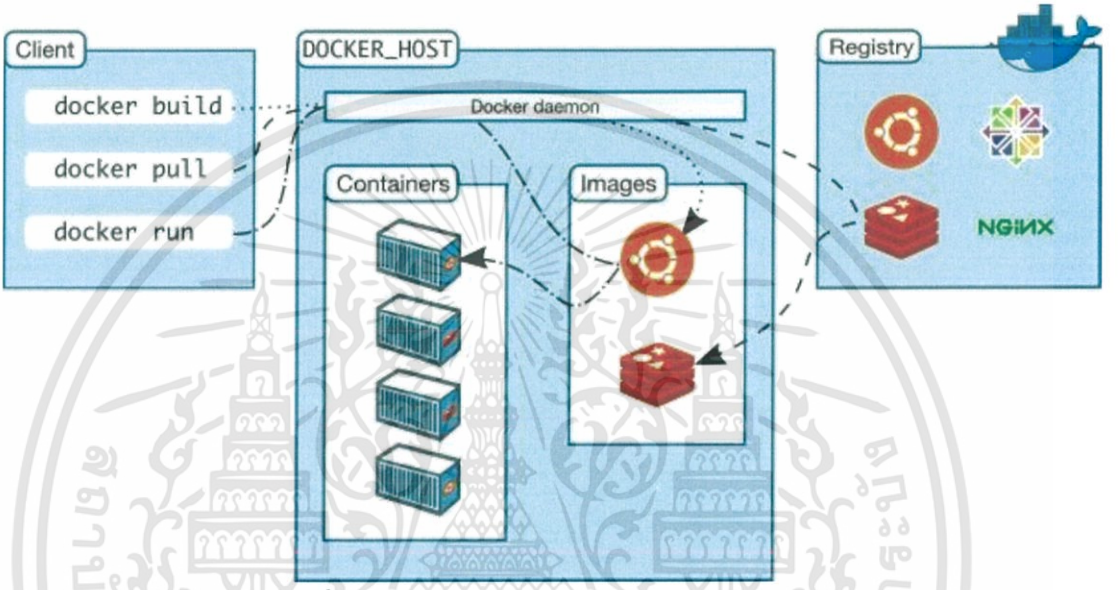
Docker container สามารถมองได้เสมือนกล่อง ซึ่งนำ docker image มาติดตั้ง เพื่อให้สามารถใช้งาน service ที่ต้องการจาก image นั้นๆ ได้ โดยใน container แต่ละตัวจะมีการใช้งาน RAM, CPU, ไฟล์ config ต่างๆ เป็นของแต่ละ container เอง และยังสามารถสั่ง start, stop ได้ที่ container นั้นๆ อีกด้วย



ภาพที่ 2.1.13 การสร้าง Container

ที่มา: https://saixiii.com/python-flask-web-application/

Docker Registry นอกจากเราจะสร้าง docker image มาเพื่อใช้รันบนเครื่องตัวเองแล้ว เราสามารถอัพโหลดเข้าไปสู่ server กลางสักอันเพื่อที่จะเอาไปใช้งานบนเครื่องอื่นๆ ได้ด้วย โดย server ที่ว่านี้เราจะเรียกว่าเป็น docker registry มีอันหนึ่งที่มีชื่อเสียงและถูกใช้เป็น registry หลักนั่นก็คือ DockerHub (เปรียบเหมือน Github สำหรับ docker image นั่นเอง) ซึ่งที่นี่นี่เอง ที่เป็นจุดศูนย์รวมของ docker images ต่างๆ จากผู้คนทั่วโลก และเราสามารถหยิบจับ image ที่มีอยู่แล้วมาใช้งานเลย หรือเอามาดัดแปลงให้เข้ากับการใช้งานของเราเองก็ได้



ภาพที่ 2.1.14 Work Flow ของ Docker

ที่มา: https://saixiii.com/python-flask-web-application/

2.1.7 Pytorch

ภาพที่ 2.1.15 Pytorch Logo

Pytorch ถูกพัฒนาขึ้นโดย Facebook โดยดัดแปลงมาจากไลบรารีชื่อ torch ซึ่งถูกใช้ในภาษา Lua มาก่อน เริ่มใช้งานตั้งแต่ปี 2016

ลักษณะ Pytorch จะคล้ายกับเฟรมเวิร์กที่เป็นที่นิยมอีกตัวคือ chainer คือมีการใช้เทนเซอร์ที่คล้ายกับอาเรย์ของ numpy เป็นตัวคำนวณหลัก และสร้างโครงข่ายประสาทเทียมด้วยการนิยามขณะวิ่ง (define by run)

โดยทั่วไปโครงข่ายจะถูกสร้างขึ้นจากการนำชั้นต่างๆมาประกอบกัน ชั้นต่างๆที่จำเป็นโดยทั่วไปถูกเตรียมไว้ภายในโมดูลอย่างครบถ้วนแล้ว เทียบกับ tensorflow แล้ว Pytorch มีลักษณะที่ค่อนข้างสำเร็จรูป ใช้งานง่ายกว่ามาก จึงเหมาะสำหรับผู้ที่ฝึกหัดใหม่มากกว่า แต่ก็จะไม่ได้สำเร็จรูปเท่า keras ยังปรับแต่งอะไรได้อิสระกว่า เทนเซอร์ภายใน pytorch มีคำสั่งและการใช้งานคล้ายกับ numpy มาก ทำให้คนที่ชินกับ numpy อยู่แล้วสามารถใช้ pytorch ได้โดยไม่ต้องเรียนรู้คำสั่งใหม่มาก

สรุปความสามารถโดยรวมของ pytorch

- เตรียมออบเจ็กต์ชนิด Tensor ซึ่งสามารถคำนวณได้แบบอาเรย์ของ numpy แต่เพิ่มความสามารถในการคำนวณอนุพันธ์เข้ามา
- เตรียมชั้นต่างๆสำหรับใช้ประกอบเป็นโครงข่ายไว้พร้อม แค่นำมาต่อๆกันก็สร้างโครงข่ายประสาทเทียมแบบต่างๆได้อย่างง่าย
- มีฟังก์ชันสำหรับจัดการข้อมูลเบื้องต้นก่อนนำมาใช้เป็นข้อมูลป้อนเข้า
- มีฟังก์ชันสำหรับแปลงและตัดแต่งรูปภาพเพื่อเพิ่มความหลากหลายให้ข้อมูลรูปภาพ
- มีฟังก์ชันช่วยดึงชุดข้อมูลตัวอย่าง เช่น MNIST, CIFAR, ฯลฯ
- สามารถใช้ GPU ในการคำนวณได้

2.1.8 Gensim

ภาพที่ 2.1.16 Gensim Logo

Gensim เป็น Library ฟรีของ Python ที่ออกแบบมาเพื่อแยกหัวข้อหรือความหมายโดยอัตโนมัติจาก document และ Gensim ยังออกแบบมาเพื่อประมวล raw data, unstructured digital texts ("plain text") อีกด้วย

Algorithm ที่อยู่ใน Gensim library ก็จะมี Word2Vec, FastText, Latent Semantic Analysis (LSI, LSA), Latent Dirichlet Allocation (LDA) และอื่นๆอีกมากมาย

ทาง Gensim สามารถหาโครงสร้างความหมายของ Document ได้อัติโนมัติ โดยการตรวจสอบรูปแบบการเกิดร่วมกันเชิงสถิติภายในคลัง training document และAlgorithm พวกนี้เป็นประเภท unsupervised

2.1.9 Google Translate API

ภาพที่ 2.1.17 Google translate

Google Translate API คือ API ตัวนึงที่ทาง Google ปล่อยออกมาให้ใช้ สำหรับวิธีใช้งาน จะมีหลากหลายวิธี เพราะสามารถนำไปใช้บนเว็บหรือเขียนใน Python ก็ได้

สำหรับขั้นตอนการใช้คือ

1. ทำการสมัคร Google Account
2. ทำการเข้า Google Cloud ผ่านเว็บ www.cloud.google.com
3. ทำการ Active free 300$ trial ของ Google เพื่อทดลองใช้งาน API บน Google ฟรี 300$ (การใช้งาน Google API จริงๆจะต้องเสียเงิน แต่ทาง Google ได้ให้ใช้งานฟรี 300$ ภายใน 1 ปี)
4. สร้าง Project ใน Google Cloud
5. ทำการ Authentication บน Environment ของเครื่องเพื่อใช้ API (เหตุผลที่ต้อง Authentication เพื่อที่จะสามารถส่ง Request ไปหา Google ได้)

เราสามารถำการใช้งาน Google API บน python ได้ด้วยการ install library ของ Google Cloud (ยกตัวอย่างเช่น pip install google-cloud) และสามารถใช้งานได้เลย ซึ่งสามารถไปศึกษาเพิ่มเติมได้ใน Document ของทาง Google API

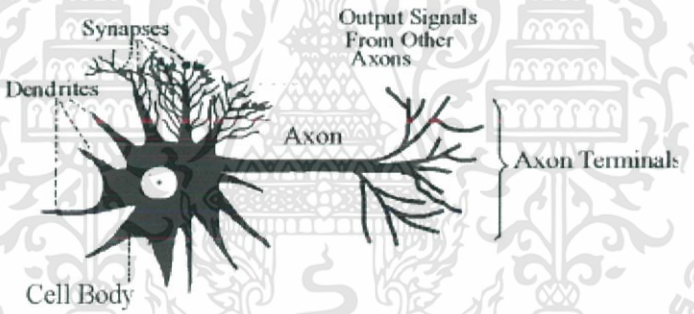
2.2. ทฤษฎีที่เกี่ยวข้อง

2.2.1 Neural Network

Neural Network หรือ Artificial Neural Networks (ANNs) คือโมเดลที่จำลองการทำงานแบบสมองมนุษย์ การทำงานของสมองมนุษย์นั้นค่อนข้างมีความซับซ้อนและเป็นสิ่งมหัศจรรย์ แม้แต่การทำงานของสมองสัตว์ชนิดๆต่าง ก็มีความซับซ้อนเช่นกันเมื่อเทียบกับคอมพิวเตอร์ คอมพิวเตอร์มีความสามารถที่ทำอะไรหลายอย่างได้ดี แต่บางทีก็ไม่สามารถเข้าใจระบบง่ายๆได้

การทำงานของสมองมนุษย์ยังคงเป็นเรื่องที่ลึกลับและซับซ้อนอยู่ มีแค่บางส่วนเท่านั้นที่พอจะเข้าใจ แต่สิ่งที่เป็นพื้นฐานที่เราสามารถเข้าใจได้ดีที่สุดคือ เซลล์ประสาท (Neuron) โดยเซลล์ประสาทเหล่านี้ทำให้เราจำ คิดและเรียนรู้เกี่ยวกับประสบการณ์ที่เคยผ่านมาได้ และแสดงออกมาได้ในทุกการกระทำของเรา

โดยพื้นฐาน neuron จะรับ input มาจากที่ใดที่นึงหลังจาก นั้นจะทำการรับ input ทั้งหมดด้วยวิธีใดวิธีนึงซึ่งจะได้คำตอบมา และหลังจากนั้น จะนำคำตอบมาผ่าน operation บางอย่างที่เป็น nonlinear และส่งออกเป็น output



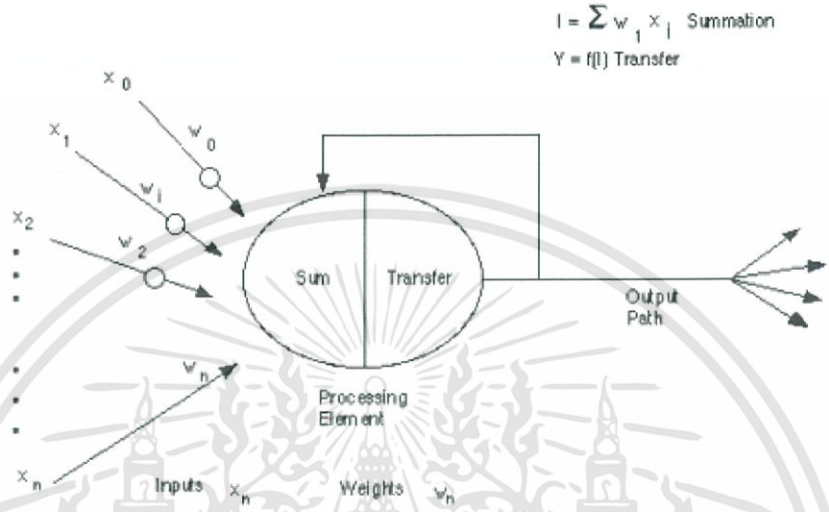
ภาพที่ 2.2.1 Biological Neuron

ที่มา:

http://shodhganga.inflibnet.ac.in/bitstream/10603/48/6/chaper%204_c%20b%20bangal.pdf

- Cell body (Soma) : เป็นร่างกายของเซลล์ประสาทซึ่งประกอบไปด้วย nucleus และทำหน้าที่เปลี่ยนแปลงทางชีวเคมีที่จำเป็นต่อชีวิตของเซลล์ประสาท
- Dendrite : เซลล์ประสาทแต่ละตัวมีเส้นผมละเอียดคล้ายท่อรอบๆเซลล์ มันแตกออกเป็นต้นไม้รอบๆเซลล์ และคอยรับสัญญาณที่เข้ามา
- Axon : คือท่อที่ยาวและบางซึ่งทำหน้าเป็นท่อส่ง

- Synapse : เซลล์ประสาทที่เชื่อมต่อกันอย่างซับซ้อน เมื่อ axon ถึงปลายทางสุดท้าย มันจะเริ่มแตกกิ่งอีกรอบ ซึ่งเราเรียกว่า terminal arborization ที่ปลายท่อ axon นั้นมีความซับซ้อนสูงและมีโครงสร้างที่พิเษเรียกว่า synapse การเชื่อมต่อระหว่าง 2 neuron จะเกิดขึ้นที่ synapse



ภาพที่ 2.2.2 Neuron Network

ที่มา:

http://shodhganga.inflibnet.ac.in/bitstream/10603/48/6/chaper%204_c%20b%20bangal.pdf

จากภาพ input ต่างๆ ถูกแทนด้วย สัญลักษณ์ทางคณิตศาสตร์ โดย input แต่ละตัว (X_n) จะถูกคูณด้วยค่า weight (W_n)ซึ่งผลลัพธ์คือ product หลังจากนั้นจะทำการนำ product ทุกตัวมารวมกันและป้อนให้กับ transfer function (Activation function) เพื่อสร้าง result ซึ่งจะถูกส่งออกไปเป็น output ในที่สุด

Neural Network มีส่วนประกอบหลักที่สำคัญทั้งหมด 7 อย่าง โดยที่ส่วนประกอบเหล่านี้ถูกต้องแน่นอนไม่ว่าจะถูกใช้ใน layer ใดก็ตาม

2.2.1.1 Weighting Factors

เซลล์ประสาทมักได้รับ input หลายๆครั้งพร้อมกัน โดย input แต่ละตัวนั้นจะมี relative weight เป็นของตนเอง ซึ่งจะก่อให้เกิดผลกระทบเกี่ยวกับ input ที่ได้รับเข้ามา จึงจำเป็นต้องใช้ processing element's summation function เพราะบาง input มีความสำคัญมากกว่าตัวอื่น weight จะมีการปรับค่าตามการจดจำและฝึกฝนของ neural network

2.2.1.2 Summation Function

input และ weight ถูกแสดงอยู่ในรูปของ vector (i1, i2, i3 ... in) และ (w1, w2, w3 ... wn) และผมรวมทั้งหมด (product) จะเกิดจาก (i1 * w1) + (i2 * w2) ... + (in + wn) โดยผลรวมจะออกมาเป็นค่าเดียว (Single value)

Summation function อาจจะมีกระบวนการซับซ้อนมากกว่าแค่การบวกระหว่าง input ทุกตัว บางทีอาจจะใช้ค่าต่ำสุด สูงสุด หรือค่าที่มีความถี่สูงสุดเพื่อสร้างเหมาะสมต่องาน และในบางครั้งอาจจะมีการเพิ่ม Activation function กับ product ที่ได้มาก่อนที่จะถูกส่งไปให้ transfer function เพื่อ product นั้นมีค่าแตกต่างกันตามเวลาที่ถูกป้อนเข้ามา

2.2.1.3 Transfer Function

ผลลัพธ์ของ Summation function จะถูกเปลี่ยนเป็น output ซึ่งผ่านกระบวนการอัลกอรึทึมที่เรียกว่า Transfer function ใน Transfer function สามารถนำ Summation function มาเปรียบเทียบกับเกณฑ์บางอย่างเพื่อหา output ของ Neural Network ถ้าผลรวมมากกว่า threshold จะสร้าง signal และถ้าค่านั้นน้อยกว่า threshold จะไม่มีการ signal การตอบสนองทั้ง 2 ประเภทมีความสำคัญ เกณฑ์หรือ transfer function ทั่วไปคือ non-linear.

2.2.1.4 Scaling and Limiting

หลังจากผ่าน Transfer function มาแล้ว เราสามารถนำ result มาผ่าน addition function นี้ได้ เพื่อทำการ scaling โดยการคูณ transfer value และ offset เข้าไป แล้วทำการ limiting เพื่อเช็คว่าค่าที่สเกลมานั้นมีขนาดเกินขอบเขตที่เคยกำหนดไว้หรือไม่

2.2.1.5 Output Function (Competition)

neuron แต่ละ node จะมี output อยู่แค่หนึ่งตัว ซึ่งสามารถส่ง output นี้ต่อให้กับ neuron อื่นอีกเป็นร้อยๆ node โดยปกติผลลัพธ์ของ output มักจะเป็นผลลัพธ์ที่ผ่าน transfer function มาแล้ว

2.2.1.6 Error Function and Back-Propagated Value

ในการเรียนรู้ของ network นั้นส่วนใหญ่ความแตกต่างระหว่าง current output กับ desired output ถูกคำนวณออกมาเป็น error ซึ่งจะถูกแปลงโดย error function เพื่อปรับให้ตรงกับโครงสร้างของ network โดยโครงสร้างทั่วไปมักใช้ error นี้โดยตรง แต่บางโครงมีการปรับเปลี่ยนให้ตรงกับวัตถุประสงค์ของเขา error จะถูกส่งกลับไปยังชั้นก่อนหน้า error ที่ถูกส่งกลับมาอาจเป็นค่าที่ error หรือ output อื่นๆ โดยปกติ Back-Propagated Value หลังจากถูกปรับขนาดที่ learning function แล้ว จะทำการคูณกับน้ำหนัก connection weight ที่เข้ามา เพื่อปรับเปลี่ยนก่อนที่จะถูกส่งไปยัง learning cycle

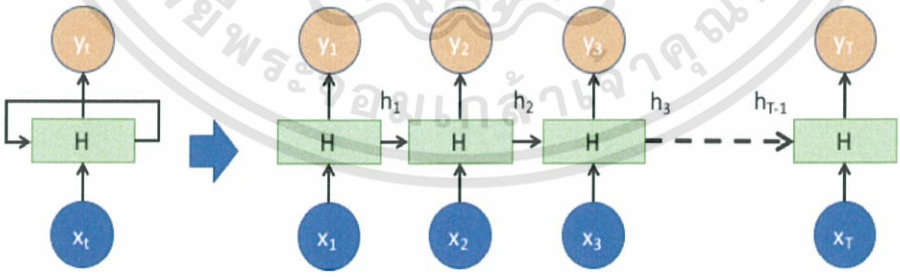
Learning Function : จุดประสงค์เพื่อปรับเปลี่ยนน้ำหนักของ input ในแต่ละ neuron ให้เข้ากับ algorithm ของ network นั้นๆ

2.2.2 Recurrent Neural Network (RNN)

RNN หรือ Recurrent Neural Network เป็นโครงสร้าง network ที่เหมาะกับข้อมูลที่มีลักษณะเป็นลำดับ (sequence) เช่น video (sequence of images) หรือ text (sequence of words) เพื่อที่ให้เข้าใจภาพการทำงานของ RNN ได้ง่ายขึ้น ขอยกตัวอย่างการอ่านหนังสือซึ่งมีลักษณะการทำงานแบบ sequential data เวลาอ่านหนังสือเราจะอ่านทีละคำ จากซ้ายไปขวา (สำหรับภาษาไทย หรือภาษาอังกฤษ) การที่เราสามารถรู้เรื่องได้ว่าประโยคที่เรากำลังอ่านนั้นเกี่ยวกับอะไร เกิดจากการที่เราเอาเรื่องราวจากสิ่งที่เราอ่านผ่านไปแล้ว (state ก่อนหน้า) มาผสมกับคำที่กำลังอ่านอยู่ (input data) ทำให้เราเข้าใจความหมายในส่วนตรงที่กำลังอ่านได้ ซึ่ง RNN ก็ใช้หลักการเดียวกัน คือ การปรับรูปแบบของ Neural network เดิม เพื่อให้สามารถเอา state หรือความรู้ก่อนหน้า มาบวกกับ input data ตัวใหม่ที่เข้ามา เพื่อทำความเข้าใจอะไรสักอย่างไปเรื่อยๆ เพื่อความง่ายต่อการอธิบายการทำงานต่อไป ขอกำหนดตัวแปรไว้ดังนี้

- input data คือ $x_1, x_2, ..., x_t$
- hidden state ที่เวลา t จะใช้ตัวแปรว่า h_t

ถ้า input คือ ประโยคว่า “ฉันกินข้าว” เราก็จะอ่านประโยคนี้ทีละคำๆ ดังนั้น เราจะได้ว่า x_1 = “ฉัน”, x_2 =” กิน” และ x_3 =“ข้าว”



ภาพที่ 2.2.3 RNN

ที่มา: https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6

โดยที่

- H = hidden layer
- y_t = output จาก RNN ที่เวลา t
- x_t = input data ที่เวลา t
- h_t = hidden state ที่เวลา t

ขั้นตอนการทำงานของ RNN สามารถสังเกตได้จากรูปฝั่งขวามือ จะเห็นว่า Hidden state จะถูกนำไปเป็นส่วนนึงในการคิด y_t ใน timestep ต่อไปเสมอ state และเมื่อมาดูในส่วนของสมการ

- $h_t = f_h(U_h h_{t-1} + W_h x_t + b_h)$
- $y_t = f_y(W_y h_t + b_y)$

โดยที่

- f_h คือ activation function ของ hidden layer (เช่น tanh หรือ ReLU หรือ sigmoid function)
- f_y คือ activation function ของ output layer (เช่น softmax function)
- W_h คือ weight matrix ของ hidden layer
- U_h คือ hidden-state-to-hidden-state matrix (หรือ transition matrix)

จะเห็นว่าการที่จะคำนวณ hidden state ที่เวลา t ออกมาได้นั้น (h_t) ก็จะต้องใช้ 2 ตัวแปรสำคัญคือ hidden state ก่อนหน้า (h_{t-1}) และ input data ณ ตอนนั้น (x_t)

ปัญหาหลักของ RNN คือ Vanishing Gradient เพราะกว่าจะได้ output จะต้องผ่านมาหลาย timestep ซึ่งจากสูตร

$$\frac{\partial E}{\partial w} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial w}$$

E	คือ	ค่าความผิดพลาดทั้งหมด
o	คือ	ผลลัพธ์ที่ได้ในแต่ละโหนด
h	คือ	ผลรวมของค่า weight คูณกับ input แต่ละโหนด
w	คือ	ค่า weight

จะเห็นว่าการคำนวณ gradient ของ Loss E นั้นจะต้องอาศัยการคูณกันของ derivation หลายๆตัว นั้นหมายความว่าถ้า derivation มีค่าน้อยกว่า 1 การคูณกันหลายๆตัวแบบนี้จะทำให้ gradient มันจะลดลงไปเรื่อยๆตามแต่ละ timestep และสรุปได้ว่า RNN ยังไม่เหมาะกับข้อมูลที่มี timestep ยาวๆ

2.2.3 Long-Short Term Memory (LSTM)

LSTM หรือ (Long Short-Term Memory) ถูกสร้างมาเพื่อแก้ปัญหาของ RNN ที่มีต่อ sequence ยาวๆ โดย LSTM นั้นจะมี memory ที่อยู่ภายในโครงสร้างของ neural network เพียงแต่ memory ของ LSTM สามารถกำหนดได้ว่าจะ Write Read หรือ Forget ซึ่งจะสังเกตได้ว่ามีลักษณะการทำงานคล้าย memory ของ computer เพียงแต่ว่า memory ใน LSTM มีลักษณะเป็น analog เท่านั้นเอง
องค์ประกอบหลักๆของ LSTM ที่ควรรู้จักคือ

- Cell state เป็นตัวเก็บ state ของ memory
- Gate เป็นตัวควบคุมการไหลของข้อมูล ซึ่งก็คือค่า analog ที่คอยควบคุมการ read, write และ forget

จริงๆหลักการของ gate จะคล้ายกับ node ใน neural network ที่ดูว่าค่าความแรงของสัญญาณที่เข้ามานั้น active หรือไม่ คราวนี้มาดูกระบวนการต่างๆ ของ LSTM

2.2.3.1 Forget

Forget คือกระบวนการหนึ่งที่ตัดสินใจว่า cell จะถูกลบหรือไม่ โดยส่วนที่ตัดสินใจคือ forget gate ส่วนวิธีการตัดสินใจนั้นจะได้จากสมการนี้

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

จากสมการจะเห็นได้ว่าเราจะนำ input (x_t) และ previous hidden state (h_{t-1}) มาคิดโดยใช้ sigmoid function เป็นตัวตัดสิน (sigmoid function จะทำหน้าที่คืนค่าออกมาเป็น 0 หรือ 1 เท่านั้น) ถ้า $f_t = 1$ หมายความว่า ยังเก็บข้อมูลไว้ใน cell แต่หาก $f_t = 0$ นั่นหมายความว่าจะทำการลบ cell นั้นออกไป

2.2.3.2 Write

เมื่อมี input data เข้ามา มี 2 process ที่จะเกิดขึ้นคือ

2.2.3.2.1 Update cell state ด้วย input data หรือไม่

2.2.3.2.2 ถ้า update จะ update ด้วยค่าอะไร

เริ่มที่ 2.2.3.2.1 ก่อน วิธีการคิดจะคิดตามสมการนี้

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

โดยจากสมการนี้จะเห็นได้ว่าจะนำ input data (x_t) และ previous hidden state (h_{t-1}) มาคำนวณด้วยเช่นกัน หลังจากนั้นก็ครอบได้ sigmoid function เพื่อให้ค่าที่ได้ออกมาเป็น 0 หรือ 1 เท่านั้น

มาในส่วนของ สมการของหัวข้อที่ 2.2.3.2.2 คือ

$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

สมการนี้มีชื่อว่า input modulation gate โดยจะใช้ input data (x_t) กับ previous hidden state (h_{t-1}) โดยครั้งนี้จะนำ tanh function มาครอบแทน tanh function จะทำให้ค่า output ที่ออกมาอยู่ในช่วง -1 ถึง 1 จะมองว่าผลลัพธ์ที่ได้จากสมการเป็น cell state candidate ก็ได้

2.2.3.3 Update cell state

หลังจากที่ได้ค่า forget gate, input gate และ input modulation gate ซึ่งก็เพียงพอต่อการ update cell แล้ว คราวนี้เราจะนำทุกอย่างมารวมกัน

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

จากสมการจะเห็นได้ f_t (forget gate) จะเป็นตัวบ่งบอกว่าจะนำ c_{t-1} (previous cell state) มาคิดหรือไม่ ส่วนอีก expression จะเห็นว่า i_t (input gate) จะเป็นตัวตัดสินว่าจะนำ g_t (input modulation gate) มาคิดหรือไม่ ถ้าค่า f_t เป็น 1 นั้นหมายความว่า เราจะเก็บค่า c_{t-1} ไว้คิดด้วย หรือถ้าค่า i_t เป็น 1 ก็จะใช้ค่า g_t update เลย

2.2.3.4 Read

การ read ในนี้ ถ้าจะตีความให้ถูกคือการอนุญาตให้การทำงานข้างนอกมา read ตัว h_t จาก 2 สูตรคือ

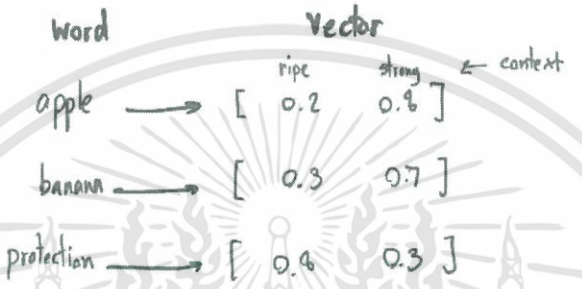
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$h_t = o_t \odot tanh(c_t)$$

โดยสมการแรกคือสมการในการคำนวณการอนุมัติ จากสมการจะเห็นได้ว่า เราจะนำ input data (x_t) และ previous hidden state (h_{t-1}) มาคำนวณและทำการครอบด้วย sigmoid function เพราะจะได้ output ออกมาเป็นแค่ 0 กับ 1 หลังจากนั้น นำไปเข้าสมการที่ 2 จะพบว่า o_t เป็นตัวกำหนดว่าจะให้คนอื่นสามารถ read h_t ของเราได้ไหม ถ้า o_t ของเราเป็น 0 ก็จะไม่สามารถอ่านได้นั่นเอง

2.2.4 Word2Vec

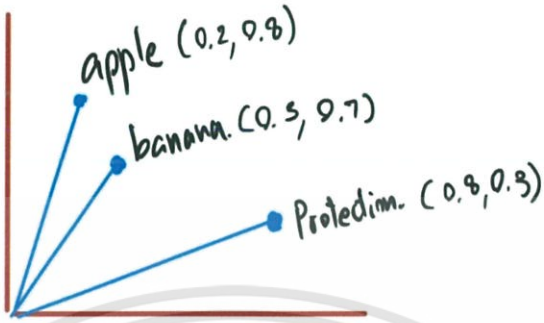
Word2Vec คือ โมเดลที่ใช้สร้าง word embedding พัฒนาโดยทีมนักวิจัยของ Google นำโดย Tomas Mikolov ซึ่งโมเดลนี้สามารถทำงานได้ดีกว่าวิธีแบบเดิม ๆ (Latent Semantic Analysis) ซึ่ง word2vec ก็คือการแสดง “คำ” ให้อยู่ในรูปของ “vector” นั่นแหละ แต่จะไม่ได้ใช้ one-hot encoding ในการสร้างตัวเลข vector แบบเดิมแล้ว word2vec จะใช้วิธีการคำนวนตัวเลขของคำนั้น ๆ จาก context รอบๆ คำนั้น (ไอเดียมาจาก language model)



ภาพที่ 2.2.4 ตัวอย่างการแปลงจาก Word ไปเป็น Vector

ที่มา: https://lukkiddd.com/word-embedding-และ -word2vec-คืออะไร-e60bdf6d78d3

และถ้าหากเราเอา vector เหล่านั้นมา plot ก็จะได้ดังนี้



ภาพที่ 2.2.5 นำ vector ที่ได้มา plot

ที่มา: https://lukkiddd.com/word-embedding-และ -word2vec-คืออะไร-e60bdf6d78d3

แกน y(แนวตั้ง) คือคำว่า ripe ส่วนแกน x (แนวนอน) คือคำว่า strong และเนื่องจากการนำ vector 2 ตัวมาคูณกันในรูปแบบสเกลาร์ (dot product) กัน มันคือการหาค่าความคล้ายกันของ 2 vector หรือที่เรียกว่า PMI(point-wise mutual information) ดังนั้นเราเลยสามารถหาค่าความคล้ายกันของคำ(Word similarity) ได้ด้วยวิธีนี้

$$\hat{apple} \cdot \hat{banana} = (0.2 \times 0.3) + (0.8 \times 0.7)$$
$$= 0.06 + 0.56$$
$$= 0.62$$

$$\hat{apple} \cdot \hat{protectim} = (0.2 \times 0.8) + (0.8 \times 0.3)$$
$$= 0.16 + 0.24$$
$$= 0.4$$

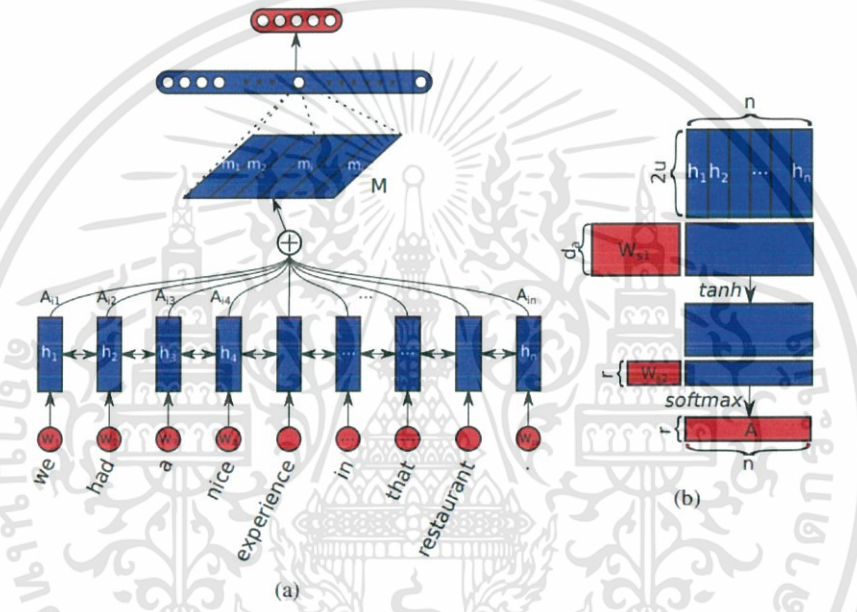
ภาพที่ 2.2.6 สมการการหาความคล้าย

ที่มา: https://lukkiddd.com/word-embedding-และ -word2vec-คืออะไร-e60bdf6d78d3

2.3. งานวิจัยที่เกี่ยวข้อง

2.3.1 A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING

เป็นงานวิจัยที่พูดถึง model ใหม่ที่ใช้สำหรับ extract ความสำคัญของข้อความโดยใช้ Self-attention ในตัวงานวิจัยจะใช้ 2-D matrix ในการ embedding โดยในแต่ละ row ของ matrix แสดงถึง attending ในแต่ละส่วนของประโยค และในงานวิจัยยังพูดกลไกการทำงานของ self-attention และ special regularization สำหรับ model ไว้อีกด้วย และยังสามารถ visualize เพื่อให้คนทั่วไปสามารถเข้าใจ output สุดท้ายของ model ได้อย่างไม่ยาก



ภาพที่ 2.3.1 Model Structure

ที่มา: https://arxiv.org/pdf/1703.03130.pdf

ภาพที่ 2.3.1 คือภาพที่แสดงให้เห็นถึงโครงสร้างของการรวมกันระหว่าง sentence embedding model เข้ากับ softmax layer เพื่อทำ sentiment analysis

(a) Sentence embedding (M) ถูกคำนวณจากผลรวมของ weight ของ hidden state จาก bidirectional LSTM (h_1, ..., h_n) โดยที่ ผลรวมของ weight ที่จะทำการจะถูกแสดงอยู่ในรูปของ A_1, ..., A_n

(b) กล่องสีฟ้าจะแสดงถึง hidden state และกล่องสีแดงจะแสดงถึง weight, annotation, input/output

โครงสร้าง Model จะประกอบไปด้วย 2 ส่วน คือ LSTM และ Self-Attention mechanism โดยส่วนของ Self-Attention mechanism เป็นส่วนที่ส่งชุดผลบวกของ weight vector ของ Hidden state ใน LSTM (summation of weight vector) ซึ่งเราสามารถ อธิบายภาพได้ดังนี้

สมมติเรามีประโยคหนึ่ง ที่จะคำเท่ากับ n คำ (token)

$$S = (w_1, w_2, w_3, \ldots, w_n)$$

- w_i คือ vector ที่มีจำนวน word embedding dimension เท่ากับ d สำหรับคำที่ *i*-th
- S คือลำดับของ 2-D matrix ของ word embedding ทั้งหมด

คราวนี้ S มีความเป็นอิสระต่อกันมากจนเกินไป เพื่อให้มีความเกี่ยวข้องกับคำข้างๆ เราจึงต้องนำ LSTM เข้ามาใช้

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, \overrightarrow{h_{t-1}})$$

โดย h_t คือ hidden state และเพื่อความง่ายเราจะทำการรวบ h_t ให้อยู่ในรูป H

$$H = (h_1, h_2, \ldots, h_n)$$

จุดประสงค์เราคือการ encode ประโยคที่มีหลากหลายความยาวไปเป็นขนาดท่กำหนด โดยเราจะให้ input เป็น LSTM hidden state (H) ทั้งหมด และให้ output (a) เป็น vector ของ weight

$$a = softmax(w_{s2} tanh(w_{s1} H^t))$$

- W_{s1} คือ matrix weight ที่ลักษณะเป็น d_a-by-2u
- W_{s2} คือ vector ของ parameter size d_a
- d_a คือ hyperparameter ที่เราสามารถ set ได้ตามใจชอบ
- tanh คือการทำให้ผลลัพธ์ที่ได้จาก $w_{s1}H^t$ มีค่าอยู่ในช่วง (-1, 1)
- Softmax ทำหน้าที่ทำให้ weight ที่คำนวณได้ทั้งหมดรวมกันจะได้ 1

Vector แสดงถึงการให้น้ำหนักกับองค์ประกอบเฉพาะของประโยค เช่น ชุดคำพิเศษหรือวลีที่เกี่ยวข้อง ดังนั้นจึงคาดว่า vector จะทำให้เห็นถึงลักษณะหรือองค์ประกอบของความหมายในประโยคได้ แต่ถึงอย่างไร ในประโยคนึงอาจมีหลายองค์ประกอบโดยเฉพาะประโยคยาวๆ (ตัวอย่างประโยคที่ถูกเชื่อมด้วย “และ”) ดังนั้นเพื่อแสดงความหมายโดยรวมของประโยคเราจึงต้องมีตัวที่สร้างความมุ่งเน้นส่วนต่างๆของประโยค ดังนั้นเราจึงต้องมีการดำเนินการหลาย hops ของ attention สมมติว่าเราต้องแยกส่วนต่างๆ

ออกจากประโยค เราต้องขยาย W_{s2} ลงใน matrix r-by-d_a W_{s2} เป็น vector ที่เป็นผลลัพธ์ที่จะกลายมาเป็นคำอธิบายประกอบของ matrix A

$$A = softmax(w_{s2}tanh(w_{s1}H^t))$$

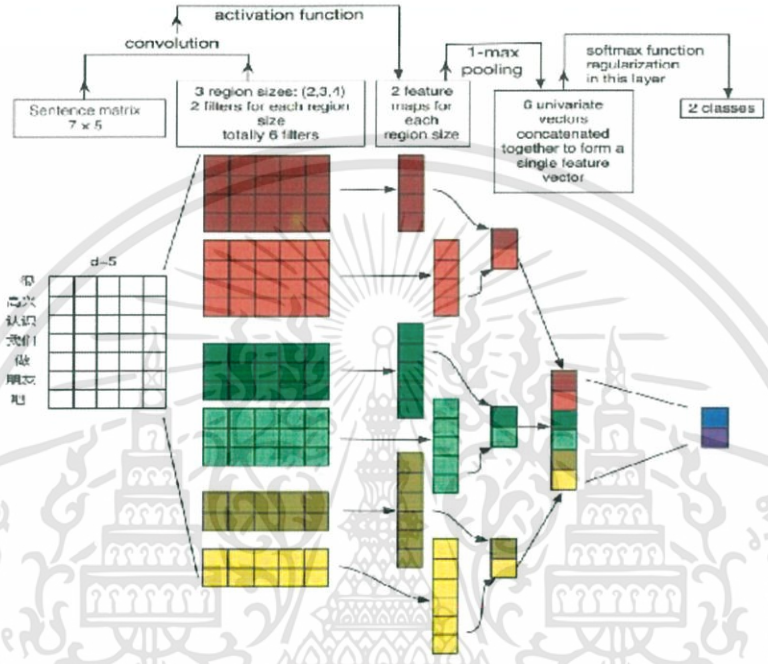
Sentence embedding is:

$$M = AH$$

- M คือ sentence embedding ซึ่งเป็น matrix ที่มีขนาดเท่ากับ matrix ของ A คูณ กับ matrix ของ H
- A คือ Attention เป็นผลลัพธ์ที่เกิดจากสมการข้างต้น โดย A มีลักษณะเป็น matrix
- H คือ Hidden state ของ LSTM

2.3.2 Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus

เป็นงานวิจัยที่ทำเกี่ยวกับการทำ Sentiment analysis โดยใช้ Convolutional Neural Network (CNNs) ซึ่งงานวิจัยได้ผลการทดลองมากมายรวมทั้ง Dataset ที่ทางวิจัยได้จัดทำขึ้นซึ่งงานวิจัยนี้ได้มีการ Public dataset ไว้ที่ Github ทั้งหมด 2 ชุดคือ Sentiment_XS_30k.txt และ Sentiment_XS_test.txt

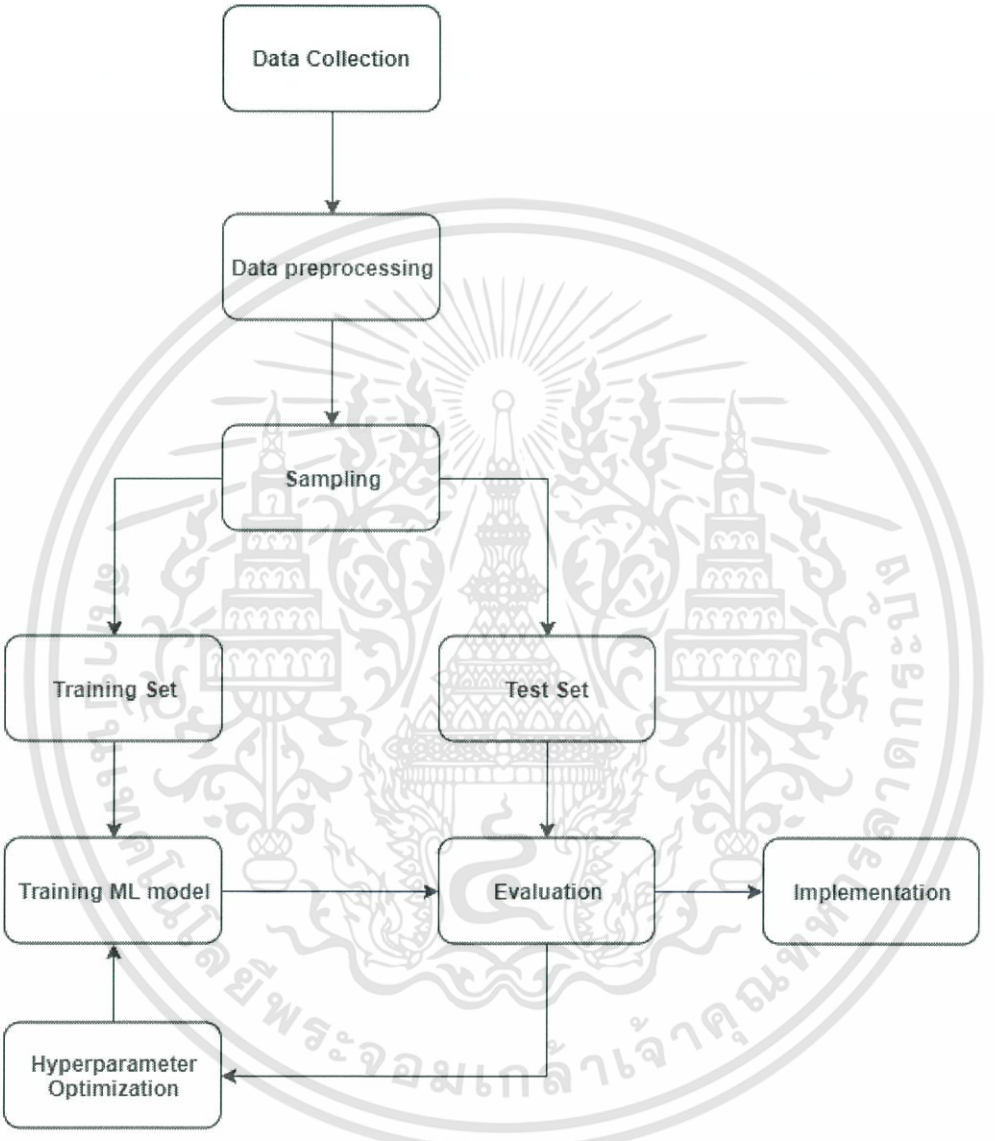


ภาพที่ 2.3.2 Convolutional neural network

ที่มา: https://www.semanticscholar.org/paper/Sentiment-Classification-with-Convolutional-Neural-Zhang-Chen/f5fa0bacbe1ba7d372a2940d8f139ef6531c39d9/figure/0

บทที่ 3
วิธีดำเนินการวิจัย

3.1. วิธีการดำเนินการวิจัย



ภาพที่ 3.1.1 ภาพรวมวิธีการดำเนินงานวิจัย

3.1.1 การเก็บรวบรวมข้อมูล (Data Collection)

กระบวนการนี้เราจะทำการหา Dataset ที่มีความเป็นกลางที่สุด ไม่เอนเอียงไปทาง Domain ใดจนเกินไป จึงพยายามหา dataset ประเภท conversation มาทดลองกับงานนี้โดย dataset ที่ใช้ในการทำงานนี้จะถูกแบ่งเป็น 2 ประเภทดังนี้

3.1.1.1 Dataset ภาษาจีน

โดย Dataset ภาษาจีนเป็น Data ที่มีชื่อว่า Chinese conversation sentiment ซึ่งสามารถโหลดได้ จากทางhttps://github.com/z17176/Chinese_conversation_sentiment ซึ่งเป็น Public dataset ที่อยู่บน Github โดยมีลักษณะดังนี้

```
labels,text
positive,奖励 就是 亲
positive,谢谢 妹妹 加 朋友
positive,喜欢 吃 南浮
positive,好 吧 小 很 强悍 怕 不
negative,他 女朋友 旁边 所以 不 方便 跟 说话
positive,讲 的话 真 幽默
negative,真的 那 好 心疼
positive,无所不能 可是 刚才 差点 要 上 百度
positive,就是 漂亮 嘛 完美
negative,跟 一个 低级 计算机系统 说话 要 那么 讲究
negative,不行 必须 起
positive,这么 丑 还 漂亮
negative,一点 也 不好 没有 人 懂 知道
positive,噢 咋 设置 美女
positive,石家庄 五星级 酒店
positive,机器人 怎么 可能 忙
positive,哪有 这是 微笑 微笑
negative,无聊 阿肿 办
negative,无语 看来 白说
negative,快点 别 废话 多
```

ภาพที่ 3.1.2 Dataset ภาษาจีน

โดยวิธีการเก็บข้อมูล Dataset ภาษาจีนในระบบนั้น จะทำการเก็บอยู่ในรูป text file โดยสามารถเรียกใช้งานผ่าน Python ได้

3.1.1.2 Dataset ภาษาอังกฤษ (Dataset ภาษาอื่น)

ในส่วนของ Dataset ภาษาอื่นเราเลือก twitter Airline Sentiment มาใช้ในการทดสอบครั้งนี้ ซึ่งเป็น Dataset ภาษาอังกฤษที่ปล่อย Public ฟรีอยู่ใน Kaggle

(https://www.kaggle.com/crowdflower/twitter-airline-sentiment) โดยมีลักษณะดังนี้

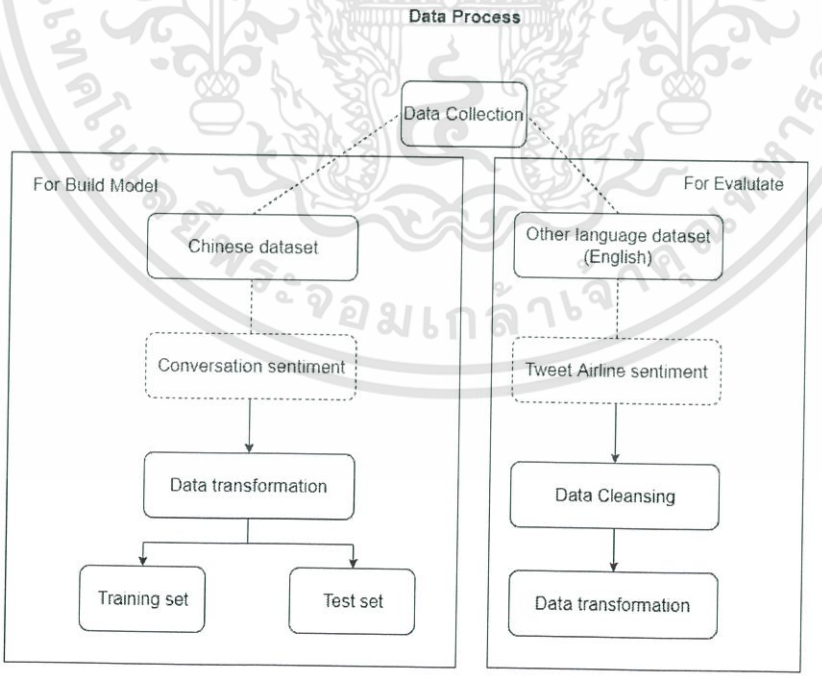
airline_se	airline_se	negativeri	negativeri	airline	airline_se	name	negativeri	retweet_c	text	tweet_co	tweet_cre	tweet_loc	us
neutral	1			Virgin America		cairdin		0	What @dhepburn sa		########		Ea
positive	0.3486		0	Virgin America		jnardino		0	plus you've added c		########		Pa
neutral	0.6837			Virgin America		yvonnalynn		0	I didn't today... Mus		########	Lets Play	Ce
negative	1	Bad Flight	0.7033	Virgin America		jnardino		0	it's really aggressive		########		Pa
negative	1	Can't Tell	1	Virgin America		jnardino		0	and it's a really big b		########		Pa
negative	1	Can't Tell	0.6842	Virgin America		jnardino		0	seriously		########		Pa
positive	0.6745		0	Virgin America		cjmcginnis		0	yes, nearly every tir		########	San Franci	Pa

ภาพที่ 3.1.3 Dataset ภาษาอังกฤษ

โดย Dataset นี้จะถูกเก็บอยู่ในรูปของ Excel file ซึ่งสามารถเรียกใช้งานได้ผ่าน Python โดยใช้ Pandas ซึ่งเป็น library ที่สามารถจัดการไฟล์ Excel บน Python ได้

3.1.2 การเตรียมข้อมูลก่อนเข้ากระบวนการ (Data preprocessing)

ก่อนที่จะนำข้อมูลต่างๆไปใช้งาน จำเป็นจะต้องมีการเตรียมความพร้อมให้ข้อมูลเหล่านั้น ในขั้นตอน data preprocessing นี้จะถูกแบ่งออกเป็น 2 ส่วนหลักๆคือ



ภาพที่ 3.1.4 กระบวนการต่างๆของการจัดการ Data

3.1.2.1 Data Cleansing

Dataset ภาษาจีน: ไม่มีการทำ Cleansing เนื่องจาก dataset ที่ได้มา ได้ทำการ Cleansing ไว้แล้ว

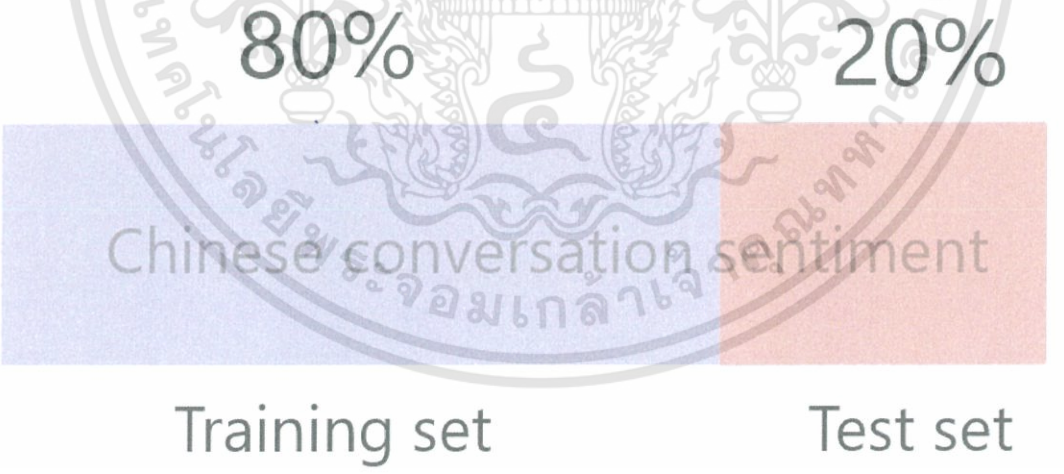
Dataset ภาษาอังกฤษ: dataset ที่ได้มาเป็น Twitter Airline Sentiment จะมีการจัดการดังนี้

- เลือก Column ที่เป็น Sentiment และ Text เท่านั้น
- ลบ Name Entity ออกจาก Text เพื่อให้เหลือแต่เพียงคำที่เกี่ยวข้องกับ Sentiment เท่านั้น
- ทำการลบ record ที่มี Sentiment เป็น neutral ออกเนื่องจาก Model จะสามารถทำนายได้แต่ค่า positiveและ negative เท่านั้น

3.1.3 การแบ่งข้อมูล (Data Sampling)

Dataset ภาษาจีน: ทำการแบ่ง Dataset ภาษาจีนออกเป็น 2 ส่วนด้วยกันคือ Training set และ Test set โดยจะแบ่งตามสัดส่วน 80/20 ซึ่งเป็นสัดส่วนเริ่มต้น ที่งานวิจัยต่างๆนิยมใช้กัน โดย Training set จะเป็น Dataset ที่ใช้สำหรับการฝึกฝน Model และ Test set จะเป็น Dataset ที่ใช้สำหรับการวัดผล Model

Dataset ภาษาอังกฤษ: สำหรับ Dataset ภาษาอังกฤษจะไม่มีการแบ่งเนื่องจะถูกนำมาเป็นตัววัดผลของ Model



ภาพที่ 3.1.5 สัดส่วนของ training set ต่อ test set

3.1.4 การฝึกฝน Model (Training Model)

Model คือกระบวนการ Deep learning ที่สามารถจำแนก Sentiment ของประโยคที่ใส่เข้าไปได้ โดย Model จะสามารถทำหน้าที่ได้ดีขึ้นหากมีการฝึกฝนบ่อยๆ ซึ่งสามารถสังเกตได้จากค่า Loss ที่ลดน้อยลง ในขั้นตอนนี้ เราจะนำ Dataset ภาษาจีนที่เตรียมไว้ มาเป็นตัวฝึกฝนของ Model โดยผู้จัดทำได้ทำการฝึกฝน Model ทั้งหมด 100 รอบ (epoch) และทำการเก็บความสามารถของ Model ในแต่ละรอบไว้เพื่อนำไปวัดผลในลำดับถัดไป

3.1.5 การวัดผล (Evaluation)

สำหรับขั้นตอนการวัดผลนี้จะนำ Model ที่ผ่านการฝึกฝนมาแล้ว มาวัดความสามารถ ซึ่งรายละเอียดจะถูกอธิบายไว้ในบทที่ 4 ทั้งหมด

3.1.6 การปรับแต่ง Parameter (Hyperparameter Optimization)

สำหรับขั้นตอนนี้เป็นขั้นตอนที่จะต้องการเพิ่มประสิทธิภาพให้กับ Model โดยปรับแต่ง parameter ที่เกี่ยวข้องทั้งหมด โดย parameter ที่เกี่ยวข้องมีดังนี้

1. Epoch : จำนวนรอบที่ทำการ train
2. Learning rate : ความเร็วในการเรียนรู้ของ Model
3. Batch size : จำนวนที่จะนำเข้าไป train ในแต่ละครั้ง
4. LSTM dimension : จำนวน dimension ใน LSTM
5. Attention dimension : จำนวน dimension ใน Attention
6. Attention hops : จำนวน Hops ของ Attention
7. Penalization Coefficient : ค่าสัมประสิทธิ์ของ Penalization

โดยกระบวนการปรับแต่ง Parameters นั้นเราได้ทดลองปรับแต่งไปเรื่อยๆเพื่อหา Model ที่มี Accuracy สูงสุด และนี้คือผลสรุปของ parameters ทั้งหมด

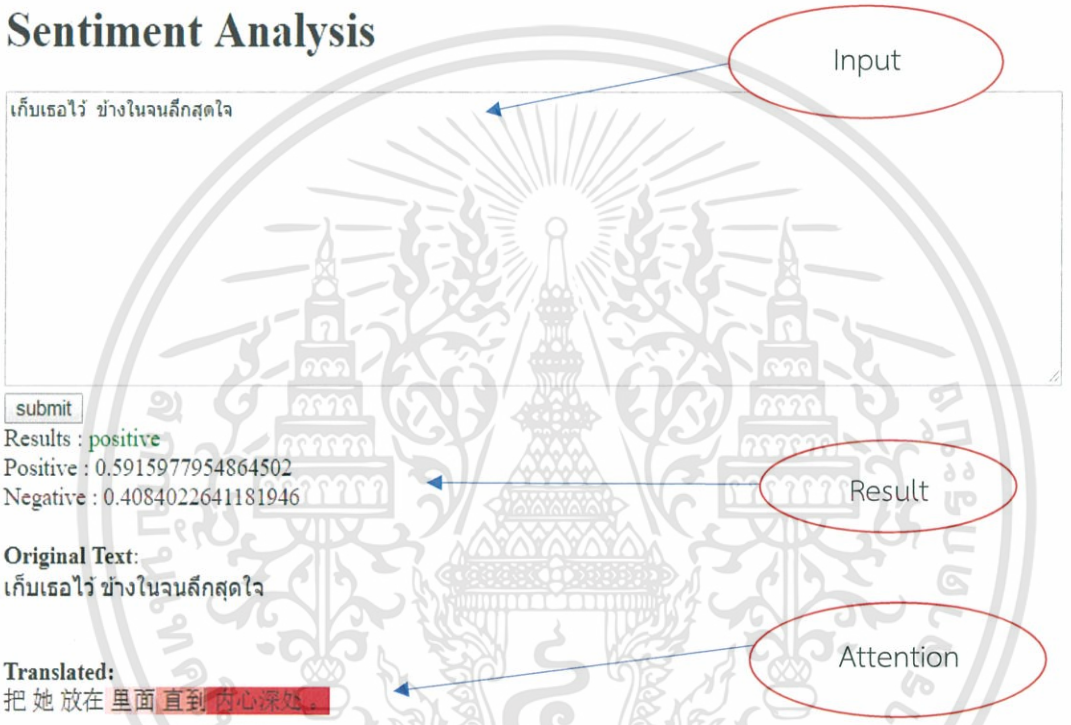
ตาราง 3.1.1 ตาราง parameter

ชื่อ Parameter	ค่าของ parameter
LSTM dimension	64
Batch Size	32
Attention dimension	100
attention hops	10
Penalization coefficient (C)	0.03
Optimizer	Adam
learning rate	1.00E-03

3.1.7 การนำไปใช้งาน (Implementation)

นำ Model ที่ผ่านการวัดผลมาแล้ว ไปทำเป็นตัว Application ซึ่งในตัว Application จะถูกสร้างขึ้นโดย Flask ซึ่งเป็น framework สำหรับพัฒนาเว็บของ python ในส่วนของตัวเว็บมีการใช้งานดังนี้

1. Input คือส่วนที่คอยรับ text จาก User
2. Result คือส่วนที่คอยแสดง Sentiment score
3. Attention คือส่วนแสดง Attention ของประโยค



ภาพที่ 3.1.6 ตัวอย่าง Web application

3.2. ขั้นตอนการทำงานของระบบ

ภาพที่ 3.2.1 ภาพรวมของระบบ

3.2.1 รับ Input (text)

การรับ Input ของระบบนี้ จะรับเป็นข้อความหรือประโยคที่มีความยาวไม่เกิน 500 คำได้ ซึ่งประโยคสามารถเป็นภาษาอะไรก็ได้ ตัวอย่างเช่น

- ฉันชอบรถสีแดงคันนี้
- This brand is awesome.
- Te odio

3.2.2 Translation

นำ input ที่ได้มา translate ให้เป็นภาษาจีนตัวย่อ (Simplified Chinese) โดยขั้นตอนนี้ ทางผู้จัดทำได้ใช้ Google translate API มาเป็นตัว translate ซึ่งจะได้ผลลัพธ์ดังนี้

- ฉันชอบรถสีแดงคันนี้ = 我喜欢这辆红色轿车
- This brand is awesome. = 这个品牌很棒
- Te odio = 我讨厌你

```python
# Imports the Google Cloud client library
from google.cloud import translate

# Instantiates a client
translate_client = translate.Client()

# The text to translate
text = u'Hello, world!'
# The target language
target = 'ru'

# Translates some text into Russian
translation = translate_client.translate(
    text,
    target_language=target)

print(u'Text: {}'.format(text))
print(u'Translation: {}'.format(translation['translatedText']))
```

ภาพที่ 3.2.2 ตัวอย่างการใช้งาน Google translate API

3.2.3 Word segmentation

นำประโยคที่ผ่านการ translate มาทำการตัดคำ ซึ่งทางผู้จัดทำได้ใช้ Jieba ซึ่งเป็น Library ของ Python ที่สามารถตัดคำภาษาจีนได้โดยจะได้ผลลัพธ์ดังนี้

- 我喜欢这辆红色轿车 = 我 / 喜欢 / 这辆 / 红色 / 轿车
- 这个品牌很棒 = 这个 / 品牌 / 很棒
- 我讨厌你 = 我 / 讨厌 / 你

```
import jieba

seg_list = jieba.cut("我喜欢这辆红色轿车", cut_all=True)
print(" / ".join(seg_list))  # 精确模式

seg_list = jieba.cut("这个品牌很棒", cut_all=True)
print(" / ".join(seg_list))  # 精确模式


seg_list = jieba.cut("我讨厌你", cut_all=True)
print(" / ".join(seg_list))  # 精确模式
```

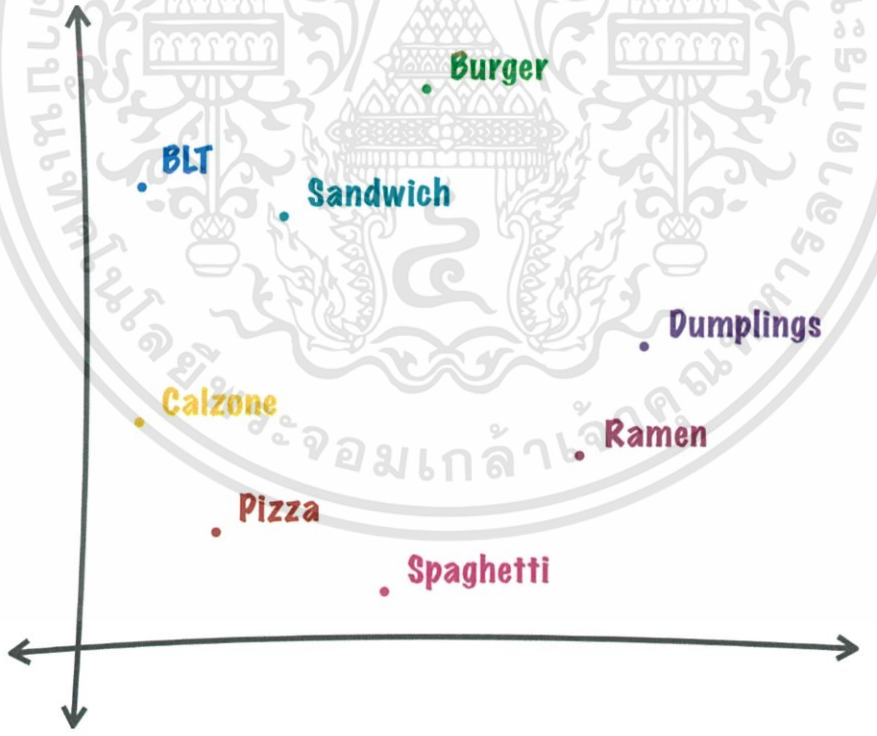
```
我 / 喜欢 / 这辆 / 红色 / 轿车
这个 / 品牌 / 很棒
我 / 讨厌 / 你
```

ภาพที่ 3.2.3 ตัวอย่างการตัดคำภาษาจีนโดยใช้ jieba

3.2.4 Word embedding

ขั้นตอนนี้มีจุดประสงค์เพื่อแปลงคำให้อยู่ในรูปของ Vector เพื่อทำให้คอมพิวเตอร์เข้าใจในคำๆนั้นได้ โดยทางผู้จัดทำได้ใช้ Word2Vec ซึ่งเป็นวิธีการนึงในการทำ Word Embedding และมี Library ของ Python ที่สามารถอำนวยความสะดวกในการทำ Word2Vec ได้ โดยมีชื่อว่า Gensim ซึ่งสามารถอธิบายได้ดังภาพต่อไปนี้

ภาพที่ 3.2.4 วิธีการใช้งาน Word2Vec

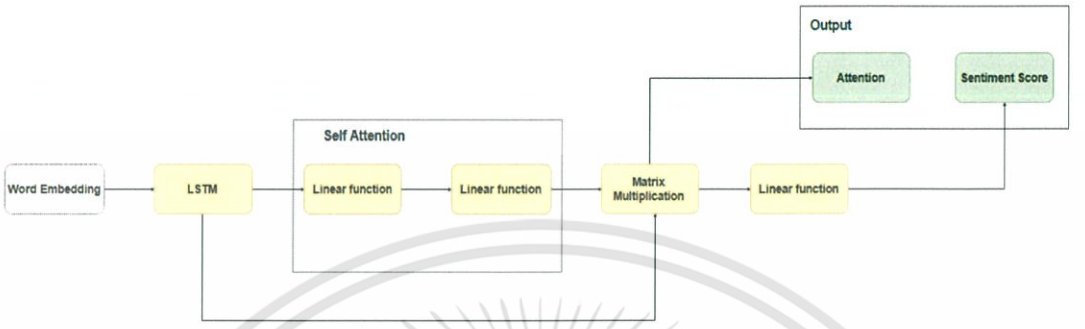


ภาพที่ 3.2.5 ตัวอย่าง word2vec

ที่มา : https://medium.com/square-corner-blog/caviars-word2vec-tagging-for-menu-item-recommendations-13f63d7f09d8

3.2.5 Model

ดั่งที่กล่าวไว้ในหัวข้อ 3.1.4 ว่า Model คือกระบวนการฝึกฝน (learning) ที่สามารถจำแนก Sentiment ของประโยคที่ใส่เข้าไปได้ โดยโครงสร้างของ Model จะมีลักษณะดังนี้



ภาพที่ 3.2.6 โครงสร้าง Model

3.2.6 Output

Output คือผลลัพธ์ที่ได้ออกมาจาก Model ซึ่งประกอบไปด้วย 2 ส่วนดังนี้

1. Sentiment score

   ในส่วนของ Sentiment score จะถูกแบ่งเป็นเป็น 2 ประเภทคือ

   - Positive probability คือค่าความเป็นบวกโดยมีค่าอยู่ในช่วง 0 ถึง 1
   - Negative probability คือค่าความเป็นลบโดยมีค่าอยู่ในช่วง 0 ถึง 1

   ตัวอย่าง

   positive : 0.11

   negative : 0.89

2. Attention คือน้ำหนักของแต่ละคำ ซึ่งเราจะนำไปใช้ในการทำ Visualize ได้ดังนี้

ภาพที่ 3.2.7 ภาพตัวอย่าง Attention

บทที่ 4
ผลการวิจัย

4.1. วิธีการวัดผล

สำหรับวิธีการวัดผลเราจะใช้หลัก Confusion matrix ในการวัดผลซึ่งเป็นไปตามสมการดังนี้

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1- = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

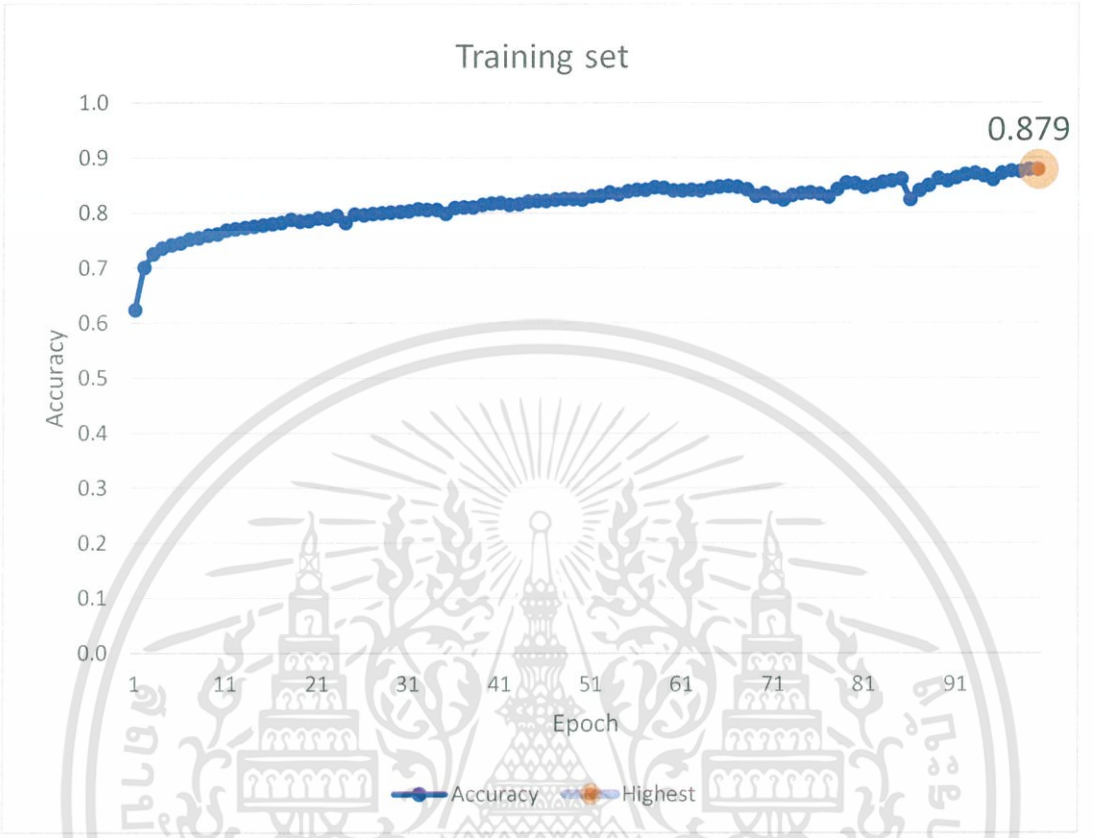
และได้ผลลัพธ์ดังนี้

ตาราง 4.1.1 Precision, Recall, F1-score

	Precision	Recall	F1-score	Support
Positive	0.84	0.78	0.81	7820
Negative	0.37	0.45	0.41	2180
Average / Total	0.73	0.71	0.72	10000

4.2. วัดผลกับภาษาเดียวกัน

วัดผลกับชุดข้อมูลที่เป็น Training set ทั้งหมด 100 epoch จะได้ผลลัพธ์ของ Accuracy ดังนี้

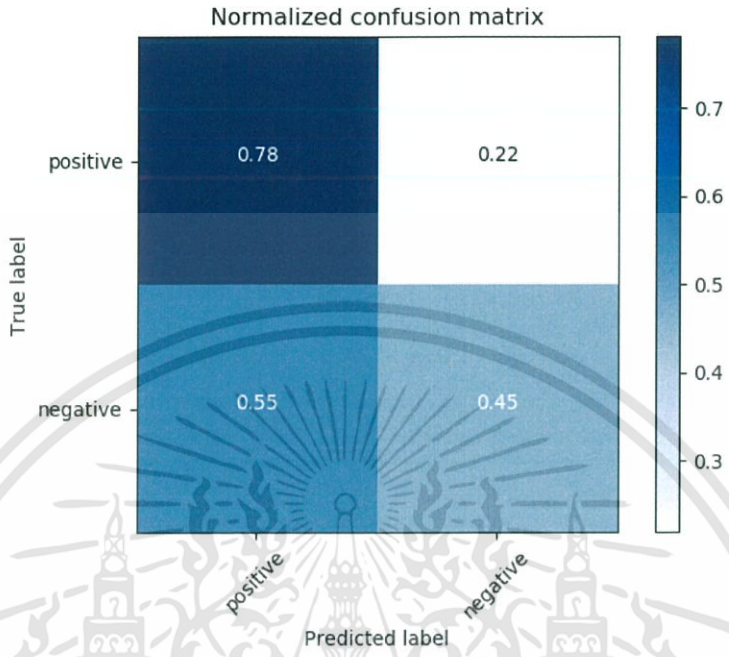


ภาพที่ 4.2.1 Evaluate with training set Graph

จากภาพที่ 4.2.1 คือการ train model ทั้งหมด 100 รอบ (epoch) โดยทุกครั้งที่รอบจะถูกนำมาวัดความแม่นยำ (accuracy) กับชุด training set เพื่อศึกษาความสัมพันธ์ระหว่างจำนวน (epoch) ต่อความแม่นยำ ซึ่งจะแสดงให้เห็นว่า model มี accuracy ที่วิ่งเข้าหา 1 ซึ่งมีความเป็นไปได้ว่า ถ้ายิ่งใช้รอบที่ train (epoch) มากขึ้นเท่าไหร่ ยิ่งมีความ overfit กับข้อมูลที่นำมา train ด้วยเท่านั้น

4.3. วัดผลกับภาษาอื่น

สามารถสรุปเป็น Confusion matrix ได้ดังนี้



ภาพที่ 4.3.1 confusion matrix with normalized

ภาพที่ 4.3.1 และ 4.3.2 คือภาพที่แสดงการวัดผลระหว่าง Twitter Airline sentiment (English) กับ model ซึ่งภาพที่ 4.3.1 คือภาพ Confusion matrix แบบ normalized ที่สามารถสรุปได้ว่า model นี้สามารถทำงานได้ดีกับประโยคที่เป็น positive ใน Twitter Airline sentiment เพราะสามารถสร้าง accuracy ได้ถึง 0.78 หรือ 78% จากประโยคทั้งหมด

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

สรุปผลการวิจัย

การทำ Sentiment Analysis ข้ามภาษา ด้วย Machine Learning โดยมี Dataset ที่ใช้สำหรับการ Train เป็น “ประโยคพูดคุยในภาษาจีน (Chinese Conversation)” และใช้ Google Translate เป็นตัวกลางสำหรับการแปลภาษาอื่นๆ มาเป็นภาษาจีน สามารถสร้างแม่นยำในชุด Training set ได้ถึง 82.5% และมีความแม่นยำในชุด Test set ถึง 80.4% ซึ่งถือเป็นความแม่นยำที่ผู้ใช้พอใจ และเมื่อนำไปใช้กับ Twitter Airline sentiment ซึ่งมีจำนวน 1 หมื่นประโยค สามารถสร้างความแม่นยำได้สูงถึง 70.88%

อุปสรรค

1) Machine learning เป็นความรู้ใหม่สำหรับผู้จัดทำ จึงทำให้การเริ่มต้นของ Project ค่อนข้างใช้เวลา ในการทำความเข้าใจ Machine learning เบื้องต้น
2) การศึกษาเกี่ยวกับ Pytorch ซึ่งเป็น Tool ที่ใช้ในการทำ Machine learning ใช้เวลาค่อนข้างนานในการเริ่มต้นเพราะมีรายเอียดที่เยอะ และผู้จัดไม่ทำไม่เคยใช้มาก่อน
3) การหา Dataset ใย Domain ที่ต้องการค่อนข้างยาก
4) การหา Neural network ให้กับ Model ที่จะสร้างความแม่นยำให้กับ Model ได้ดีใช้เวลาในค้นหาสูง เนื่องจากต้องอ่าน Paper ต่างๆและต้องใช้เวลาทำความเข้าใจนาน
5) ใช้เวลา Train Model นาน

ข้อเสนอแนะ

1) ควรเลือก dataset ให้ตรงกับจุดประสงค์ที่ตั้งไว้ เพราะจะทำให้เกิดความแม่นยำใน Model สูงกว่าการนำไปวัดผลข้าม Domain เช่น ไม่ควรนำ Model รีวิวอาหารไปทำนายการรีวิวหนัง เป็นต้น
2) ควรศึกษาพื้นฐานของ Machine learning ให้เข้าใจก่อน เพราะโครงสร้างของ Machine learning มีความซับซ้อนและมีรูปแบบที่หลากหลาย สามารถนำไปปรับใช้ให้เหมาะสมกับรูปแบบต่างๆได้

บรรณานุกรม

1. **Long Short-Term Memory (LSTM)**, Retrieve from : https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6
2. **ARTIFICIAL NEURAL NETWORKS**, Retrieve from http://shodhganga.inflibnet.ac.in/bitstream/10603/48/6/chaper%204_c%20b%20bangal.pdf
3. **Attention Is All You Need.**, Retrieve from https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
4. **Flask คืออะไร**, Retrieve from https://saixiii.com/python-flask-web-application/
5. **A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING**, Retrieve from : https://arxiv.org/pdf/1703.03130.pdf
6. **Pytorch เบื้องต้นบทที่ ๑**: บทนำ, Retrieve from https://phyblas.hinaboshi.com/tomoshibi01
7. **Activation function**, Retrieve from https://en.wikipedia.org/wiki/Activation_function
8. **แนะนำเกี่ยวกับ Jupyter Notebook เพื่อใช้ในการทำ Data Science**, Retrieve from http://www.mindphp.com/forums/viewtopic.php?f=144&t=47184
9. **Git คืออะไร**, Retrieve from https://medium.com/@pakin/git-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-git-is-your-friend-c609c5f8efea
10. **Python3 คืออะไร**, Retrieve from https://www.aosoft.co.th/article/322/Python-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2-python-%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%97%E0%B9%8D%E0%B8%B2%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html
11. **Docker คืออะไร**, Retrieve from https://blog.datawow.io/ease-datasci-works-with-nvidia-docker-bc8f8d58bf48

12. **โปรแกรม Visual Studio Code คืออะไร**, Retrieve from https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%A1/microsoft/4829-visual-studio-code.html
13. **Word Embedding และ Word2Vec คืออะไร**, Retrieve from https://lukkiddd.com/word-embedding-%E0%B9%81%E0%B8%A5%E0%B8%B0-word2vec-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-e60bdf6d78d3
14. **Twitter US Airline Sentiment (Dataset),** Retrieve from https://www.kaggle.com/crowdflower/twitter-airline-sentiment
15. **Chinese_conversation_sentiment (Dataset),** Retrieve from https://github.com/z17176/Chinese_conversation_sentiment
16. **Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus,** Retrieve from: **https://www.semanticscholar.org/paper/Sentiment-Classification-with-Convolutional-Neural-Zhang-Chen/f5fa0bacbe1ba7d372a2940d8f139ef6531c39d9?navId=extracted**
17. **Caviar's Word2Vec Tagging For Menu Item Recommendations,** Retrieve from https://medium.com/square-corner-blog/caviars-word2vec-tagging-for-menu-item-recommendations-13f63d7f09d8