การวิเคราะห์ความรู้สึกในโพสต์ Twitter

Sentimental analysis using Twitter data

นายซิง แซ่เสี่ย

Mr.Qin Xie

นายกฤษณะ นีโรลา

Mr.Kritshana Nirola

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ.2561

คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงงาน การวิเคราะห์ความรู้สึกในโพสต์ Twitter

Sentimental analysis using Twitter data

นิสิต นายชิง แซ่เสี่ย58102010190

นายกฤษณะ นีโรลา58102010800

ปริญญา วิทยาศาสตรบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงงาน อ.ดร.ศิริสรรพ เหล่าหะเกียรติ

ลงชื่อ.....

(อ.ดร.ศิริสรรพ เหล่าหะเกียรติ)

อาจารย์ที่ปรึกษาโครงงาน

บทคัดย่อ

ปัจจุบันผู้คนมากมายนิยมโพสต์สิ่งต่างๆบนสังคมออนไลน์มากมาย ทั้งFacebook Twitter YouTube ซึ่งแต่ละโพสต์จะสามารถบอกความรู้สึกของผู้โพสต์ได้ ในการวิจัยนี้เราสนใจปัญหาการวิเคราะห์และทำนายความรู้สึกของข้อความว่า เป็นบวก เป็นกลาง หรือเป็นลบ โดยใช้การเรียนรู้แบบมีผู้สอน ซึ่งเราทำการทดลองการแบ่งกลุ่มโพสต์โดยใช้ข้อความในโพสต์แต่ละโพสต์ มาทำการวิเคราะห์ด้วยเทคนิคการเรียนรู้เชิงลึก คือเทคนิค Long Short-Term Memory(LSTM) เพื่อทำการประเมินประสิทธิภาพและเปรียบเทียบผลลัพธ์ของเทคนิคดังกล่าว ว่ามีความเหมาะสมและ มีประสิทธิภาพเพียงใด

Abstract

Nowadays, many people post things on various social networks like Facebook, Twitter and YouTube. In this study, we performed sentiment analysis over posts in Twitter to determine how users think about a topic in the social media. In this study, we adopted long short-term memory (LSTM) as a classifier used in the prediction. Posts are mapped using the word2vec method to map text to vectors for further analysis. We evaluate the performance of the proposed system in comparison with other algorithms to determine the suitability of the proposed system. The expert mental results show that the proposed system slightly outperform the compared algorithms

ประกาศคุณูประการ

การวิเคราะห์ความรู้สึกในโพสต์ Twitter (Sentimental analysis using Twitter data) ผู้วิจัยได้มุ่งมั่นศึกษาและค้นคว้าอย่างต่อเนื่องจนได้รับความรู้และความเข้าใจจนสามารถสรุปได้ว่า ผู้ใช้ Twitter ในปัจจุบันมีความรู้สึกต่อหัวข้อที่สนใจในทางด้านบวก กลาง หรือด้านลบ โครงงานนี้สามารถสำเร็จลุล่วงเพราะได้รับความเมตตาและความช่วยเหลือจากบุคคลหลายๆ ท่าน ตั้งแต่เริ่มต้นจนเสร็จสิ้นโครงงาน คณะผู้วิจัยขอขอบพระคุณบุคคลทุกท่านที่ให้ความรู้ คำแนะนำ แนวทางการแก้ไขปัญหา และให้ความอนุเคราะห์เพื่อการศึกษาโครางงานนี้

ขอขอบพระคุณ อ.ดร.ศิริสรรพ เหล่าหะเกียรติ อาจารย์ที่ปรึกษาโครงงาน ผู้ซึ่งให้คำแนะนำ คำปรึกษา ความใส่ใจ และความช่วยเหลืออย่างเต็มที่รวมไปถึงการตรวจสอบ แก้ไข ตลอดจนให้คำปรึกษาในทุกๆ รายละเอียดของขั้นตอนการทำงานจนโครงงานบรรลุวัตถุประสงค์ตามเป้าหมายที่ตั้งไว้

ขอขอบพระคุณแหล่งข้อมูลที่ใช้ในการวิเคราะห์ ซึ่งได้แก่ Twitter Developer ที่สามารถให้นำข้อมูลจากTwitter มาใช้ในการวิเคราะห์ได้ และผู้ใช้Twitter ทุกท่าน ที่เขียนโพสต์ต่างๆ ซึ่งเป็นข้อมูลสำคัญสำหรับการวิเคราะห์นี้

นายซิง แซ่เสี่ย

นายกฤษณะ นีโรลา

สารบัญ

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันสังคมเรามีการก้าวหน้าในด้านการติดต่อสื่อสารกันอยู่มากซึ่งทำให้เกิดสังคมออนไลน์หรือ Social network ขึ้น ทำให้ผู้คนส่วนใหญ่มีความสนใจคอยอ่าน หรือติดตามข่าวสารกันทางออนไลน์มากขึ้น เพราะมีความเร็วและสะดวกสบาย ทำให้การที่มีเหตุการณ์อะไรเกิดขึ้นในสังคมจริงนั้นจะมีการกล่าวถึงในสังคมออนไลน์กันอย่างรวดเร็ว ทำให้ในปัจจุบันผู้คนสนใจข้อมูลทางสังคมมากขึ้นกว่าแต่ก่อน

การศึกษาความรู้สึกของคนส่วนใหญ่ในสังคม กับประเด็นหนึ่งๆ สามารถสะท้อนให้เห็นแนวทางของสังคมได้ ดังนั้น จึงมีการศึกษา sentiment analysis ใน social network ชนิดต่างๆ ทั้งนี้ Twitter เป็น social network ที่ได้รับความนิยมอย่างมาก อีกทั้งยังอนุญาติให้ผู้ใช้ สามารถเข้าถึงข้อมูลได้อย่างกว้างขวาง ดังนั้น ในการศึกษานี้ เราจึงเลือกวิเคราะห์ social network ชนิดนี้ ซึ่งเราสนใจที่จะวิเคราะห์ความรู้สึกของผู้ใช้ Twitter ในหัวข้อที่อยู่ในความสนใจ ของคนในสังคม เช่น การเลือกตั้ง62ที่จะถึงนี้ โดยเราอาจเลือกโพสต์ในTwitter ที่กล่าวเกี่ยวกับการเลือกตั้ง จากนั้นทำการวิเคราะห์ความรู้สึกของโพสต์ทั้งหมดที่เลือกมานั้น ว่ามีการพูดถึงในแง่บวกหรือแง่ลบ แล้วทำการเปรียบเทียบ สรุปผล ทำให้เราสามารถคาดการณ์แนวโน้มของสังคมในอนาคตได้ไม่มากก็น้อย โดยเราจะสร้างโมเดลโดยใช้ lstm(Long Short-Term Memory) เป็นหลักซึ่งจะนำมาเทียบกับโมเดลแบบอื่นๆ ซึ่งผลที่ได้พบว่าโมเดลที่เรากล่าวใช้มีผลที่ออกมามีความแม่นยำกว่าโมเดลอื่นๆ และสามารถทำนายความรู้สึกของโพสต์จากทวิตเตอร์ได้

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อวิเคราะห์ความรู้สึกหรือความนึกคิดของแต่ละโพสต์ในทวิตเตอร์

1.2.2เพื่อดูว่าปัจจุบันผู้ใช้งานTwitterมีการพูดถึงการเลือกตั้งในแง่ใดบ้าง

1.2.3 เพื่อศึกษาความรู้สึกต่อหัวข้อการเลือกตั้ง

1.3 ขอบเขตของโครงงาน

1.3.1 วิเคราะห์โพสต์ที่สนใจ เช่น เกี่ยวกับการเลือกตั้ง โดยที่จะทำการแบ่งออกเป็นทางด้านบวกกับด้านลบ

1.3.2 วิเคราะห์ sentiment analysis ที่มีการกล่าวถึงการเลือกตั้งตั้งแต่ช่วงมกราคม -มีนาคม 2562

1.3.3 ใช้ภาษาไพธอน เป็นหลักในการเขียนโปรแกรมเพื่อทำการวิเคราะห์ข้อมูล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

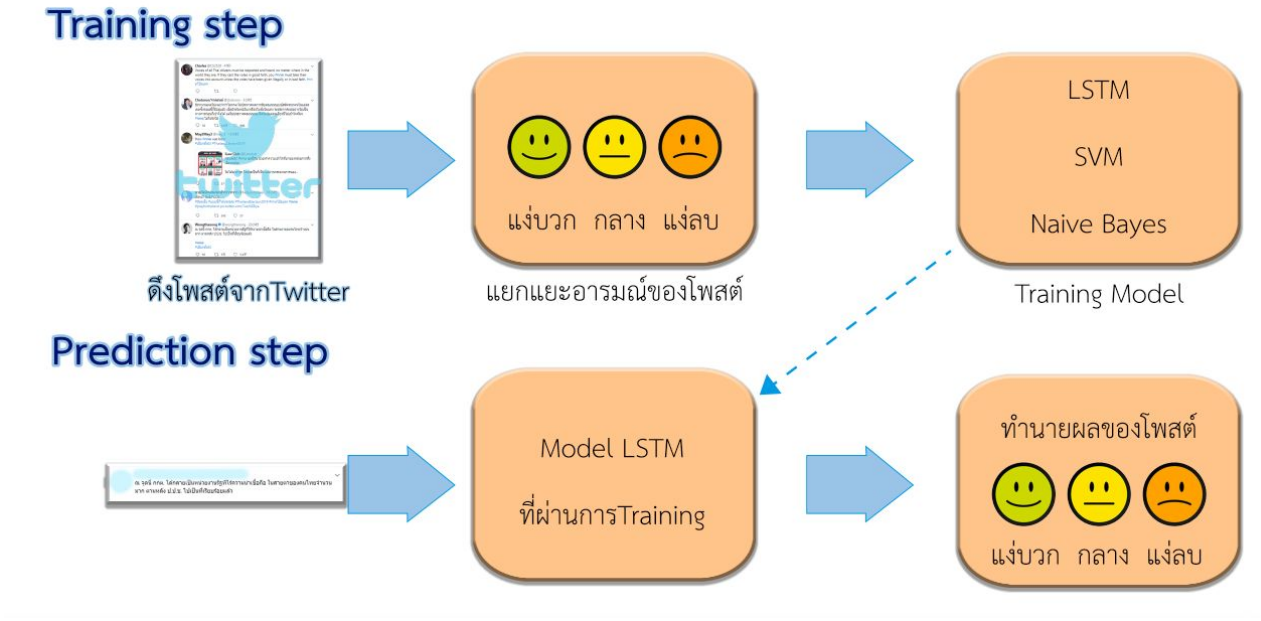
1.4.1 เพื่อศึกษาการวิเคราะห์โพสต์ในทวิตเตอร์ และสามารถแยกแยะความคิดของผู้โพสต์ว่าอยู่ในแง่บวกหรือลบได้

1.4.2 เพื่อดูว่าสังคมปัจจุบันมีความคิดและความรู้สึกเกี่ยวกับหัวข้อที่สังคมสนใจเป็นอย่างไร ซึ่งจะสามารถศึกษาผลกระทบและแนวโน้มในสังคมได้

1.4.3 ศึกษาวิธีการทำ Sentiment Analysis

1.4.4 ศึกษาวิธีการเก็บข้อมูลจาก Social network

1.5 กรอบแนวคิดการดำเนินงาน



รูปที่ 1-1 Training step กับ Pretiction step

ขั้นตอนการทำงาน ดังรูปที่ 2-1 แบ่งออกเป็น 2 ขั้นตอน คือ

1.5.1 Training step

ทำการดึงโพสต์จาก Twitter แล้วทำการแยกแยะอารมณ์ของโพสต์ต่างๆ โดยการกำหนด 1=แง่ลบ 2=กลาง และ 3=แง่บวก จากนั้นผ่านการTraining Model ในแบบต่างๆ คือ LSTM, SVM, Naive Bayes ซึ่ง LSTM เป็น Model ที่ใช้งานได้ผลดีที่สุด จึงเลือกใช้ LSTM เป็น Model หลัก

1.5.2 Prediction step

นำโพสต์จาก Twitter มา Pretiction โดยใช้ LSTM Model ที่ผ่านการ Training แล้ว ทำนายผลของโพสต์ว่าอยู่ในแง่ใดบ้าง

บทที่ 2

องค์ความรู้ที่เกี่ยวข้อง

2.1 องค์ความรู้เกี่ยวกับการ Retrieve Twitter data

เป็นกระบวนการในการดึงข้อมูลจากทวิตเตอร์เพื่อที่จะนำมาใช้ในการวิเคราะห์ข้อมูลเกี่ยวกับโพสต์นั้นๆ ที่จะใช้ API ในการดึงข้อมูล โดยเทคนิคในกระบวนการนี้มีอยู่ 2 ประเภทคือ

2.1.1 REST API

Representational State Transfer Application Programming Interface ในการเข้าถึงข้อมูลแบบ REST ปัจจุบันบริการ API ของ Facebook, Twitter, Google ก็เลือกใช้ REST API ที่เราจะสามารถเชื่อมต่อบริการในการดึงข้อมูล

2.1.2 Twitter Streaming API

ทวีตเตอร์ สตรีมมิ่ง API เป็นส่วนเชื่อมต่อโปรแกรม (API) ที่ทำงานผ่านเว็บ ซึ่งสตรีมมิ่งจะมีการเชื่อมต่อ HTTP อยู่ตลอด และส่งข้อมูลกับมาให้อย่างต่อเนื่อง ขณะที่ REST เมื่อรับส่งข้อมูลแล้วจะตัดการเชื่อมต่อ และต้องเชื่อมต่อไปใหม่อีกครั้ง ในการทำสตรีมมิ่งจะใช้ Tweepy ซึ่งเป็น open-sourcePython library เชื่อมต่อกับ Twitter stream สำหรับดึงข้อมูลมาเพื่อทำการวิเคราะห์

2.2 องค์ความรู้เกี่ยวกับ Sentiment Analysis

Sentiment Analysis หรือชื่อในภาษาไทย "การวิเคราะห์ความรู้สึก" เป็นการวิเคราะห์อารมณ์และความรู้สึกจากข้อความ เพื่อบ่งบอกความรู้สึกของผู้คนที่มีต่อบางสิ่งบางอย่าง เช่น ความรู้สึกดี (Positive) หรือ ความรู้สึกที่ไม่ดี (Negative)

ปัจจุบันนี้ได้มีการนำ Sentiment Analysis มาใช้งานในด้านต่าง ๆ เช่น ด้านการตลาด ด้านการสื่อสาร เป็นต้น ซึ่งในกระบวนการนี้ได้ใช้เทคนิค ดังนี้

2.2.1 nltk

Natural Language Toolkit ซึ่งเป็นโมดูลในภาษาไพธอนที่ได้รับความนิยมในการทำ การประมวลภาษาธรรมชาติ โดยใช้ Apache License, Version 2.0 และรองรับทั้ง Python 2 และ Python 3

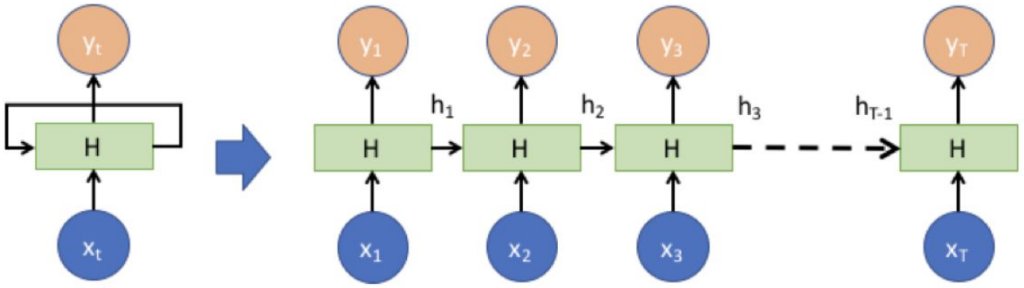
2.2.2 Deep learning

เป็นแนวทางใหม่ของวงการ AI , โดย deep learning มีโครงสร้างที่ประกอบด้วย input layer , hidden layer และ output layer โดยคำว่า deep นั้นหมายถึงการที่มี hidden layer มากกว่า 2 layer อีกทั้งยังมี neural networks หลากหลายประเภท ซึ่งแต่ละประเภทมีโครงสร้างที่ถูกออกแบบมาเพื่อทำงานที่แตกต่างกันไป ตัวอย่างเช่น CNN จะทำงานได้ดีกับรูปภาพ ส่วน RNN จะทำงานได้ดีข้อมูลประเภทลำดับเวลา(time series) และการพิสูจน์อักษร (text analysis)

2.2.3 Recurrent Neural Network

RNN แนวคิดหลักๆ ของ RNN เพื่อจะใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (sequence) เช่น video (sequence of images) หรือ text (sequence of words) เพื่อที่ให้เข้าใจภาพการทำงานของ RNN ได้ง่ายขึ้น ขอ

ยกตัวอย่างการทำ sequential data ก่อน โดยขอเทียบกับการอ่านหนังสือ ซึ่งเป็นลำดับของคำที่ต่อๆ กัน (sequence of words) การอ่านทีละคำต่อๆกันทำให้เราสามารถรู้เรื่องได้ว่าประโยคที่เรากำลังอ่านนั้นเกี่ยวกับอะไร เรื่องราวจากสิ่งที่เราอ่านผ่านไปแล้ว (ขอเรียกว่า hidden state หรือ state ก่อนหน้า) มาผสมกับคำที่เพิ่งจะอ่านไป (input data หรือ คำที่เรากำลังอ่าน ณ ตอนนั้น) ทำให้เราเข้าใจความหมายในส่วนตรงที่กำลังอ่านได้ ซึ่ง RNN ก็ใช้หลักการเดียวกัน คือ การปรับรูปแบบของ Neural network เดิม เพื่อให้สามารถเอา state หรือความรู้ก่อนหน้ามาบวกกับ input data ตัวใหม่ที่เข้ามา เพื่อทำความเข้าใจอะไรสักอย่างไปเรื่อยๆ



รูปที่ 2-1 ภาพแสดงหลักการทำงานของ RNNs

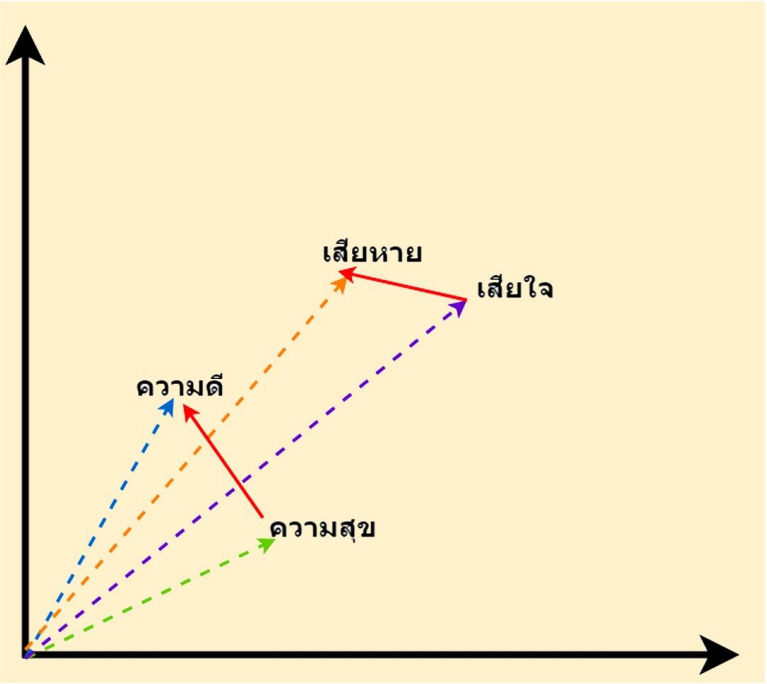
2.2.4 Long Short-Term Memory

LSTM เพื่อแก้ปัญหาของ RNN ที่มีต่อ sequence ยาวๆ ของข้อมูล จึงมีการเสนอการใช้ Long Short-Term Memory หรือ LSTM นี้ขึ้นมาโดย Sepp Hochreiter และ Juergen Schmidhuber

2.2.5 Woed2Vec

เป็นโมเดลที่ใช้สร้าง word embedding พัฒนาโดยทีมนักวิจัยของ Google นำโดย Tomas Mikolov ซึ่งโมเดลนี้สามารถทำงานได้ดีกว่าวิธีแบบเดิม ๆ (Latent Semantic Analysis) ซึ่ง word2vec คือการแสดง “คำ” ให้อยู่ในรูปของ “vector” แต่จะไม่ได้ใช้ one-hot encoding ในการสร้างตัวเลข vector แบบเดิมๆ word2vec จะใช้วิธีการคำนวนตัวเลขของคำนั้น ๆ จาก context รอบๆ คำนั้น (ไอเดียมาจาก language model) นอกจากนั้น vector ของคำต่างๆยังแสดงความสัมพันธ์ของคำ ทำให้สามารถนำ vector ของแต่ละคำ มาคำนวนต่อได้เช่น

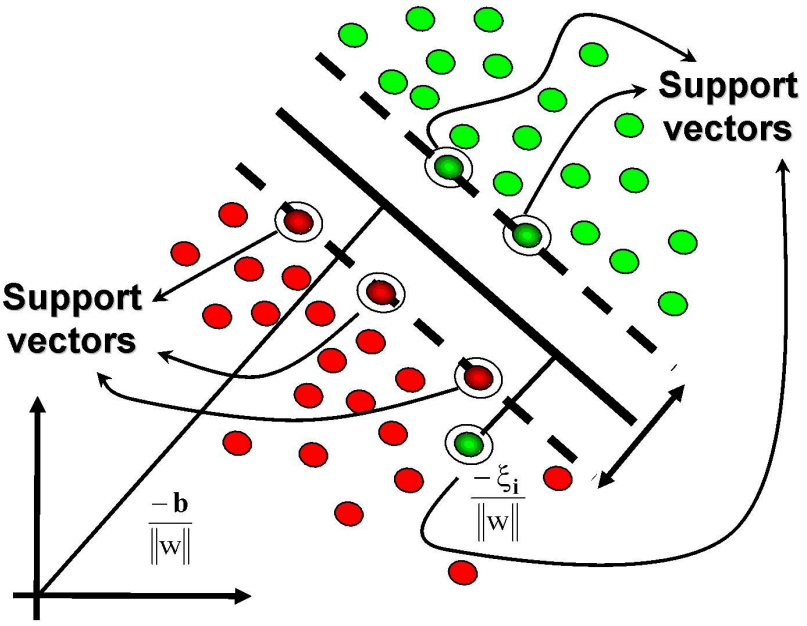
สามารถหาความคล้ายของคำต่างๆจากการคำนวน scalar product ระหว่าง vectors ของคำต่างๆ เพื่อคำนวนหาคำที่มีความคล้ายคลึงกันได้เช่น ดีใจ คล้ายกับ ความสุข แฮปปี้ และอื่นๆ สามารถหาคำเปรียบเทียบได้เช่น ถ้า ชื่นชม คือ คำแง่บวก แล้ว ยินดี จะคือคำแง่อะไร เป็นต้น



รูปที่ 2-2 ภาพแสดงตัวอย่าง Word2Vec การแบ่งคำให้อยู่ในตำแหน่งบนกราฟต่างๆ

2.2.6 Support Vector Machine

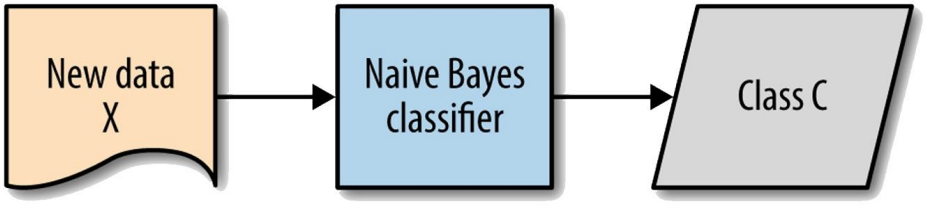
SVM เป็นเทคนิคหนึ่งที่ได้รับความ นิยมอย่างแพร่หลายในงานที่เกี่ยวข้องกับการจดจำรูปแบบตลอดจนการแก้ปัญหาการจัดกลุ่ม (classification problem) โดยอาศัยหลักการของการหา สัมประสิทธิ์ของสมการเพื่อสร้างเส้นแบ่งแยกกลุ่มข้อมูลที่ถูกป้อนเข้าสู่กระบวนการสอนให้ระบบเรียนรู้ โดยเน้นไปยังเส้นแบ่งแยกกลุ่มข้อมูลได้ดีที่สุด (optimal separating hyperplane) เมื่อเราพิจารณาข้อมูล ที่ประกอบด้วยข้อมูล 2 กลุ่ม



รูปที่ 2-3 ภาพประกอบการอธิบาย เกี่ยวกับหลักการสำคัญของ SVM

2.2.7 Naive Bayes

ใช้เทคนิคการวิเคราะห์ความน่าจะเป็น ด้วย Bayes theory เพื่อสร้างโมเดลสำหรับจำแนกแยกแยะกลุ่มของข้อมูล สามารถนำเทคนิคนี้ มาใช้ จำแนกแยกแยะข้อมูลเพื่อ วิเคราะห์ความรู้สึกได้



รูปที่ 2-4 ภาพรวมของการแบ่งแยกข้อมูลโดย Naive Bayes

2.2.8 Math plot graph

เป็นการนำข้อมูลเชิงคณิตศาสตร์มาทำการสร้างเป็นแผนภูมิ เพื่อให้ผู้อ่านสามารถเข้าใจได้อย่างรวดเร็ว

2.3 งานวิจัยที่เกี่ยวของ

2.3.1 A study of sentiment analysis using deep learning techniques on Thai Twitter data

ผู้ทำ : Peerapon Vateekul Thanabhat Koomsubha

เป้าหมายของงานวิจัย:การใช้เทคนิค deep learning ในการทำ Sentiment Analysis กับข้อมูลใน Twitter

ผลการทดลอง: การใช้เทคนิค deep learning ช่วยเพิ่มประสิทธิภาพในการทำ sentiment analysis ให้ความแม่นยำมากกว่า อัลกอริทึมชนิดอื่นๆ

บทที่ 3

วิธีการดำเนินงาน

3.1 กรอบแนวคิดการวิจัย

จากการที่กลุ่มของผู้วิจัยได้ทำการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับการนำข้อมูลมาวิเคราะห์ โดยข้อมูลที่ใช้ เช่น โพสต์ที่ติดHashtagเลือกตั้ง62(#การเลือกตั้ง62) เป็นต้น ผู้วิจัยจะรับข้อมูลจากการใช้ Library ในไพธอน ที่ศึกษาดึงข้อมูลออกมา แล้วนำข้อมูลมาทำการวิเคราะห์ความรู้สึกของแต่ละโพสต์ที่ติดHashtagที่เกี่ยวข้อง เช่นหัวข้อการเลือกตั้ง62 เป็นต้น ว่าโซเชียลมีเดียมีความคิดต่อการเลือกตั้งในแง่บวกหรือแง่ลบ

3.2วิธีการดำเนินงาน

3.2.1 ศึกษาหลักการใช้งาน Twitter Developer ของ Twitter

3.2.2 ศึกษาการทำงาน Library ที่เกี่ยวข้อง

3.2.3 ทำการเก็บข้อมูลที่เราสนใจใน Twitter

3.2.4 ทำการคัดกรองข้อมูลมาเฉพาะเกี่ยวกับหัวข้อเราสนใจ เช่น เลือกตั้ง เป็นต้น

3.2.5 ทำการศึกษาความรู้สึกของโพสต์แต่ละโพสต์ โดยใช้เทคนิคทาง sentiment analysis

3.2.6 ทดสอบข้อมูลที่วิเคราะห์ และประเมินผล

3.2.7 สรุปขั้นตอนการดำเนินการวิจัยทั้งหมด

3.3 แผนการดำเนินงาน

แผนการดำเนินงาน	2561					2562				
	ส.ค	ก.ย	ต.ต	พ.ย	ธ.ค	ม.ค	ก.พ	มี.ค	เม.ย	พ.ค
1.วางแผนการดำเนินงาน										
ประชุมเพื่อเลือกหัวข้อ										
ศึกษาหลักการและทฤษฎีที่เกี่ยวข้อง										
ศึกษางานวิจัยที่เกี่ยวข้อง										
ค้นหา Dataset ที่จะนำมาใช้งาน										
2.เตรียมข้อมูลให้พร้อม										
3. ออกแบบและสร้างโมเดล										
ออกแบบโมเดล										
ดำเนินการสร้างโมเดล										
4.สรุปผลและประเมินผลลัพธ์โมเดล										
วัดประสิทธิภาพโมเดล										
สรุปผลการดำเนินงาน										
5.จัดทำรูปเล่มรายงาน										

3.4 ดำเนินการสร้างโมเดล

3.4.1 ดึงข้อมูลมาจาก twitter โดยใช้ Twitter Api (โค้ดจากไฟล์ twitter.ipynb)

```
#Import the necessary methods from tweepy library
import tweepy
import csv
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream, API
from google.colab import drive
drive.mount('/content/gdrive')
import numpy as np
import pandas as pd
import re

#Variables that contains the user credentials to access Twitter API
access_token = "926273096448954368-GJucrewoThINRMihXyms0ZiZAL732em"
access_token_secret = "nosyzDuxhzGscEqOq04xOA1xF8gXLOwDSkJBCRgJfAbTx"
consumer_key = "oYXFQfDikbkDbwCk95j1aTTRq"
consumer_secret = "iVjawYb4CjcF54Zo4zTQkCVc5B7YJuEPwqaER2ioB10njQG4K6"

#def twitter_setup():
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)

df = pd.DataFrame(columns=['Date','Post'])
```

```
# Open/Create a file to append data
#csvFile = open('/content/a.csv', 'a')
#Use csv Writer
#csvWriter = csv.writer(csvFile)
#clear datafreame
df = df.iloc[0:0]
file = open('twitter_text.txt','w')
for tweet in tweepy.Cursor(api.search, q='เลือกตั้ง', tweet_mode="extended").items(1500):
  if  not tweet.retweeted and ('RT @' not in tweet.full_text)
                          and ('BNK' not in tweet.full_text)
                          and ('@' not in tweet.full_text):
    print (tweet.created_at, tweet.full_text)
    #df.append(pd.Series([textCre, textMsg]) , ignore_index=True)
    df = df.append(pd.DataFrame({"Date": [tweet.created_at], "Post": [tweet.full_text]}))
df.to_csv('/content/test1.csv', encoding='utf-8')
file.close()
```

```
2019-04-26 01:12:26 ไม่มีใครเอา #ประยุทธ์ #ประยุทธ์จันทร์โอชา #กกต์โป๊แตก #กกตซุ่ย #เลือกตั้ง #เลือกตั้ง62 #คอรัปชั่น
                    #thailandelcetion2019 #protest #โกงการเลือกตั้ง62 #คสช https://t.co/jZuPojLkUO
2019-04-25 15:40:02 ดู #AvengersEndgame เหมือนกำลังดูลุงตู่สู้กับ พรรคไม่เอา #ลุงตู่ ยังไงยั้งอะ คนที่ออกไปใช้สิทธิ #เลือกตั้ง เหมือนทีมอเวนเจอร์ไปเอา
                    อัญมณี แต่ลุงตู่ก็มาแย่งคืน ส่วนอุตตม เหมือนอีผีที่ช่วยฑานอสอะ ดึงอัญมณีกลับไป #เอาอำนาจกูกลับมา
2019-04-25 13:30:30 โคตรซึกเหิม หลังจากคุณธนาธรให้สัมภาษณ์เสร็จ สู้เพื่อประชาธิปไตย #สู้ๆค่ะพ่อ 🖤🖤#savethanathron #ธนาธร #อนาคตใหม่
                    #พรรคอนาคตใหม่ #เลือกตั้ง https://t.co/e4sXW6DyPR
```

รูปที่ 3-1 ภาพแสดงตัวอย่าง การดึงข้อมูล

3.4.2 ทำการ label data โดยระบุเป็นหมายเลข1,2,3 แทนแง่ต่างๆ ใน excel

Date	Sentiment	Post
2/22/19 6:47	1	คือถ้าเป็นแบบนี้มึงก็ไม่ต้องเลือกตั้งก็ได้มั้ง ได้คะแนนฟรีจากคนไม่ได้เลือก โหวตโน ไหนจะเก้าอี้ลม 250 สว. แล้วถ้าสมมติเกิดช
2/22/19 6:46	1	ไม่ใช่แค่ปั้ม กม.ปัญหาถี่ยิบ เอฟทีเอว็อทช์ แจ! รบ.คสช.โค้งสุดท้ายเตรียมยื่นเข้า #CPTPP ตัดหน้า รบ.เลือกตั้ง | ประชาไท http
2/22/19 6:45	3	#TPTV #โทรทัศน์รัฐสภา ให้ความรู้เรื่องการเลือกตั้งชัดเจนดี ชื่นชมๆ ?
2/22/19 6:45	1	การเมืองแบบเก่า มีพรรคการเมืองเก่าแก่ พรรคหนึ่ง มีความพยายามจ้างหัวคะแนน เพื่อเกณฑ์คนไปฟัง หัวหน้าพรรคและแกนนำ
2/22/19 6:45	2	#savethanaton มีใครเลือกตั้งนอกเขตบ้าง ที่ลงทะเบียนในเว็บอ่ะ
2/22/19 6:44	1	@sandandee ก็ลาออกมาเปนรักษาการณ์สิคะ ก็เขาไม่ยอมลาออกเองอะ โทษใครละทีนี้ จะเลือกตั้งยุแล้วยังออกกฎหมายโครม
2/22/19 6:43	1	นั่งคิดอยู่ว่าทำไมลุงต้องเล่นงานคนอื่นด้วยทั้งๆที่ตัวเองก็มี สว. 250 เสียง ยังไงก็ได้เป็นรัฐบาลแน่ๆอยู่แล้ว เพิ่งมาเข้าใจ ที่ผ่านม
2/22/19 6:43	1	# แนวร่วมพรรคฝ่าย ประชาธิปไตยแห่งชาติ# เตรียมตัวรับสถานการณ์ # วงจรอุบาทว์ทางการเมือง# ก่อนและหลังเลือกตั้ง 24 มีค
2/22/19 6:42	2	@popohobi แคมเปญจบ บังทันมาเลือกตั้งค่ะ
2/22/19 6:42	1	เป็นผู้สมัคร ? ถ้าใช่พูดจาข่มขู่แบบนี้ผิดกฎหมายเลือกตั้งไหมเนี่ย https://t.co/BTWok3D3mN
2/22/19 6:41	2	16 มีค 17 เลือกตั้ง 18 ทำงาน คือข้านจะได้ไปงานของมาสฟลลมั้ย?????
2/22/19 6:41	1	@prasitchai_k ถึงวันเลือกตั้งจะเหลืออยู่พรรคเดียว. ไม่ต้องเดาใครๆๆๆก็รู้???
2/22/19 6:41	1	เมื่อไรตุบจะลาออกคะ นี่ก็จะเลือกตั้งแล้ว? #เลือกตั้ง62
2/22/19 6:40	2	เลือกตั้ง 2562: ปัดเร่งคดี "ธนาธร" ไลฟ์วิจารณ์คสช. https://t.co/PnFYLuWhir #ThaiPBSnews #เลือกตั้ง62
2/22/19 6:39	1	@PINKXTHIEF อย่าเขียนไปจริงสิ อ้ย 5555555555 ค่อยๆคิด อีกกี่วันเลือกตั้ง พี่ยังไม่รู้เลย
2/22/19 6:37	2	เมื่อวันที่ 22 กุมภาพันธ์ 2562 เวลา 12.00 น. คุณหญิงสุดารัตน์ เกยุราพันธุ์ ประธานยุทธศาสตร์การเลือกตั้งพรรคเพื่อไทย พบประ ภาพ... https://t.co/Upkk9c9axt
2/22/19 6:36	1	เขียนนโยบาย กับดำเนินนโยบาย มันไม่ได้ง่ายเหมือนกันนะ ตลอดช่วยชีวิตที่เลือกตั้งได้ ไม่เห็นมีรัฐบาลไหน ทำอย่างพูดได้หมด
2/22/19 6:36	1	เขียนกติกาเอาเปรียบแล้วลงเล่นเอง แถมยังเตะตัดขา เจาะยางตลอดเวลา....จะมีเลือกตั้งไปทำไมวะก็อยู่มันไปอีก 20ปีเลยซี้.....เ

3.4.3 ทำการสร้างโมเดลขึ้นมา (โค้ดจากไฟล์ Naive_Svm.ipynb)

จากไฟล์ Naive_Svm.ipynb มีโมเดล 2 แบบ คือ Naive Bayes และ SVM

```
!pip install deepcut
import pandas as pd
import deepcut
import numpy as np
import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB
from gensim.models.word2vec import Word2Vec
from sklearn.model_selection import train_test_split
from sklearn import svm
%matplotlib inline
from google.colab import drive
drive.mount('/content/gdrive')
```

```
df = pd.read_csv("/content/gdrive/My Drive/CleanDataLatest.csv", encoding='utf-8')
df.tail(10)
```

	Unnamed: 0	text	class
2142	2142	ชอบบรรยากาศการเลือกตั้งค่ะรู้สึกสวยหัวกระไดไม่...	3
2143	2143	เลือกตั้งแล้วดีใจจัง	3
2144	2144	เลือกตั้งครั้งนี้ขอให้ได้รัฐบาลที่ดี	3
2145	2145	เก่งดีไม่โกงเห็นแก่ส่วนรวม	3
2146	2146	ดีใจจังอนาคตใหม่ได้คะแนนเยอะมากกว่าที่คิด	3

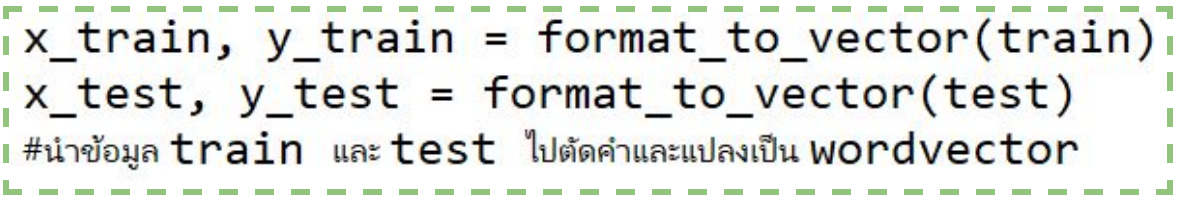
รูปที่ 3-2 ภาพแสดง dataframe ของข้อมูลเพื่อสร้างโมเดล svm และ naive bayes

```
wordvector = Word2Vec.load('/content/gdrive/My Drive/th.bin')
```

รูปที่ 3-3 ภาพแสดงทำการเก็บค่า wordvector โดยที่ไฟล์ th.bin จะเป็นไฟล์ที่มีให้โหลดโดยทั่วไป

```
def format_to_vector(df, maxlen=18):
    sentences = []
    for index, row in df.iterrows():
        s = []
        for word in deepcut.tokenize(row['text']):
            try:
                s.append(wordvector.wv[word])
            except:
                continue
        if len(s) == 0:
            df = df.drop(index)
            continue
        if len(s) < maxlen:
            tmp = [np.zeros(len(s[0]))] * (maxlen - len(s))
            s.extend(tmp)
        else:
            s = s[:maxlen]
        sentences.append(s)
    return np.asarray(sentences), np.asarray(df['class'].tolist())
#ตัดคำแล้วนำคำมาแปลงเป็น wordvector
```

```
train, test  = train_test_split(df, test_size=0.2)
#แบ่งข้อมูลสำหรับ train test
```



รูปที่ 3-4 ภาพแสดงการนำข้อมูล train และ test ไปตัดคำและแปลงเป็น wordvector

```
def format_to_vector_2(df):
    new_a = []
    new_y = []
    for x in df:
        temp_a = []
        sorted_labels = sorted(x, key=lambda x: x[1], reverse=True)
        new_y.append(sorted_labels[0][0])
        for z in x:
            temp_a.append(z[1])
        new_a.append([abs(x) for x in temp_a])
    return new_a
```

รูปที่ 3-5 ภาพแสดง function สำหรับการแปลงคำเป้นเวกเตอร์

```
x_train_2 = format_to_vector_2(x_train)
x_test_2 = format_to_vector_2(x_test)
```

```
alg = MultinomialNB()
demoNB = alg.fit(x_train_2, y_train)
```

รูปที่ 3-6 ภาพแสดงส่วนของการสร้าง Naive Bayes Model

```
alg_2 = svm.SVC()
demoSVM = alg_2.fit(x_train_2, y_train)
```

รูปที่ 3-7 ภาพแสดงส่วนของการสร้าง SVM Model

3.4.4 ต่อไปจะเป็นกระบวนการสร้างโมเดล โดยใช้ lstm จากไฟล์ ModelLSTM

```
label = preprocessing.LabelBinarizer()
label.fit([1, 2, 3])
#เป็นการ set label
```

รูปที่ 3-8 ภาพแสดงการset label

```
train, test = train_test_split(df, test_size=0.2)
#แบ่งข้อมูลโดย test เป็น 20% และ train เป็น 80%
```

รูปที่ 3-9 ภาพแสดงการแบ่งข้อมูลโดย test เป็น 20% และ train เป็น 80%

```
x_train = train['text'].tolist()
y_train = label.transform(train['class'])
#x_train คือ ข้อความ y_train คือ ความรู้สึก
```

รูปที่ 3-10 ภาพแสดงการกำหนด x_train คือข้อความ y_train คือความรู้สึก

```
x_test = test['text'].tolist()
y_test = label.transform(test['class'])
```

```
print('Train: {}'.format(len(train)))
print('Test: {}'.format(len(test)))
#เช็คจำนวนสัดส่วนการแบ่งของ train test
```

รูปที่ 3-11 ภาพแสดงการเช็คจำนวนสัดส่วนการแบ่งของ train test

```
Train: 1721
Test: 431
```

```
wordvector = Word2Vec.load('gdrive/My Drive/th.bin')
```

```
def get_word_vectors(sentence, model, maxlen=200):
    tokenize = deepcut.tokenize(sentence)
    w = np.zeros((maxlen, 300))
    for i, word in enumerate(tokenize):
        if i > maxlen:
            break
        try:
            w[i] = model.wv[word]
        except:
            continue
    return w
  #function สำหรับการตัดคำ และแปลงเป็น เวกเตอร์
```

รูปที่ 3-12 ภาพแสดง function สำหรับการตัดคำ และแปลงเป็น เวกเตอร์

```
x_train_WoVe = []
for x in x_train:
    x_train_WoVe.append(get_word_vectors(x, wordvector))
x_train_WoVe = np.asarray(x_train_WoVe)
#นำข้อมูล x_train_WoVe ไปทำการตัดคำและแปลงเป็น wordvector
```

รูปที่ 3-13 ภาพแสดงการนำข้อมูล x_train_WoVe ไปทำการตัดคำและแปลงเป็น wordvector

```
x_test_WoVe = []
for x in x_test:
    x_test_WoVe.append(get_word_vectors(x, wordvector))
x_test_WoVe = np.asarray(x_test_WoVe)
#นำข้อมูล x_test_WoVe ไปทำการตัดคำและแปลงเป็น wordvector
```

รูปที่ 3-14 ภาพแสดงนำข้อมูลx_test_Wove ไปทำการตัดคำและแปลงเป็น wordvector

```
def build_model(x_train):
    model = Sequential()
    model.add(LSTM(250, return_sequences=True, input_shape=x_train.shape[1:]))
    model.add(Bidirectional(LSTM(250)))
    model.add(BatchNormalization())
    model.add(Dense(3, activation='softmax'))
    model.compile(optimizer='rmsprop',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
    return model
  #method สำหรับtrain
```

รูปที่ 3-15 ภาพแสดง method สำหรับ train

```
model = build_model(x_train_WoVe)
model.summary()
#สร้างโมเดล LSTM 4 layer มี LSTM 2, Bidirectional 1, Dense 1 ปรับเป็น Softmax
```

รูปที่ 3-16 ภาพแสดง การสร้างโมเดล LSTM 4 layer

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_7 (LSTM)                (None, 200, 240)          519360
_________________________________________________________________
bidirectional_4 (Bidirection (None, 480)               923520
_________________________________________________________________
batch_normalization_6 (Batch (None, 480)               1920
_________________________________________________________________
dense_18 (Dense)             (None, 3)                 1443
=================================================================
Total params: 1,446,243
Trainable params: 1,445,283
Non-trainable params: 960
_________________________________________________________________
```

รูปที่ 3-17 ภาพแสดงผลจากการ build model x_train_WoVe

```
start_time = datetime.datetime.now()
history = model.fit(
                    x_train_WoVe,
                    y_train,
                    epochs=8,
                    batch_size=200,
                    validation_split=0.2
                )
end_time = datetime.datetime.now()
#ขั้นตอนการtrain ข้อมูล
```

รูปที่ 3-18 ภาพแสดงขั้นตอนการ train ข้อมูล

```
Train on 1376 samples, validate on 345 samples
Epoch 1/5
1376/1376 [==============================] - 8s 6ms/step - loss: 0.0387 - acc: 0.9906 - val_loss: 1.7780 - val_acc: 0.6667
Epoch 2/5
1376/1376 [==============================] - 9s 7ms/step - loss: 0.0339 - acc: 0.9927 - val_loss: 3.5589 - val_acc: 0.5594
Epoch 3/5
1376/1376 [==============================] - 9s 6ms/step - loss: 0.1038 - acc: 0.9680 - val_loss: 1.7797 - val_acc: 0.6348
Epoch 4/5
1376/1376 [==============================] - 8s 6ms/step - loss: 0.0294 - acc: 0.9920 - val_loss: 1.6631 - val_acc: 0.6783
Epoch 5/5
1376/1376 [==============================] - 8s 6ms/step - loss: 0.0297 - acc: 0.9935 - val_loss: 1.7087 - val_acc: 0.6493
```

รูปที่ 3-19 ภาพแสดง

3.4.5 ต่อไปจะเป็นกระบวนการPredict แบบ Realtime จากไฟต์ ModelLSTM

```
class MyStreamListener(StreamListener):
    def on_status(self, status):
        global j
        global p
        global n
        global ne
        global realtime_data
        global piechart_data
        update_time = time.time() - self.start_time
        #print(update_time)
        realtime_data = realtime_data.append(pd.DataFrame({"Post": [status.text],"Sentiment": 'E'}))
        realtime_data['Post'] = realtime_data['Post'].str.replace('http\S+|www.\S+|\n| #\S+|@\S+|#\S+|RT', '', case=False)
        realtime_data.drop_duplicates(subset=['Post'], keep=False)
        for index, row in realtime_data.iterrows():
          if row['Sentiment'] == 'E' :
            row['Sentiment'] = predictionData(row['Post'])
            if row['Sentiment'] == 'Negative':
              n = n+1
            if row['Sentiment'] == 'Neutral':
              ne = ne+1
            if row['Sentiment'] == 'Positive':
              p = p+1
        if (update_time >=j):
          # Pie chart, where the slices will be ordered and plotted counter-clockwise:
          labels = 'Negative', 'Neutral', 'Positive'
          sizes = [n, ne, p]
          explode = (0, 0.0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')
          fig1, ax1 = plt.subplots()
          ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',shadow=True, startangle=90)
          ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
          plt.show()
          print("จำนวนแง่บวก :",p)
          print("จำนวนกลาง :",ne)
          print("จำนวนแง่ลบ :",n)
          print("- - - - - - - - - - - ","Update Time : ",j,"- - - - -  - - - - - - - - - - ")
          j= j+ 900
    def __init__(self, time_limit=0):
        self.start_time = time.time()
        self.limit = time_limit
        self.count = 0
        self.clmt = 0 #
        super(MyStreamListener, self).__init__()
myStreamListener = MyStreamListener()
myStream = Stream(auth = api.auth, listener=myStreamListener, tweet_mode='extended')
myStream.filter(track=['เลือกตั้ง', 'เลือกตั้ง62','เลือกตั้ง2562'])
```

บทที่ 4

ผลการดำเนินงาน

หลังได้สร้างโมเดล ทั้งสามแบบเรียบร้อยจะพบว่า

โมเดลของ Naive Bayes มีความแม่นยำอยู่ที่ประมาณ 54% ดังนี้

```
(true_result*100)/(true_result+neg_result)
```

```
53.59628770301624 5
```

โมเดลของ SVM มีความแม่นยำอยู่ที่ประมาณ 55% ดังนี้

```
(true_result*100)/(true_result+neg_result)
```

```
54.98839907192575 5
```

ส่วนของโมเดล LSTM อยู่ที่ 65 %

```
accuracy_score(y_test.argmax(axis=1), model.predict_classes(x_test_WoVe))*100
```

```
66.82134570765661
```

```
print(classification_report(y_test.argmax(axis=1), model.predict_classes(x_test_WoVe)))
```

```
              precision    recall  f1-score   support

           0       0.57      0.82      0.67       158
           1       0.79      0.66      0.72       239
           2       0.20      0.03      0.05        34

   micro avg       0.67      0.67      0.67       431
   macro avg       0.52      0.50      0.48       431
weighted avg       0.66      0.67      0.65       431
```

ซึ่งเนื่องจากในขั้นตอนต่อไป เราต้องใช้ lstm ทำจึงทำการทดลอง predict ข้อความเกี่ยวกับการเลือกตั้ง ดังภาพข้างล่าง

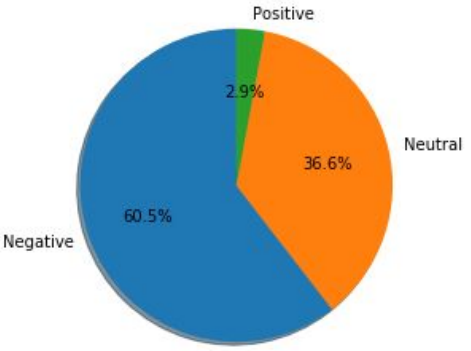
```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " ไม่ทำห่าไรเลยวันๆก็เอาแต่ ก็อบนโยบายเขาคิดเองไม่เป็น "
ผลการทำนาย      : " แง่ลบ :( "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " ก็ดีนะการดีเบตเนี้ยทำให้เห้นมุมมองของแต่ละพรรคได้ดีมาก "
ผลการทำนาย      : " แง่บวก :) "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " อย่าลืมไปใช้สิทธิเลือกตั้ง 24 นี้นะครับ "
ผลการทำนาย      : " เป็นกลาง :| "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " เลือกใครก็ได้ที่ไม่ใช่ลุงอะไม่งั้นประเทศล่มแน่ "
ผลการทำนาย      : " เป็นกลาง :| "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " เป็นกำลังใจให้ลุงนะคะ "
ผลการทำนาย      : " แง่บวก :) "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ข้อความของโพสต์ : " อยากให้พรรคอนาคตใหม่ชนะเลือกตั้งค่ะ พวกเขาน่าจะพาประเทศเราไปได้ไกล "
ผลการทำนาย      : " เป็นกลาง :| "
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

ซึ่งหากพบภาษาไทยที่ใช้คำศัพท์ผิด ก็จะพบว่าจะ predict ได้ค่าที่เพี้ยน

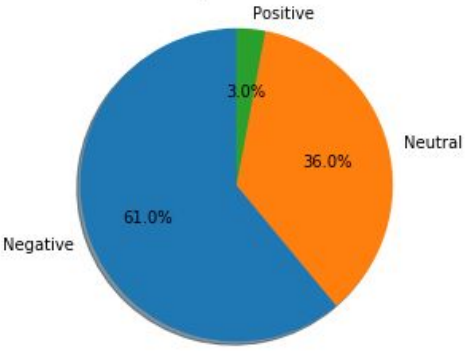
ข้อความของโพสต์ : " เป็นกำลังใจให้ลุงนะคะ "
ผลการทำนาย : " แง่ลบ :("

ข้อความของโพสต์ : " อยากให้พรรคอนาคตใหม่ชนะเลือกตั้งค่ะ พวกเขาน่าจะพาประเทศเราไปได้ไกล "
ผลการทำนาย : " แง่ลบ :("

ผลจาก Real time prediction



จำนวนแง่บวก : 6
จำนวนกลาง : 75
จำนวนแง่ลบ : 124
---------- Update Time : 900 -------- -------



จำนวนแง่บวก : 11
จำนวนกลาง : 134
จำนวนแง่ลบ : 227
---------- Update Time : 1800 -------- -----

บทที่ 5

สรุปผลการทดลอง

จากการทดลองในงานนี้สามารถทำการทำนายแบบเรียลไทม์ได้ แต่ยังพบว่าความแม่นยำที่ได้ในนั้นยังต่ำอยู่ ทำให้การทำนายความรู้สึกของโพสต์แบบเรียลไทม์มีความน่าเชื่อถือที่ต่ำ สืบเนื่องมาจาก data ที่นำมาใช้นั้นมีปัญหา คือข้อความของโพสที่ดึงมาจาก twitter เพื่อ train/test มีความกำกวมไม่ชัดเจนและภาษาวิบัติส่งผลให้สร้างโมเดลได้ไม่ดี จึงเป็นผลให้ความแม่นยำต่ำ

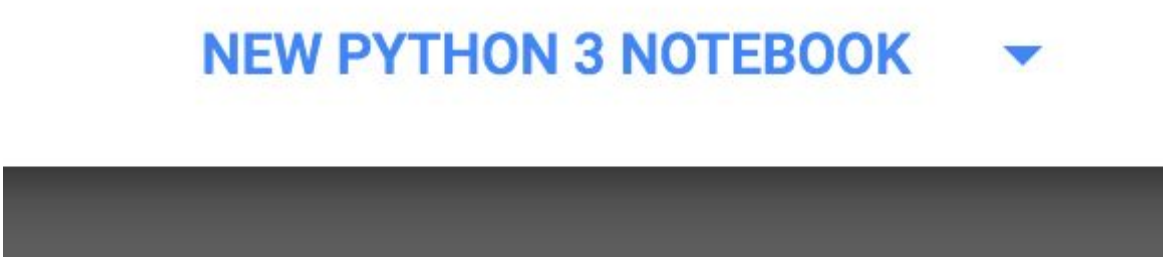
ภาคผนวก

:: วิธีใช้เบื้องต้น ::

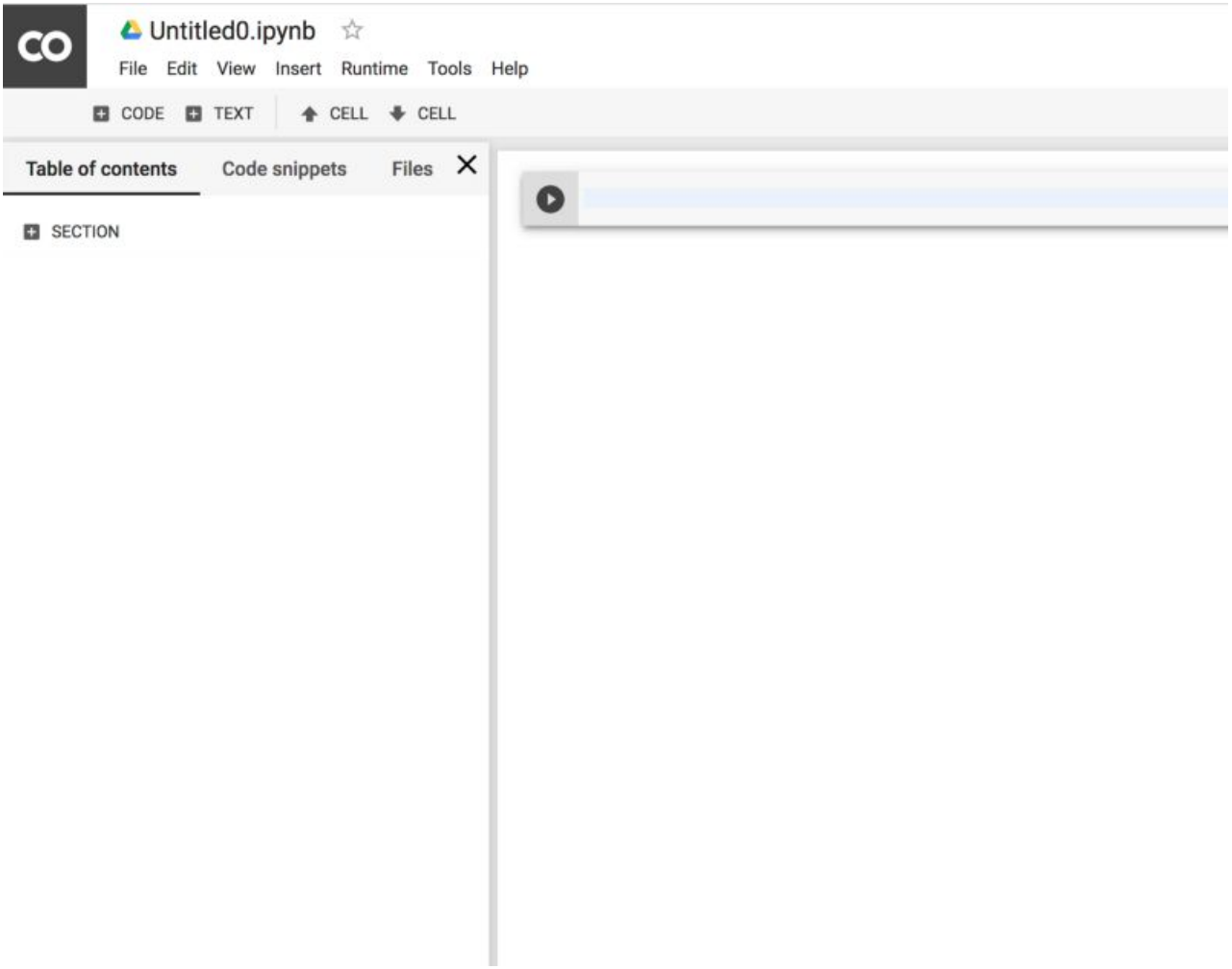
1. การสร้างไฟล์ใหม่

เข้าไปที่เว็บ http://colab.research.google.com/

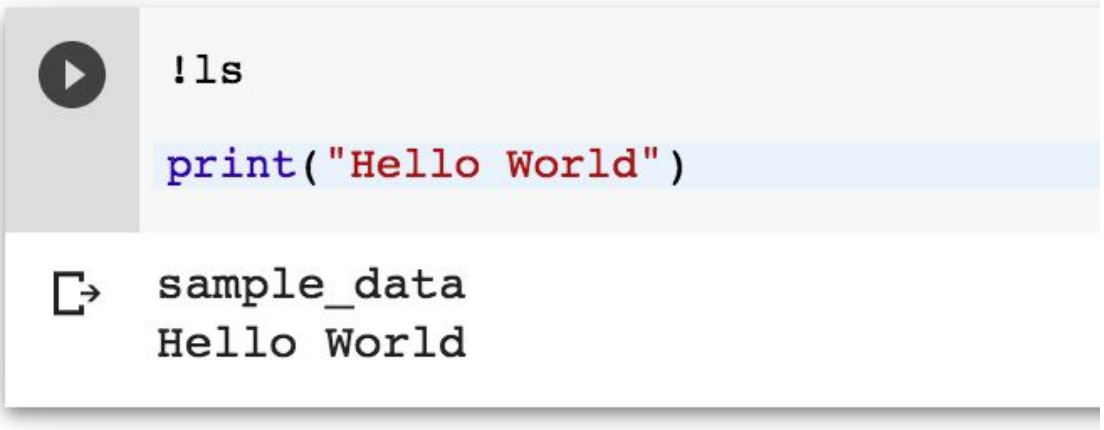
จากนั้นกดปุ่ม “NEW PYTHON 3 NOTEBOOK” เพื่อสร้าง Notebook ใหม่



เริ่มแรกจะได้เป็นหน้าเปล่า ๆ และให้ช่องสำหรับรันสคลิปเพียงช่องเดียว



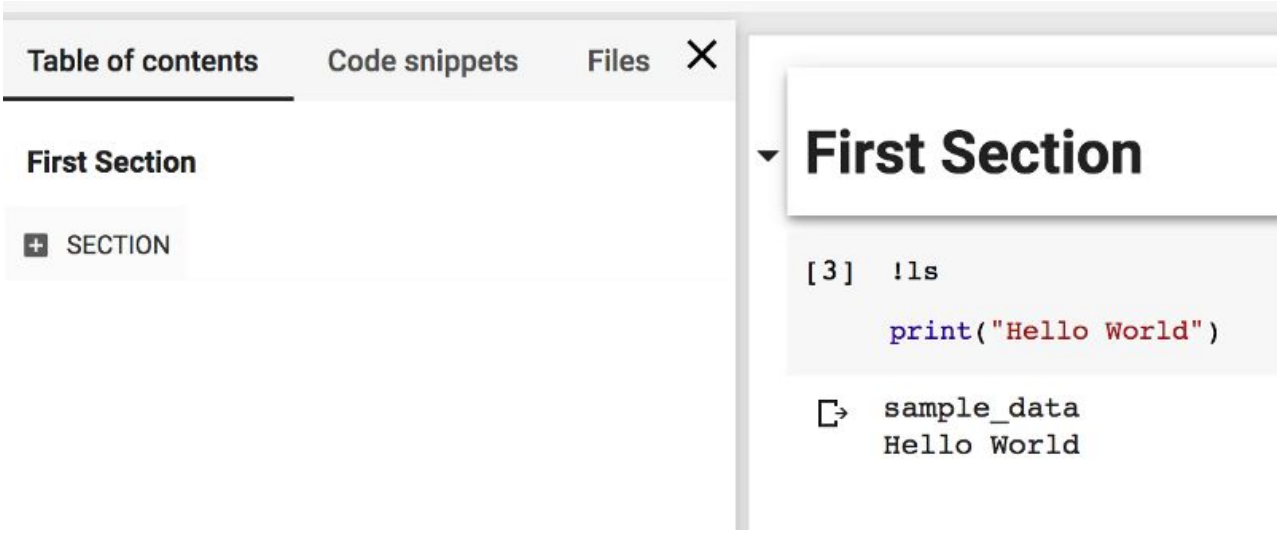
โดยช่องที่เค้าให้เรามานั้น สามารถที่จะใส่คำสั่ง Bash ร่วมกันกับโค้ดภาษา Python ได้เลย เช่น



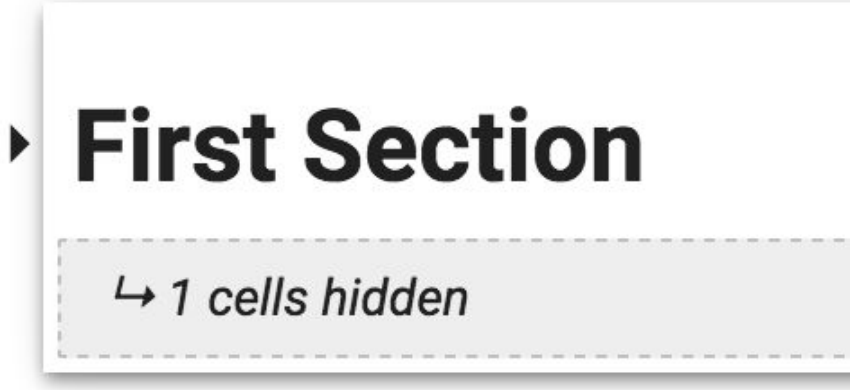
เพียงแค่ใส่เครื่องหมาย “!” เพื่อแยกระหว่างคำสั่ง Bash กับคำสั่งภาษา Python

2. สามารถแบ่งโค้ดออกเป็นส่วน ๆ

โดยกดปุ่ม “+ Section” ด้านซ้ายมือสุด



เมื่อเรากดย่อ โค้ดที่ต่อจาก Section นั้นก็จะถูกย่อตาม



3. การเพิ่มช่องสำหรับใส่สคลิปให้กดปุ่ม “CODE”

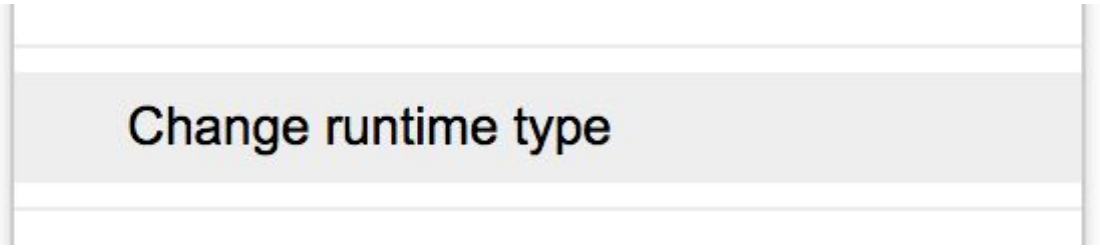
4. หากต้องการใส่ข้อความอธิบายใน Notebook ของเรากดปุ่ม “TEXT” และ

5. หากต้องการย้ายตำแหน่งวัตถุต่าง ๆ บนหน้าจอให้เลือก Cell นั้น ๆ ก่อนแล้วค่อยกดปุ่มดังรูปนี้

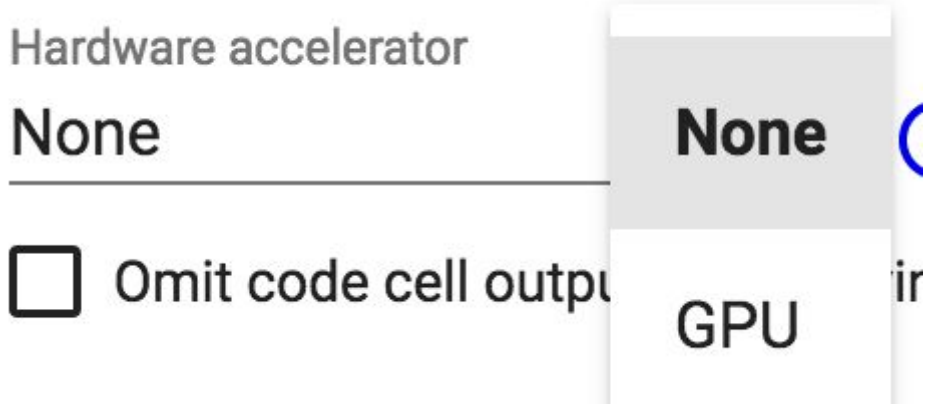
:: การเลือกใช้ GPU หรือ CPU และเวอร์ชัน Python ::

1. เลือกเมนู “Runtime” ตรงแถบเมนูด้านซ้ายบน

เลือก “Change runtime type”

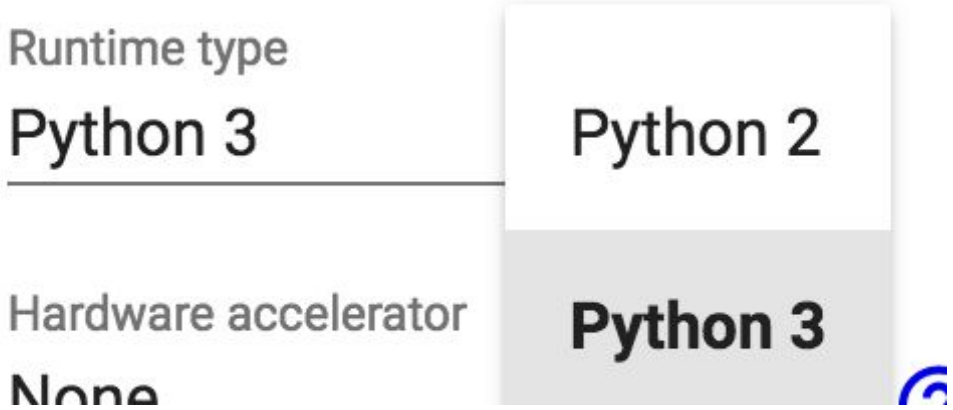


กดเลือก “GPU” เพื่อเปิดใช้งาน



2. เลือกเวอร์ชัน Python ที่จะให้ Compile

โดย Runtime Type มีให้เลือกอยู่ 2 เวอร์ชัน คือ Python 2.7.14 และ Python 3.6.3

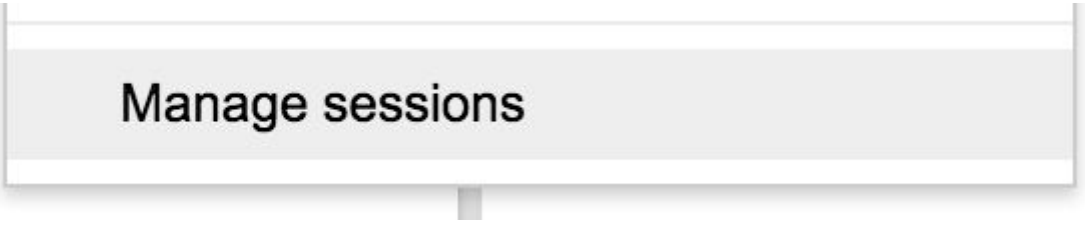


:: จัดการ Session ที่รันอยู่ ::

หากต้องการ Terminate งานที่รันอยู่สามารถทำได้ 2 วิธี

1 . กด Terminate

เข้าไปที่เมนู “Runtime” -> “Manage sessions”



จะปรากฏเซสชันให้เลือกทำลายทิ้งได้ หากต้องการทำลายให้กด “TERMINATE”

2. ผ่านการรัน Bash

ด้วยคำสั่งดังนี้

!kill -9 -1

:: การใช้ร่วมกับ Google Drive ::

ให้พิมพ์คำสั่งดังต่อไปนี้

```
from google.colab import drive

drive.mount('/content/drive')
```

ระหว่างรันสคลิปจะปรากฏช่องว่างให้ใส่ Authen code ที่ได้จากการกดลิงก์ด้านบน เพื่อเชื่อมต่อกับ Google Drive ของเรา

```
Go to this URL in a browser: https://accounts.google.com/

Enter your authorization code:
```

หากต้องการเปลี่ยน root path ของไดเรกทอรี่ เพื่อลดความยาวของการเรียก Path ไฟล์ โดยจะได้ไม่ต้องเขียน Path ที่ยาว ให้ใช้คำสั่งดังต่อไปนี้

import os

os.chdir("drive/My Drive/(ชื่อโฟลเดอร์ใน Google Drive ที่ต้องการ)")

เมื่อเปลี่ยน Path เข้ามาในโฟลเดอร์ของไดร์ฟเราแล้ว เวลาแก้ไขไฟล์เช่น ลบหรือสร้างใหม่ มันก็จะอัพเดทไปยัง Google Drive ให้เราทันทีด้วยเลย ซึ่งสะดวกอย่างมาก

แหล่งอ้างอิง

[1]S. Sudprasert, A. Kawtrakul, "Thai word segmentation based on Global and Local Unsupervised learning", NCSEC, 2003.

[2]F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, T. By, "Sentiment Analysis on Social Media", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 919-926, 2012.

[3]M. Kumar, A. Bala, "Analyzing Twitter sentiments through big data", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 2628-2631, 2016.

[4]Y. Bengio, R. Ducharme, P. Vincent, F. Morin, C. Jauvin, "Neural probabilistic language models", Journal of Machine Learning Research, pp. 1137-1155, 2003.

[5] W. Wunnasri, T. Theeramunkong, C. Haruechaiyasak, "Solving unbalanced data for Thai sentiment analysis", Proceedings of the 2013 10th International Joint Conference on Computer Science and Software Engineering JCSSE 2013, pp. 200-205, 2013.

[6] Cheng-Ying Liu, Ming-Syan Chen, Chi-Yao Tseng, "IncreSTS: Towards Real-Time Incremental Short Text Summarization on Comment Streams from Social Network Services," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 11, 1 Nov 2015.

```
Train on 1376 samples, validate on 345 samples
Epoch 1/5
1376/1376 [==============================] - 8s 6ms/step - loss:
0.0387 - acc: 0.9906 - val_loss: 1.7780 - val_acc: 0.6667
Epoch 2/5
```

```
1376/1376 [==============================] - 9s 7ms/step - loss:
0.0339 - acc: 0.9927 - val_loss: 3.5589 - val_acc: 0.5594
Epoch 3/5
1376/1376 [==============================] - 9s 6ms/step - loss:
0.1038 - acc: 0.9680 - val_loss: 1.7797 - val_acc: 0.6348
Epoch 4/5
1376/1376 [==============================] - 8s 6ms/step - loss:
0.0294 - acc: 0.9920 - val_loss: 1.6631 - val_acc: 0.6783
Epoch 5/5
1376/1376 [==============================] - 8s 6ms/step - loss:
0.0297 - acc: 0.9935 - val_loss: 1.7087 - val_acc: 0.6493
```