การสร้างและทำการพัฒนาตัวตัดประโยคทางภาษาไทย
โดยใช้เทคนิคการเรียนรู้เชิงลึก

สรทรรศน์ ศิริรัตนจักริน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่
วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2564

DEEP LEARNING-BASED MODELS FRAMEWORK FOR THAI LANGUAGE SENTENCE SEGMENTER

SORRATAT SIRIRATTANAJAKARIN

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Big Data Engineering,

College of Innovative Technology and Engineering,

Dhurakij Pundit University

2021

ใบรับรองงานวิทยานิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การสร้างและทำการพัฒนาตัวตัดประโยคทางภาษาไทย โดยใช้เทคนิคการเรียนรู้เชิงลึก

เสนอโดย นายสรทรรศน์ ศิริรัตนจักริน

สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.ดวงใจ จิตคงชื่น

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(ดร.สรรพฤทธิ์ มฤคทัต)

.....กรรมการและอาจารย์ที่ปรึกษาหลัก

(ผู้ช่วยศาสตราจารย์ ดร.ดวงใจ จิตคงชื่น)

.....กรรมการและอาจารย์ที่ปรึกษาร่วม

(ผู้ช่วยศาสตราจารย์ ดร.พีระศักดิ์ อินทรไพบูลย์)

.....กรรมการ

(ดร.ธนภัทร ฆังคะจิตร)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

.....

(ดร.ชัยพร เขมะภาตะพันธ์)

คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

วันที่ 31 เดือน กรกฎาคม พ.ศ. 2564

หัวข้อวิทยานิพนธ์	การสร้าง และทำการพัฒนาตัวตัดประโยคทางภาษาไทย โดยใช้เทคนิคการเรียนรู้เชิงลึก
ชื่อผู้เขียน	สรทรรศน์ ศิริรัตนจักริน
อาจารย์ที่ปรึกษาหลัก	ผศ.ดร. ดวงใจ จิตคงชื่น
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.พีรศักดิ์ อินทรไพบูลย์
สาขาวิชา	วิศวกรรมข้อมูลขนาดใหญ่
ปีการศึกษา	2563

บทคัดย่อ

ข้อมูลทางภาษาเป็นข้อมูลประเภทหนึ่งที่สำคัญอย่างยิ่งในการต่อยอดทำแอพพลิเคชั่นต่างๆ อาทิเช่น การทำการสรุปข่าว การทำการแปลภาษา หรือการทำองค์ความรู้ในรูปแบบเครือข่ายของข้อมูล ซึ่งแอพพลิเคชั่นที่กล่าวมานั้นล้วนจำเป็นต้องใช้ข้อมูลทางภาษาที่เป็นระดับของประโยคเพื่อช่วยในการสร้าง และทำให้มีประสิทธิภาพ สำหรับภาษาไทยนั้น การให้ได้ซึ่งมาของข้อมูลระดับประโยคเป็นเรื่องที่ท้ายทายอย่างยิ่ง ทั้งนี้เพราะหลักไวยกรณ์ทางภาษาไทย ไม่มีกฎที่ชัดเจนเพื่อใช้ในการระบุตำแหน่งหยุดของประโยค ไม่เหมือนกับภาษาอังกฤษ ที่ระบุแต่ละประโยคได้ด้วยเครื่องหมายจุด จากปัญหาดังกล่าวที่เกิดขึ้นการสร้างกฎในการตัดประโยคนั้นมีความซับซ้อนมากในโครงสร้างเชิงภาษา อีกทั้งข้อมูลในปัจจุบันไม่ได้มาจากการเขียนเชิงวิชาการ หรือบทความเพียงเท่านั้น แต่ในปัจจุบันยังมีข้อมูลอีกจำนวนมากที่มาจากช่องทาง เครือข่ายสังคมออนไลน์ต่างๆ โดยที่มีโครงสร้างทางภาษาที่มีหลักไวยกรณ์ไม่เหมือนเดิม ดังนั้นงานวิจัยนี้จึงมีจุดประสงค์เพื่อมุ่งเน้นในการสร้าง และพัฒนาโมเดลโดยใช้เทคนิคการเรียนรู้เชิงลึกเพื่อใช้ในการตัดประโยคภาษาไทย โดยทำการทดลองในรูปแบบต่างๆกัน ทั้งในมุมของสถาปัตยกรรมของโมเดลการเรียนรู้เชิงลึกในรูปแบบต่างๆ รูปแบบการสร้างเวกเตอร์ของภาษาด้วยวิธีต่างๆ ตลอดจนพารามิเตอร์ที่ดีที่สุด เพื่อให้ได้กรอบแนวคิดของโมเดลที่ดีที่สุดสำหรับการใช้งานเพื่อตัดประโยคภาษาไทย

Thesis Title	DEEP LEARNING-BASED MODELS FRAMEWORK FOR THAI LANGUAGE SENTENCE SEGMENTER
Author	Sorratat Sirirattanajakarin
Thesis Advisor	Asst.Prof.Dr. Duangjai Jitkongchuen
Co-Thesis Advisor	Asst.Prof.Dr. Peerasak Intarapaiboon
Department	Big Data Engineering
Academic Year	2020

ABSTRACT

Text data is one of the essential data types used to build high-level applications such as News Summarization, Machine Translation, or Knowledge Graph. Those applications require a piece of a sentence to create an application and make the application more efficient. In the Thai language, to get a sentence is very challenging. Unlike in the English language, each sentence has a sentence boundary by using a full stop sign. With this problem, detecting a Thai sentence boundary by creating a rule is very complicating.

Moreover, text in recent days is not only generated in books or academic writing. We also generate different writing styles on social network platforms. As a result, the research aims to create and enhance the Deep Learning-Based Models Framework for the Thai sentence segmenter. This research is focusing on finding the optimal framework for Thai sentence segmenter by vary Architecture of deep learning model, word embedding techniques, and the best parameter for Deep Learning-Based Models Framework.

กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดีนั้นผู้วิจัยต้องขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ. ดร. ดวงใจ จิตคงชื่น และอาจารย์ที่ปรึกษาร่วม ผศ. ดร. พีระศักดิ์ อินทรไพบูลย์ ที่ช่วยกรุณาให้คำแนะนำ คำปรึกษา รวมถึงช่วยตรวจสอบ และแก้ไขงานวิจัย และร่างวิทยานิพนธ์มาโดยตลอดทั้งช่วงเวลาทำงาน และนอกเวลาทำงาน ทางผู้วิจัยต้องขอขอบพระคุณอาจารย์ไว้ ณ โอกาสนี้

ผู้วิจัยต้องขอขอบพระคุณ ผศ. ดร. สรรพฤทธิ์ มฤคทัต ที่กรุณาให้เกียรติเป็นประธานโดยมี ผศ. ดร. ดวงใจ จิตคงชื่น และอาจารย์ที่ปรึกษาร่วม ผศ. ดร. พีระศักดิ์ อินทรไพบูลย์ เป็นกรรมการในการสอบวิทยานิพนธ์ ตลอดจน นางสาวกุลธิดา รอดบุญ รวมถึงเจ้าหน้าที่ บัณฑิตมหาวิทยาลัยธุรกิจบัณฑิตย์ ทุกท่าน ที่ช่วยอำนวยความสะดวก และประสานงานในเรื่องต่างๆ ตลอดจนค้นคว้าหาข้อมูลในการจัดทำวิทยานิพนธ์ของผู้วิจัยจนสำเร็จลุล่วงไปได้ด้วยดี

ยิ่งไปกว่านั้นทางผู้วิจัยต้องขอขอบพระคุณ ดร. บุญทวี สันติศรีวราภรณ์ และ ผศ. ดร. พีระศักดิ์ อินทรไพบูลย์ ที่เป็นเพื่อนร่วมงานในขณะที่ทำโครงการร่วมกันที่บริษัท ธนาคารไทยพาณิชย์ โดยให้การช่วยเหลือผู้วิจัยทั้ง งานที่ทำประจำ งานวิจัยในที่ทำงาน และยังช่วยจุดประกายในการทำงานวิจัยนี้อีกด้วย รวมถึงขอบขอบพระคุณบริษัท 3dsinteractive co. ltd. ที่ได้มอบโอกาสในการทำงานด้านวิทยาศาสตร์ข้อมูล เป็นที่แรกของการทำงานอีกด้วย

นอกจากนี้ทางผู้วิจัยขอขอบพระคุณงานวิจัยต่างๆ และผู้วิจัยทุกท่านที่สร้างสรรค์งานวิจัยออกมาเพื่อเป็นแรงบันดาลใจในการต่อยอดงานวิจัยของทางผู้วิจัย อีกทั้งขอขอบพระคุณอาจารย์ ดร. เอกสิทธิ์ พัชรวงศ์ศักดา ที่มอบโอกาสทั้งด้านการงาน ด้านการเงิน รวมถึงคำปรึกษาที่มีคุณค่าในการเรียน และขอขอบพระคุณ อาจารย์ ดร.ธนภัทร ฆังคะจิตร ที่ช่วยให้คำแนะนำ และชี้แนะในการเรียน ตลอดจนการสอบวิทยานิพนธ์

สุดท้ายนี้ทางผู้วิจัยขอขอบพระคุณคุณแม่ อิสรีย์ ศิริรัตนจักริน ผู้ให้กำเนิด คอยสนับสนุน ชี้แนะ มอบความกล้า และให้กำลังใจตลอดการทำงานวิจัย และนอกเวลาการทำวิจัยตลอดทั้งเป็นผู้ผลักดันอยู่เบื้องหลัง ให้งานวิจัยนี้สำเร็จลุล่วงจนสำเร็จการศึกษา

สรทรรศน์ ศิริรัตนจักริน

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ภาษาไทยเป็นภาษาหนึ่งที่ไม่มีสิ่งที่ระบุจุดจบประโยคที่ชัดเจน ไม่เหมือนในภาษาอังกฤษที่สามารถระบุแต่ละประโยคได้โดยใช้เครื่องหมายจุด และการสร้างกฎในการตัดประโยคนั้นมีความซับซ้อนมากในโครงสร้างเชิงภาษา อีกทั้งข้อมูลในปัจจุบันไม่ได้มาจากการเขียนเชิงวิชาการ หรือบทความเพียงเท่านั้น แต่ในปัจจุบันยังมีข้อมูลอีกจำนวนมากที่มาจากช่องทางเครือข่ายสังคมออนไลน์ต่างๆ ที่มีรูปแบบการเขียนที่หลากหลาย ไม่เหมือนกับรูปแบบการเขียนเนื้อหาในเชิงวิชาการ

โดยการที่ข้อมูลทางภาษามีโครงสร้างทางภาษาที่มีหลักไวยกรณ์ไม่เหมือนเดิม และมีการเปลี่ยนแปลงอยู่ตลอดเวลาทำให้การสร้างกฎเพื่อนำมาใช้ในการตัดประโยคเป็นไปได้ยาก และใช้เวลาที่มาก อีกทั้งยังต้องอาศัยผู้เชี่ยวชาญทางด้านภาษาศาสตร์ซึ่งมีค่าใช้จ่ายที่สูงรวมถึงในหลายๆ แอพพลิเคชั่นในระดับสูง ที่มีความจำเป็นต้องใช้ข้อมูลในรูปแบบของบทความนั้นจำเป็นต้องใช้ข้อมูลทางภาษาที่ได้รับการตัดประโยคมาเป็นที่เรียบร้อยแล้ว ตัวอย่างของแอพพลิเคชั่นที่ต้องการข้อมูลในระดับประโยค ได้แก่การทำการสรุปข่าว การทำการแปลภาษา หรือการทำองค์ความรู้ในรูปแบบเครือข่ายของข้อมูล เป็นต้น

ดังนั้นหากต้องหาเครื่องมือใดเครื่องมือหนึ่งเพื่อมาใช้ในการสร้างกฎที่มีความซับซ้อนและหลากหลายกับข้อมูลทางภาษา การใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) จึงเป็นอีกทางเลือกหนึ่งที่ผู้วิจัยมุ่งเน้นศึกษาเพื่อทำการสร้างโมเดลเพื่อใช้ในการตัดประโยค โดยมุ่งเน้นไปที่ภาษาไทย และด้วยประโยชน์ของเทคนิคการเรียนรู้เชิงลึก ที่สามารถปรับเปลี่ยนโมเดลให้เหมาะสมกับข้อมูลที่เปลี่ยนแปลงไปในปัจจุบัน จึงเป็นเทคนิคหนึ่งที่เลือกนำมาใช้ในการแก้ไขปัญหาสำหรับงานวิจัยนี้

ในด้านของเทคนิคที่ใช้กันในปัจจุบันนั้น สามารถมองปัญหานี้ได้เหมือนกับโจทย์ของการประมวลผลภาษาทางธรรมชาติ ที่มีการทำนายคำตอบเป็นแบบลำดับต่อเนื่องกัน (Sequence) เช่นการทำนายขอบเขตของคำศัพท์เพื่อทำการตัดคำ (Word Segmentation) การทำนายหาที่มีชื่อเฉพาะ (Named-Entity Recognition) หรือ การทำนายหน้าที่ของคำ)Part of Speech) ซึ่งปัญหา

เหล่านี้เป็นการมองข้อมูลเป็นลำดับในระดับคำ และในระดับตัวอักษร ตัวอย่างโมเดลที่ประสบความสำเร็จในการตัดคำของภาษาไทย อาทิเช่น (Deepcut, 2019), (Sertis, 2017) และ (AttaCut, 2019) ซึ่งแต่ละโมเดลมีคุณลักษณะที่แตกต่างกันออกไป ทั้งสามโมเดลล้วนสร้างขึ้นด้วยเทคนิคการเรียนรู้เชิงลึกซึ่งสามารถนำโมเดลไปปรับปรุงให้ดีขึ้นต่อได้

งานวิจัยที่ผ่านมาสำหรับ การทำนายขอบเขตของคำศัพท์ (Word Segmentation) การทำนายหาคำนามที่มีชื่อเฉพาะ (Named-Entity Recognition) หรือ การทำนายหน้าที่ของคำ)Part of Speech) นั้นล้วนแล้วแต่ใช้เทคนิคการเรียนรู้เชิงลึกที่มีความแตกต่างกันออกไป สำหรับการตัดคำนั้น โมเดลส่วนใหญ่เน้นไปที่การใช้เทคนิค โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network หรือ CNN) ซึ่งการตัดคำให้ได้นั้นจำเป็นต้องมุ่งเน้นไปยังระดับของตัวอักษร ต่างจากงานที่ต้องทำนายระดับคำว่าคำไหนเป็นคำนามที่มีชื่อเฉพาะ หรือทำนายหน้าที่ของคำ ซึ่งมุ่งเน้นไปยังโมเดลที่ใช้เทคนิค ลองชอตเทอมเมมโมรี่ชนิดสองทาง (Bidirectional Long Short-Term Memory หรือ BiLSTM) ตัวอย่างเช่นงานวิจัยของ (J. P. Chiu, 2015) ใช้ BiLSTM ร่วมกับ CNNs เพื่อทำการรู้จำชื่อเฉพาะ, (G. Lample, 2016) และ (B. Y. Lin, 2017) ใช้ BiLSTM ร่วมกับแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส (Conditional Random Field หรือ CRF) เพื่อทำนายคำนามที่มีชื่อเฉพาะ และ (X. Ma, 2016) ที่ใช้เทคนิคร่วมกันระหว่าง ลองชอตเทอมเมมโมรี่ (Long Short-Term Memory หรือLSTM), CNN และ CRF ร่วมกันเพื่อแก้ปัญหาเดียวกัน

ดังนั้นงานวิจัยฉบับนี้จึงมุ่งเน้นไปที่การสร้าง และพัฒนาตัวตัดประโยคภาษาไทยโดยใช้โมเดลเทคนิคการเรียนรู้เชิงลึกในรูปแบบต่างๆมาสร้างเป็นกรอบแนวคิดใหม่ โดยทำการออกแบบการทดลองไปที่การค้นหารูปแบบของโมเดล รวมถึงชนิดของเวกเตอร์ที่ใช้เป็นตัวแทนของทั้งระดับคำ และระดับตัวอักษร ที่ส่งผลให้ประสิทธิภาพการตัดประโยคภาษาไทย ให้ผลดีที่สุด รวมถึงสามารถนำไปใช้งานต่อยอดในระดับ แอพพลิเคชั่นได้จริงในอนาคต

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาโมเดลโดยใช้เทคนิคการเรียนรู้เชิงลึกเพื่อตัดประโยคภาษาไทย

1.3 ขอบเขตงานวิจัย

1.3.1 งานวิจัยนี้ทำการทดลองผ่านผลิตภัณฑ์ Google Colaboratory Pro Version

1.3.2 งานวิจัยนี้ทำขึ้นจากข้อมูลที่สามารถแทนรูปแบบของประโยค Orchid, scb-mt-en-th-2020, และ LST-20

1.4 สมมติฐานงานวิจัย

1.4.1 การสร้างโมเดลตัวตัดประโยคภาษาไทย โดยใช้โมเดลร่วมกันระหว่างโครงข่ายประสาทแบบคอนโวลูชัน ลองชอตเทอมเมมโมรี่ชนิดสองทาง และแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ ให้ประสิทธิภาพในการตัดคำแม่นยำกว่าการโมเดลที่ใช้เพียงแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ เพียงเท่านั้น

1.4.2 โมเดลตัดประโยคภาษาไทย ที่ใช้เวกเตอร์เพื่อใช้แทนคำศัพท์ที่สร้างจากโมเดลทางภาษาโดยใช้เทคนิค คอนตินิวอัสแบคออฟเวิด (Continuous Bag-of-Words, CBOW) และสคริปแกรม (Skip-Gram) ช่วยเพิ่มประสิทธิภาพโมเดลการตัดประโยค มากกว่าการใช้เวกเตอร์เพื่อใช้แทนคำศัพท์ที่สร้างร่วมกันพร้อมกับโมเดลตัดประโยคภาษาไทย

1.4.3 การใช้ข้อมูล คำนามที่มีชื่อเฉพาะ (Named-Entity Recognition) หรือ การทำนายหน้าที่ของคำ)Part of Speech) รวมถึง คลังข้อมูลทางภาษา TCC (Thai character cluster) มาช่วยในการตัดสินใจให้กับโมเดล สามารถเพิ่มประสิทธิภาพในการตัดคำแม่นยำกว่าการที่ใช้เพียงข้อมูลในเฉพาะคำศัพท์เท่านั้น

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถสร้างโมเดลที่ใช้ในการตัดประโยคภาษาไทยที่มีประสิทธิภาพได้

1.5.2 สามารถนำไปต่อยอดสร้างแอพพลิเคชั่นระดับสูงได้เช่น การทำการสรุปข่าว การทำการแปลภาษา หรือการทำองค์ความรู้ในรูปแบบเครือข่ายของข้อมูล และเป็นการเพิ่มมูลค่าให้กับทางธุรกิจอีกมหาศาล

1.5.3 นำไปใช้เป็นซอฟต์แวร์ลิขสิทธิ์ที่มีไลเซนส์แบบโอเพนซอร์ส เพื่อให้ทั้งภาคธุรกิจและภาคประชาชนสามรถนำไปประยุกต์ใช้ในงานวิจัย หรือต่อยอดในเชิงพาณิชย์ได้

1.5.4 สร้างผลกระทบที่ดีในงานวิจัยด้านการประมวลผลภาษาทางธรรมชาติของภาษาไทย

1.6 นิยามศัพท์

1.6.1 **ตัวตัดประโยค (Sentence Segmenter)** หมายถึงโมเดลที่สามารถทำการค้นหาจุดที่ใช้แบ่งประโยคได้

1.6.2 **องค์ความรู้ในรูปแบบเครือข่ายของข้อมูล (Knowledge Graph)** หมายถึงโครงสร้างของข้อมูลที่มีรูปแบบความสัมพันธ์ของข้อมูลเป็นประเภท เอนทิตี้ หรือสิ่งของ หรือคำนาม หรือสิ่งที่เราสนใจ ซึ่งมีรูปแบบความสัมพันธ์กับอีกสิ่งหนึ่งด้วย การกระทำใดการกระทำหนึ่ง เพื่อใช้ในการเชื่อมความสัมพันธ์

1.6.4 **การสรุปข่าว (News Summarization)** หมายถึง การย่อความหรือเนื้อหาของเอกสารหรือข่าวให้มีขนาดลดลง โดยยังคงไว้ซึ่งเนื้อหาที่มีใจความสำคัญอยู่

1.6.5 **การรู้จำชื่อเฉพาะ (Named-Entity Recognition)** หมายถึง การจดจำชื่อเฉพาะชนิดต่างๆเช่น ชื่อคน ชื่อบริษัท วันที่ หรือตัวเลขที่บอกถึงจำนวนเงิน เป็นต้น

1.6.6 **การเรียนรู้เชิงลึก (Deep Learning)** หมายถึงรูปแบบการเรียนรู้ของเครื่องรูปแบบหนึ่งที่มีการจำลองระบบการเรียนรู้อ้างอิงจากระบบเครือข่ายประสาทของมนุษย์ โดยมีจำนวนชั้นของการเรียนรู้ที่มีหลากหลายชั้นการเรียนรู้

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 ความเป็นมาของตัดคำภาษาไทย

ภาษาไทยเป็นหนึ่งในภาษาที่เขียนติดกัน และไม่มีสิ่งที่บอกถึงขอบเขตของแต่ละคำอย่างชัดเจน ไม่เหมือนกับในภาษาอังกฤษที่สามารถระบุขอบเขตของแต่ละคำได้โดยใช้เครื่องหมายเว้นวรรค คำในภาษาไทยนั้นประกอบขึ้นด้วยการผสมผสานกันของ พยัญชนะ และรูปสระ โดยที่ในภาษาไทยนั้นมีพยัญชนะรวมทั้งสิ้น 44 ตัว และรูปสระทั้งหมด 21 ตัว ดังนั้นในการตัดคำงานวิจัยที่เกี่ยวข้องจึงมุ่งเน้นไปยังระดับของตัวอักษร (Character) และด้วยข้อมูลภาษาไทยที่มีอยู่ในปัจจุบันเกิดจากหลายที่มา และมาจากหลายนักเขียน ซึ่งมีสไตล์การเขียนที่แตกต่างกันไป ดังนั้นการตัดคำจึงเป็นเรื่องที่ท้าทายมากยิ่งขึ้น หากต้องสร้างกฎเพื่อทำการจำแนกแต่ละคำออกจากกัน

สำหรับภาษาไทยนั้นคำ 1 คำสามารถถูกแบ่งเพื่อเกิดคำที่ให้ความหมายใหม่ได้ หากเกิดการแบ่งคำที่คนละจุดกัน ยกตัวอย่างเช่นคำว่า ตากลม (Round eyes) แบ่งได้เป็น 2 แบบ คือ ตา (Eye) + กลม (Round) และ ตากลม (Weather) = ตาก (Dry) + ลม (Wind) หรือตัวอย่างเช่น ตู้เสื้อผ้าสีขาว ซึ่งสามารถตัดออกมาได้เป็น ตู้_เสื้อผ้าสีขาว (Closet for white clothes) และ ตู้เสื้อผ้า_สีขาว (Clothes closet that is white) ดังนั้นความยากในการตัดคำจึงไม่ได้มาจากเพียงรู้ตำแหน่งของจุดที่ทำการตัด แต่ยังขึ้นกับบริบทของคำอีกด้วยว่าสื่อถึงอะไร (Arronmanakun, 2007)

งานวิจัยยุคแรกๆที่ทำการศึกษาในการตัดคำนั้น ได้มีการนำเสนอกฎต่างๆขึ้นมาเพื่อบ่งบอกถึงจุดที่ทำการตัดแบ่งคำได้ และจุดที่ไม่สามารถตัดแบ่งคำได้ ยกตัวอย่างเช่น หากพบพยัญชนะ “ไ” ก็จะไม่สามารถตัดคำหลังสระตัวนี้ได้ จำเป็นต้องมีพยัญชนะตามมาเสมอ (Thairatananond, 1981)

สำหรับการตัดคำในภาษาไทยนั้น ในยุคต่อมานักวิจัยได้เสนอแนวคิดในการทำพจนานุกรมเพื่อทำการค้นหาคำแล้วทำการดูว่ามีคำไหนที่เหมือนกับในพจนานุกรมบ้าง ถ้าเหมือนด้วยคำที่ยาวที่สุดที่มีอยู่ ก็ให้ถือว่าเป็นการตัดที่คำคำนั้น วิธีนี้เราเรียกว่า Longest matching (Poowarawan, 1986) แต่ปัญหาต่อมาก็คือคำบางคำเมื่อใช้ Longest matching แล้วนั้นทำให้คำที่ได้นั้นไม่เกิดความหมายที่ถูกต้อง เช่น “ไปหามเหสี” เมื่อทำการตัดด้วย Longest matching แล้วจะได้

ไป|หาม|เห|สี ทั้งๆที่ควรจะเป็น “ไป|หา|มเหสี” ดังนั้นแนวคิดต่อมาแทนที่จะทำการเลือกคำที่ยาวที่สุดออกมา ก็เปลี่ยนเป็นเลือกคำที่เมื่อทำการ Maching แล้วได้ปริมาณคำน้อยที่สุด หรือเรียกวิธีนี้ว่า Maximum matching (Sornlertlamvanich, 1993)

ภายหลังการทำพจนานุกรมเพื่อใช้ในการเทียบคำแล้ว นักวิจัยได้นำแนวคิดทางสถิติมาคำนวณการเกิดขึ้นของคู่คำ (N-gram) เพื่อใช้แก้ปัญหาความกำกวมของคำ (Kawtrakul et al., 1995) แต่เทคนิคนี้ก็ยังจำเป็นต้องตัดคำให้ได้คลังคำศัพท์ของจำนวน N-gram ที่มากพอ นอกจากการใช้ข้อมูลที่เป็นพจนานุกรมมาช่วยแล้วนั้น นักวิจัยยังนำฟีเจอร์อื่นมาช่วยในการตัดสินใจว่าควรตัดคำนี้ดีหรือไม่ เพื่อช่วยให้เกิดการตัดคำที่ดีขึ้น (Meknavin et al., 1997) นอกจากนี้ยังมีการสร้าง หน่วยย่อยเพื่อเป็นตัวตัดสินใจก่อนว่าขอบเขตของการตัดคำได้นั้นมีอยู่ตำแหน่งไหนบ้าง โดยอาศัย TCC (Thai character cluster) ซึ่งช่วยบอกว่าจุดไหนไม่ควรตัด ทำส่วนที่ทำการเลือกจัดตัดคำนั้นมีจำนวนน้อยลง

ยุคต่อมาของการตัดคำภาษาไทย นักวิจัยได้ทำการใช้เทคนิคการเรียนรู้ของเครื่องอย่างแพร่หลาย โมเดลที่ได้รับความนิยม และมีประสิทธิภาพอย่างยิ่งคือ แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส (Conditional Random Field หรือ CRF) ด้วยความที่การทำนายจุดตัดของแต่ละคำนั้น เป็นการไล่ดูในระดับของตัวอักษร ดังนั้นเทคนิคหนึ่งที่ถูกนำมาใช้คือ เทคนิคโครงข่ายประสาทแบบรีเคอร์เรนท์ (Recurrent Neural Network) ซึ่งโมเดลมีความสามารถในการเรียนรู้ข้อมูลเป็นลำดับโดยรู้ว่าอักษรไหนมาก่อน และอักษรไหนมาหลัง (Sertis, 2017) นอกจากนั้นยังมีนักวิจัยที่ได้นำความสามารถของ โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network หรือ CNN) มาใช้ช่วยในการตัดคำ

2.2 ประโยคภาษาไทย

สำหรับภาษาไทย ประโยค หมายถึงข้อความที่มีทั้งส่วนที่แสดงถึงประธาน และส่วนที่แสดงออก มีใจความสมบูรณ์ รู้ได้ว่าใครทำอะไร ที่ไหน อย่างไร ซึ่งประโยคในภาษาไทยสามารถแบ่งออกได้เป็น 3 ประเภทใหญ่

2.2.1 ประโยคความเดียว หรือ เอกัตถประโยค

ประโยคความเดียว หรือ เอกัตถประโยค เป็นประโยคที่มีประธานแค่บทเดียว และบทกริยาเพียงบทเดียวเช่นกัน ยกตัวอย่างเช่น เธอเก่งมาก, ฉันนั่งอยู่ริมสระน้ำ, เจ้าพวกมิเนี่ยนกำลังรวมตัวกัน

2.2.2 ประโยคความรวม (อเนกัตถประโยค)

ประโยคความรวม หรือ อเนกัตถประโยค เป็นประโยคที่รวมประโยคความเดียวตั้งแต่ 2 ประโยคขึ้นไปเข้าด้วยกัน โดยมีคำสันธานเป็นตัวเชื่อม ยกตัวอย่างเช่น เธอเก่งมาก_และ_เธออีกคนก็เก่งมากด้วยเช่นกัน, ฉัน_และ_เขานั่งอยู่ริมสระน้ำ, เจ้าพวกมินเนี่ยน_และ_เจ้าตูบกำลังโบกมือ

2.2.3 ประโยคความซ้อน (สังกรประโยค)

ประโยคความซ้อน (สังกรประโยค) เป็นประโยคที่มีประโยคความเดียวเป็นประโยคหลัก แล้วมีประโยคความเดียวเข้ามาช่วยเสริม โดยที่ ประโยคหลัก (มุขยประโยค) กับ ประโยคย่อย (อนุประโยค) ของประโยคความซ้อน มีน้ำหนักที่ไม่เท่ากัน ยกตัวอย่างเช่น เขากำลังชม_ผู้ชายที่กำลังโบกมืออยู่, ฉันกำลังดูข่าว_ที่มีผู้ชายและผู้หญิงนั่งอยู่ที่ริมสระ, เจ้าตูบกำลังกระดิกหาง_อยู่บนเหล่ามินเนี่ยนที่กำลังโกรธเกรี้ยว ทั้งนี้ประโยคความซ้อนยังสามารถแบ่งได้อีก 3 ประเภทย่อยดังต่อไปนี้

2.2.3.1 ประโยคความซ้อนที่ประโยคย่อยทำหน้าที่เหมือนคำนาม (นามานุประโยค)

ยกตัวอย่างเช่น ฉันกำลังดูข่าว_***ที่มีผู้ชายและผู้หญิงนั่งอยู่ที่ริมสระ***/[กรรม] , **คนที่กินภาษี**/[ประธาน] _เป็นคนเอาเปรียบผู้อื่น

2.2.3.2 ประโยคความซ้อนที่มีประโยคย่อยทำหน้าที่คล้ายคำวิเศษณ์ขยายคำนามหรือขยายสรรพนาม และมีสันธาน ที่ ซึ่ง อัน เป็นเครื่องเชื่อม

ยกตัวอย่างเช่น เจ้าพวกมินเนี่ยน_***ที่_กำลังโบกมือ***โปรดมารับกล้วย, ฉันเห็นบ้านเมือง_***ซึ่ง_กำลังค่อยๆจางหาย***ไปผ่านทางโทรทัศน์ โดยคำเชื่อมประโยคหลัก และประโยคย่อยเราสามารถเรียกอีกชื่อหนึ่งได้ว่า ประพันธสรรพนาม หรือสรรพนามเชื่อมประโยค

2.2.3.3 ประโยคความซ้อนที่มีประโยคหลักและประโยคย่อย และประโยคย่อยนั้น ๆ อาจทำหน้าที่เหมือนคำนามก็ได้ ทำหน้าที่เหมือนคำวิเศษณ์ก็ได้ จะมีสันธาน เมื่อ, จน, เพราะ, ตาม, ราวกับ, ให้, ทว่า, ระหว่างที่, เพราะเหตุว่า, เหมือน, ดุจดัง, เสมือน, ฯลฯ เป็นตัวเชื่อม

ยกตัวอย่างเช่น เราจะไม่ยอมถอย_***เพียงเพราะ***_ได้ยินเสียงปืน, ฉันกำลังเขียนวิทยานิพนธ์_***เพราะ***_ต้องการที่จะเรียนจบ

2.3 ประโยคภาษาอังกฤษ

เพื่อความเข้าใจในรูปแบบของประโยคที่มากขึ้นนักวิจัยได้มีการศึกษาเพิ่มเติมถึงรูปแบบประโยคของภาษาอังกฤษที่มีรูปแบบของขอบเขตประโยคที่ชัดเจน โดยประโยคในภาษาอังกฤษ มีโครงสร้างมาจากการรวมกัน หรือการอยู่แบบเดี่ยวของอนุประโยคอิสระ (Independent clause) และอนุประโยคแบบไม่อิสระ (Dependent clause) โดยที่ อนุประโยคอิสระ เป็นประโยคที่สามารถอยู่

ได้ด้วยตัวเอง โดยที่ยังสามารถสื่อความหมายได้ ไม่ต้องพึ่งพาประโยคอื่นเพิ่มเติม ยกตัวอย่างเช่น You are minion คุณคือมิเนี่ยน, I sit near a pool ฉันนั่งริมสระ เป็นต้น ส่วนอนุประโยคแบบไม่อิสระ หมายถึงประโยคที่ไม่สามารถสื่อสารได้อย่างสมบูรณ์หากต้องอยู่ลำพัง จำเป็นต้องไปอยู่ร่วมกับ อนุประโยคแบบอิสระ เพื่อให้ความหมายสมบูรณ์ขึ้น ยกตัวอย่างเช่น I hear ฉันได้ยิน, when you are lonely เมื่อคุณเหงา เป็นต้น สำหรับการแบ่งประเภทประโยคของภาษาอังกฤษสามารถแบ่งออกได้เป็น 4 ประเภท (Aroonmanakun, 2007)

2.3.1 ประโยคความเดียว หรือ simple sentence

ประโยคความเดียว หรือประโยคที่ประกอบด้วย อนุประโยคแบบอิสระ เพียง 1 ประโยคเท่านั้น ยกตัวอย่างเช่น You are minion คุณคือมิเนี่ยน, I sit near a pool ฉันนั่งริมสระ, you are so brave คุณกล้ามาก

2.3.2 ประโยคความรวม หรือ compound sentence

ประโยคความรวม หรือ อเนกัตถประโยค รูปแบบประโยคคล้ายกับประโยคภาษาไทยที่จำเป็นต้องมีคำเชื่อมระหว่างอนุประโยคแบบไม่อิสระตั้งแต่ 2 ประโยคขึ้นไป ยกตัวอย่างเช่น I sit but you stand near a pool ฉันนั่งแต่เธอยืนอยู่ริมสระ, I am minion and I love banana ฉันเป็นมินเนี่ยนและฉันชอบกล้วย เป็นต้น

2.3.3 ประโยคความซ้อน หรือ complex sentence

ประโยคความซ้อน คือประโยคที่มี อนุประโยคแบบไม่อิสระเพียง 1 ประโยคเท่านั้น แต่สามารถมีอนุประโยคแบบอิสระตั้งแต่ 1 ประโยคขึ้นไป รูปแบบคล้าย สังกรประโยคในภาษาไทยที่มีความซ้อนกันของอนุประโยคแบบไม่อิสระ ยกตัวอย่างเช่น I know ***what you did there last night*** ฉันรู้นะว่าแกทำอะไรเมื่อคืนที่นั่น, I know the man ***who live in German*** ฉันรู้จักคนที่อาศัยอยู่ที่เยอรมัน

2.3.4 ประโยคความรวมความซ้อน หรือ compound-complex sentence หรือ matrix sentence

สำหรับประโยคความรวมความซ้อน จะมีอนุประโยคแบบไม่อิสระตั้งแต่ 2 ประโยคขึ้นไป รวมถึงมีอนุประโยคแบบอิสระตั้งแต่ 1 ประโยคขึ้นไป รูปแบบคล้ายกับประโยคความซ้อน แต่มีการเพิ่มขึ้นมาของจำนวนอนุประโยคแบบไม่อิสระ ยกตัวอย่างเช่น I know ***what you spent our tax***, so I will tell to everyone. ผมรู้ว่าคุณเอาภาษีของพวกเราไปใช้ทำอะไรดังนั้นผมจะไปบอกทุกคน เป็นต้น

2.4 ความเป็นมาของการตัดประโยคภาษาไทย

เช่นเดียวกับการตัดคำในภาษาไทย การตัดประโยคภาษาไทยก็เป็นปัญหาที่ท้ายทายนักวิจัยอย่างยิ่ง ทั้งนี้เพราะไม่มีสิ่งที่ระบุขอบเขตของประโยคที่ชัดเจน และถึงบางครั้งจะใช้การเว้นวรรคเพื่อแบ่งประโยค ประโยคที่ได้นั้นก็ไม่ได้สื่อถึงความเป็นประโยคที่ชัดเจน ไม่เหมือนกับในภาษาอังกฤษที่สามารถระบุขอบเขตของประโยคได้ด้วยเครื่องหมายจุด (Full stop) หรือ เครื่องหมายตกใจ (Exclamation) หรือ เครื่องหมายอะไรเอ่ย (Question mark) ซึ่งสามารถสร้างกฎแบบง่ายๆได้ผ่านการใช้ภาษา เรกูล่าเอกเพชชั่น (Regular expression) (Palmer, 2000)

สำหรับปัญหาการตัดประโยคภาษาไทยนั้นเนื่องจากมีความซับซ้อนสูงในการค้นหาจุดที่ใช้ในการแบ่งประโยค และเนื่องจากความซับซ่อนดังกล่าวนี้ จึงเป็นที่มาว่างานวิจัยในการทำโมเดลทางสถิติเพื่อใช้ในการตัดประโยคภาษาไทยจึงทำได้ยากยิ่งนัก รวมถึงติดคำตอบให้กับประโยคว่าจุดไหนเป็นจุดจบประโยค หรือจุดไหนเป็นจุดเริ่มต้นประโยคนั้นทำได้ยาก ทำให้ขาดแหล่งข้อมูลเพื่อใช้ในการสอนหรือทำโมเดลเพื่อสอนเครื่องจักรเพื่อโมเดลตัดประโยคภาษาไทย

ในปี 1995 (Longchupole, 1995) ได้มีงานวิจัยเพื่อทำตัวตัดประโยคภาษาไทย โดยการใช้เงื่อนไขเข้ามาช่วย ผู้วิจัยทำการตัดประโยคภาษาไทยจากบทความ (Paragraph) เป็นหน่วยทางภาษาที่มีขนาดเล็กที่สุดที่ยังคงความหมาย และหน้าที่ทางไวยกรณ์อยู่ได้ด้วยตัวเอง (Morphemes) โดยใช้คำกริยาหลักเพื่อทำการประมาณการจำนวนประโยคที่มีในบทความ จากนั้นใช้ตัวเชื่อมประโยค (Conjunction) มาทำหน้าที่เป็นตัวที่ใช้ตัดประโยค ผลลัพธ์ของงานวิจัยสามารถตัดประโยคภาษาไทยได้ด้วยความแม่นยำถึง 81.8% แต่ข้อเสียของวิธีการนี้คือเรื่องของขนาดบทความ ที่ต้องทำการวิเคราะห์เป็นรายบทความ ทำให้ส่งผลต่อการประมวลผลบทความที่มีขนาดใหญ่

เปรียบเทียบงานวิจัยในการสร้างโมเดลสำหรับการตัดประโยคภาษาอังกฤษที่มีการใช้โมเดลทางสถิติมาใช้ในการตัดประโยคภาษาอังกฤษ ในปี 1998 (Riley, 1989) ผู้วิจัยได้ใช้โมเดล CART (Classification and Regression Tree) โดยทำการสอนโมเดลด้วยข้อมูลจำนวน 25 ล้านคำ หลังจากนั้นทำการทดสอบบนชุดข้อมูล บราวน์ (Brown Corpus) พบว่าเกิดค่าผิดพลาดเพียง 0.2 % ในปี 1997 (Palmer, 1997) ได้ทำการทดลองสร้างโมเดลผ่าน 2 วิธี วิธีแรกใช้โมเดลเครือข่ายประสาทเทียม และวิธีที่สองคือใช้โมเดลต้นไม้ เพื่อทดสอบการค้นหาตำแหน่งที่ทำการตัดประโยค วิธีของ Palmer ได้นำข้อมูลไวยกรณ์ทางภาษา หรือ Part of speech เข้ามาช่วยในการเรียนของโมเดล ส่งผลต่องานวิจัยในการตัดประโยคภาษาไทยในปี 2000 (Mittrapiyanuruk, 2000) ได้นำเสนอโมเดลในการตัดประโยคภาษาไทยโดยใช้ ไวยกรณ์ทางภาษา หรือ Part of speech เข้ามาช่วยในการตัดประโยค และวิจัยผ่านชุดข้อมูลชื่อ ORCHID โดยที่ผลลัพธ์ของโมเดลอยู่ที่ 85.26% และตอบผิดอยู่ที่ 8.75%

ในการค้นหาจุดตัดประโยคของภาษาไทย ในปี 2007 (Aroonmanakun, 2007) ได้ทำการทดลองถึงการค้นหาจุดตัดประโยคของภาษาไทย โดยใช้แนวคิดการเทียบประโยคจากภาษาไทยไปเป็นภาษาอังกฤษแล้วจากนั้นทำการตัดประโยคจากภาษาอังกฤษ เพื่อที่จะได้ทราบตำแหน่งที่ต้องตัดประโยคของประโยคภาษาไทยตั้งต้น ทั้งนี้เนื่องจากภาษาอังกฤษมีรูปแบบของประโยคที่ชัดเจนกว่ารูปแบบของประโยคภาษาไทย ผลลัพธ์ที่ได้เป็นที่น่าประหลาดใจมาก เพราะจุดที่ทำการตัดประโยคภาษาไทยนั้น ไม่ได้ใช้เครื่องหมายเว้นวรรค แต่กลับเป็นข้อสังเกตุต่างๆดังต่อไปนี้

1. เมื่อมีการเปลี่ยนหัวข้อเกิดขึ้น ประธานใหม่จะทำหน้าที่เริ่มต้นประโยค

2. ถ้ายังอยู่ที่หัวข้อเดิมแต่ใช้วลีที่เป็นคำนาม หรือคำสรรพนาม โดยส่วนใหญ่แล้วจะเป็นการขึ้นประโยคใหม่

3. ประธานส่วนใหญ่ก็ทำหน้าที่ในการเชื่อมวลีก่อนหน้าเท่านั้น

4. หากพบวลีเช่น “และต่อมา”-“and then”, “ตลอดระยะเวลาดังกล่าวนี้”-“throughout this period”, “ในสมัยนี้”-“in this period”, “ในระยะแรก”-“in the first phase”, สิ่งเหล่านี้สามารถใช้ระบุเป็นข้อความ (discourse) ในประโยคภาษาไทยได้

5. ประโยคความรวม หรือประโยคความซ้อนที่พบตัวเชื่อม และคำถัดไปเป็นประธานมักจะเป็นการเริ่มต้นประโยคใหม่ เช่น “เพราะ”-“because”, “แต่”-“but”, “จึง”-“thus”

6. ประโยคความซ้อนที่มีคำว่า “ที่”-“that”, “ซึ่ง”-“that” ส่วนใหญ่พบว่าเป็นส่วนหนึ่งของประโยคทั้งนี้ขึ้นกับตัวของประธาน

7. ประธานบางตัวก็ทำหน้าที่เหมือนตัวเริ่มประโยคใหม่ โดยที่จะอยู่ถัดจากเครื่องหมายเว้นวรรค หรืออาจจะทำงานร่วมกันกับตัวเชื่อมประโยค (Conjunction)

จากผลลัพธ์ของการวิจัยนี้ช่วยเป็นตัวตั้งต้นในการเริ่มทำการค้นหาจุดที่ใช้ในการแบ่งประโยคทางภาษาไทย แต่ปัญหาก็คือแล้วเราต้องแปลงภาษาไทยไปเป็นภาษาอังกฤษทุกครั้งเลยอย่างนั้นหรือ ในปี 2010 นักวิจัยจากไมโคซอฟท์ได้เสนอโมเดลแมกซิมั่ม เอนโทรปี (Maximum entropy model) สำหรับตัดประโยคโดยการดูคำรอบข้างสองคำด้านซ้ายมือ และสองคำด้านขวามือ จากนั้นจึงนำชุดข้อมูลจำนวน 361,802 ประโยค รวมจุดที่มีการเว้นวรรค 911,075 ที่ มาทำการสอนการค้นหาส่วนที่เว้นวรรคเพื่อค้นหาว่าเป็นจุดตัดประโยคหรือไม่ และเมื่อเทียบโดยใช้ชุดข้อมูล ORCHID ให้ค่าความแม่นยำในการทำนายคำเว้นวรรคถูกถึง 91.19%

งานวิจัยด้านการตัดประโยคนั้น เราสามารถมองเป็นปัญหาที่เกี่ยวข้องกับการตอบคำถามแบบเป็นลำดับ (Sequential Labeling) ได้ โดยที่ทำการค้นหาคำไหนควรเป็นจุดเริ่มต้นของประโยค (B-CLS) จุดไหนควรเป็นคำที่จบประโยค (E-CLS) ซึ่งเราสามารถแบ่งการแก้ปัญหาดังกล่าวได้ออกเป็น 4 ประเภทดังต่อไปนี้

2.4.1 การใช้กฎในการทำนาย (Rule-based Approaches)

งานวิจัยที่แก้ปัญหาด้วยวิธีนี้จะมุ่งเน้นไปยังการสร้างกฎเกณฑ์ต่างๆ เพื่อใช้ในการทำนายข้อมูลที่เป็นลำดับ โดยที่ข้อดีของวิธีนี้คือเราสามารถอธิบายได้ว่าทำไม โมเดลที่เราใช้ทำนายถึงทำการทำนาย หรือตอบคำถามแบบนี้ด้วยกฎเกณฑ์ใด แต่ข้อเสียที่ตามมาคือ ถ้าเจอข้อมูลที่ไม่ได้อยู่ในกฎเกณฑ์ที่ตั้งไว้ ก็ไม่สามารถทำนายได้ และการสร้างกฎเกณฑ์ในการทำนายนั้นต้องอาศัยผู้เชี่ยวชาญในสาขานั้นๆในการร่วมสร้างกฎเกณฑ์ขึ้นมา นอกจากนี้กฎเกณฑ์ดังกล่าวยังไม่สามารถถ่ายทอดไปยังงานวิจัยต่างโดเมนได้ (Jing Li, 2020)

2.4.2 การใช้เทคนิคการเรียนรู้โดยไม่มีคำตอบให้ (Unsupervised Learning Approaches)

สำหรับเทคนิคนี้เป็นการใช้การเรียนรู้โดยไม่มีคำตอบให้กับโมเดล โดยให้โมเดลทำการจัดกลุ่มของคำตอบให้กับเราเอง โดยที่โมเดลจะทำการแยกแยะคำตอบโดยดูที่ข้อมูลที่มีลักษณะใกล้เคียงกันจะอยู่กลุ่มเดียวกันหรือทำนายว่าเป็นคำตอบเดียวกัน ซึ่งเทคนิคนี้ต่อยอดมาจากการสร้างกฎเกณฑ์ต่างๆและมีขนาดของข้อมูลที่ใหญ่เพียงพอที่จะทำการจัดกลุ่มโดยใช้สถิติเข้ามาช่วยในการประมวลผล (Jing Li, 2020)

2.4.3 การใช้เทคนิคการเรียนรู้โดยมีคำตอบให้ (Feature-based Supervised Learning Approaches)

งานวิจัยนี้จำเป็นต้องอาศัยคำตอบของข้อมูลในการส่งให้โมเดลทำการเรียนรู้รูปแบบเพื่อทำการทำนายคำตอบออกมา โดยที่งานวิจัยดังกล่าวยังคงเป็นต้องอาศัยความเชี่ยวชาญของนักวิจัยในการสกัดข้อมูลที่สำคัญออกมาจากข้อมูลที่ส่งให้โมเดลทำการเรียน ยกตัวอย่างในการทำนายคำตอบของข้อมูลทางภาษาที่เป็นลำดับ เช่นคำว่า “กิน” ควรจะแทนที่ด้วยตัวเลขอะไร หรือแทนที่ด้วยค่าหนึ่งในตารางเพื่อบอกว่ามีหรือไม่โดยอาศัยตัวเลขแทนการมีอยู่ของคำนั้นๆ ศูนย์ คือ ไม่มี และ หนึ่งคือ มีคำนั้นๆอยู่ นอกจากนี้ความแม่นยำของข้อมูลขึ้นอยู่กับเทคนิคการสกัดข้อมูลหรือการนำข้อมูลที่เกี่ยวข้องมาเพิ่มให้โมเดลทำการเรียนรู้ ดังนั้นจำเป้นต้องอาศัยประสบการณ์และความชำนาญของนักวิจัยที่เกี่ยวข้องกับข้อมูลชุดนั้นๆ

2.4.4 การใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning Techniques Approaches)

เทคนิคดังกล่าวเป็นที่นิยมอย่างแพร่หลายในปัจจุบันอันเนื่องมาจากความพร้อมของเทคโนโลยีที่ใช้ในการคำนวณ เพื่อประมวลผล และหลากหลายงานวิจัยที่ได้ปูทางมาก่อนหน้านี้ รวมถึงประโยชน์หนึ่งที่สำคัญของเทคนิคการเรียนรู้เชิงลึกคือสามารถช่วยสกัดฟีเจอร์ที่ซ่อนอยู่หรือฟีเจอร์ที่ไม่สามารถสกัดได้ด้วยวิธีดั้งเดิม (Hidden features) ด้วยประโยชน์นี้เอง ทำให้งานในการสกัดฟีเจอร์เป็นเรื่องที่ง่ายขึ้น และไม่จำเป็นต้องใช้ผู้เชี่ยวชาญที่เกี่ยวข้องกับข้อมูลในสาขานั้นๆเหมือนแต่ก่อน

2.5 ความเป็นมาของการทำระบบจดจำคำเฉพาะภาษาไทย

เช่นเดียวกันกับการสร้างระบบตัดประโยคภาษาไทย การสกัดคำเฉพาะออกมาจากคำในภาษาไทยนั้นเป็นเรื่องที่ท้าทาย ทั้งนี้เพราะในภาษาไทยยังคงเป็นหนึ่งในภาษาที่ไม่มีตัวกำหนดขอบเขตของคำ หรือประโยค เหมือนในภาษาอังกฤษ ทั้งนี้คำเฉพาะนั้นสามารถพบเจอได้หลังคำกิริยาที่มีการระบุเจาะจงเช่น กำลังประกอบ_, อยู่อาศัยที่_, ถูกตั้งไว้ที่_, ทำการฆ่า_ เป็นต้น ในปี 2012 (Tongtep, 2012) ได้นำเสนอเทคนิคการสกัดคำเฉพาะโดยใช้เทคนิคที่เรียกว่า Predicate-Oriented Relation โดยทำการหากิริยาที่แสดงถึงการกระทำในรูปแบบต่างๆ เพื่อดูความสัมพันธ์กับคำรอบข้าง โดยเทคนิคนี้สร้างประสิทธิภาพ F1-Score ในการสกัดคำเฉพาะได้ถึง 97.76 %, 99.19 %, 95.00 %, และ 93.50 % เมื่อเปรียบเทียบด้วยความสัมพันธ์ของเทคนิคในรูปแบบ กิริยา-สถานที่, สถานที่-กิริยา, กิริยา-คน และ คน-กิริยา ตามลำดับ ในการทดลองดังกล่าวผู้วิจัยไม่ได้เลือกเทคนิคการเรียนรู้เชิงลึกมาใช้งาน ในปี 2009 (Tirasaroj, 2009) ได้ทำการสร้างโมเดลเพื่อจำแนกคำเฉพาะโดยใช้โมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ต่อมาในปี 2019 ได้มีผู้วิจัยทำเทคนิคของการเรียนรู้เชิงลึกมาร่วมใช้กับโมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เกิดเป็นเทคนิคโมเดลสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่แบบสองทางจากนั้นต่อด้วยโมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อใช้ในการสร้างโมเดล เพื่อทำนายคำเฉพาะในภาษาไทย ผู้วิจัยได้ใช้ฟีเจอร์ที่สร้างขึ้นมาจากระดับคำ และระดับตัวอักษร ได้ประสิทธิภาพโมเดล F1-Score ที่ 91.65% ทำให้นักวิจัยในรุ่นต่อมาได้มีการต่อยอดจากโมเดลการเรียนรู้เชิงลึก ผนวกกับ โมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อใช้ในการแก้ปัญหาการตอบชุดข้อมูลที่เป็นลำดับ ซึ่งทางผู้วิจัยโมเดลตัดประโยคภาษาไทย ได้นำแนวคิดการสร้างโมเดลมาประยุกต์ใช้

2.6 โครงข่ายประสาทเทียม (Artificial Neural Network)

ในปัจจุบันเทคนิคการเรียนรู้เชิงลึก (Deep Learning) เข้ามีมีบทบาทสำคัญในชีวิตประจำวันของเราอย่างมากมาย ตั้งแต่การใช้งานแอพพลิเคชั่นต่างๆ เช่นการสั่งอาหาร หรือซื้อสินค้าผ่านช่องทางออนไลน์ ตลอดจนช่วยผลักดันธุรกิจในรูปแบบใหม่ รวมถึงระบบเศรษฐกิจอย่างก้าวกระโดดดังจะเห็นได้จากประเทศจีนที่ปัจจุบันมีแผนแม่บทวิจัยและพัฒนาระบบปัญญาประดิษฐ์โดยตั้งเป้าว่าจะสำเร็จในปี 2030 จากจุดนี้เองทำให้เทคนิคการเรียนรู้เชิงลึกยิ่งมีความนิยมในการวิจัย และต่อยอดเป็นงานวิจัยใหม่ๆอย่างกว้างขวางในหลากหลายแขนง

ในปี ค.ศ. 1943 Warren McCulloch และ Walter Pitts ได้นำเสนอแนวคิด โครงข่ายประสาทเอ็มพี (MP Neural) ซึ่งสร้างจากสมการคณิตศาตร์โดยได้แรงบรรดาลใจจากกลไกการทำงานของระบบประสาทในสิ่งมีชีวิต ต่อมาในปี ค.ศ. 1949 D. Hebb ได้นำเสนอการเรียนรู้ผ่านกฎ

ของ Hebb โดยกฎนั้นเรียบง่ายมากโดยที่เรานำโมเดลมาทำการเรียนรู้โดยการค่าน้ำหนักของข้อมูลที่มีการเชื่อมโยงกันจากจุดหนึ่งไปยังอีกจุดหนึ่ง ในปี ค.ศ. 1969 Minsky ได้แสดงให้เห็นถึงข้อจำกัดของการทำงานของ เพอร์เซปตรอนโมเดล ที่ไม่สามารถแก้ปัญหาง่ายๆอย่างปัญหา XOR ได้ จนกระทั่งถูกแก้ไขได้ด้วยการเพิ่มจำนวนชั้นเลเยอร์ของโมเดลเข้าไป และหลังจากนั้นเป็นเวลาถึง 16 ปีที่ไม่ได้มีการต่อยอดโมเดล จนกระทั่งในปี ค.ศ. 1985 Rumelhart และคณะได้นำเสนอวิธีการปรับค่าน้ำหนักจากการเรียนรู้เพื่อจะลดค่าความผิดพลาดที่เกิดจากการเรียนรู้ของโมเดล หรือเรียกว่า เทคนิค แบ็คพรอพาเกชั่น (Back Propagation) ซึ่งในเวลาต่อมาได้มีนักวิจัยพัฒนาสถาปัตยกรรมต่างๆขึ้นมาอีกมากมาย จากการค้นพบเทคนิคแบ็คพรอพาเกชั่น

2.6.1 สถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network หรือ CNN)

ในปี ค.ศ. 1957 Frank Rosenblatt ได้ทำการวิจัยโดยได้รับเงินสนับสนุนจาก United States Office of Naval Research โดยเขาได้พัฒนา เพอร์เซปตรอน (Perceptron) ให้สามารถทำการประมวลผลภาพที่มีขนาด 20 x 20 พิกเซล จากโฟโต้ดีเทคเตอร์ (Photodetector) ได้ โดยต่อมา Frank ตั้งชื่อว่า มาร์ควันเพอร์เซปตรอน (Mark I Perceptron) ปัญหาต่อมาของ มาร์ควันเพอร์เซปตรอน ถึงแม้ว่าจะสามารถทำการเรียนรู้ข้อมูลที่เป็นรูปภาพได้ แต่ตำแหน่งของข้อมูลไม่ได้ถูกนำไปเรียนรู้ด้วยทำให้ประสิทธิภาพของโมเดลทำงานได้ไม่ดีเท่าที่ควร เนื่องจากรูปภาพนั้นมีข้อมูลที่เชื่อมต่อกัน หรือมีความสัมพันธ์กันในจุดที่ใกล้เคียงกัน ทำให้โมเดลยังขาดในเรื่องของการจดจำตำแหน่งทำให้ในปี ค.ศ. 1982 Kunihiko Fukushima ได้เสนอสถาปัตยกรรม นีโอคอกนีทรอน (Neocognitron) โดยที่สถาปัตยกรรม นีโอคอกนีทรอน สามารถทำการเรียนรู้จดจำรูปร่าง โดยมีแต่ละเลเยอร์แยกส่วนกันไปทำงาน และจดจำข้อมูลแต่ละส่วน และจากนั้นจึงนำข้อมูลที่แยกออกไปทำการเรียนรู้กลับมาจดจำใหม่อีกครั้ง แต่เนื่องจาก นีโอคอกนีทรอน นั้นมีการใช้งานที่ใช้เวลาค่อนข้างนาน ประกอบกับ เทคนิค แบ็คพรอพาเกชั่น (Back Propagation) ยังไม่ได้ถือกำเนิด รวมถึงสถาปัตยกรรมดังกล่าว ยังไม่ได้รับการสนับสนุนที่มากพอ ทำให้ต่อมาในปี ค.ศ. 1998 Yann LeCun ได้นำเสนอ สถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน ขึ้นมาโดยมีจุดประสงค์เพื่อทำการเรียนรู้รูปที่มาจากลายมือ โดยที่สถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน นี้เองได้มีการเรียนรู้ข้อมูล และมีการปรับค่าความผิดพลาดได้ โดยอาศัย การเรียนรู้ความผิดพลาดในอดีตผ่านการปรับค่าน้ำหนัก หรือใช้วิธี เทคนิค แบ็คพรอพาเกชั่น นั่นเอง จากการถือกำเนิดของ สถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน นี้เองนำไปสู่การทำสกัดข้อมูลที่สำคัญ (Feature extraction) แบบใหม่จากการทำคอนโวลูชันตามจำนวนมิติที่ต้องการ โดยที่นักวิจัยไม่ต้องสกัดข้อมูลที่สำคัญด้วยตัวเอง

2.6.2 สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ (Long Short-Term Memory หรือ LSTM)

สถาปัตยกรรม มัลติเลเยอร์ เพอร์เซปตรอน ยังไม่มีความสามารถในการจดจำข้อมูลที่เป็นลำดับได้ (Sequential data) จากปัญหาดังกล่าวทำให้ในปี ค.ศ. 1982 John Hopfiel ได้ทำการพัฒนา ฮอบฟิลล์เนตเวิร์ค (Hopfield network) ของตนเองให้กลายเป็น โครงข่ายประสาทเทียมชนิดรีเคอเรนท์ (Recurrent neural network, RNN) โดยลักษณะเด่นของ RNN คือทำการจดจำข้อมูลตามลำดับเวลาที่ผ่านเข้ามา ต่อมาระบบของสถาปัตยกรรมนี้พบปัญหากับการเรียนรู้ข้อมูลที่มีลำดับของข้อมูลที่ยาว ทำให้การส่งข้อมูลเพื่อไปอัพเดทค่าน้ำหนักเพื่อปรับค่าความผิดพลาดของโมเดลเป็นไปได้ยาก หรือเกิด แกรเดียนท์ แวนิชชิ่ง (Gradient vanishihng) ซึ่งต่อมาในปี ค.ศ. 1997 สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ ก็ได้ถือกำเนิดขึ้นโดย Sepp Hochreiter ซึ่งสถาปัตยกรรม LSTM ได้มีการเพิ่มส่วนที่ใช้ลืมข้อมูลเก่าไป ทำให้ตอนที่โมเดลเรียนรู้สามารถเลือกที่จะเรียนรู้ข้อมูลเพิ่มต่อหรือเลือกที่จะหลงลือข้อมูลเก่าได้ และต่อมาสถาปัตยกรรมนี้ก็ได้ถูกต่อยอดไปเป็นสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง (Bidirectional Long Short-Term Memory หรือ BiLSTM) ซึ่งสามารถเรียนรู้ข้อมูลทั้งจากหน้าไปหลัง และจากหลังไปหน้าได้พร้อมๆกัน

2.7 แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส (Conditional Random Field หรือ CRF)

ในงานวิจัยที่เกี่ยวข้องกับ การประมวลผลภาษาทางธรรมชาติ (Natural Language Processing) ในงานที่เกี่ยวข้องกับการทำนายข้อมูลเป็นลำดับ (Sequenc Labeling) นักวิจัยยุคแรกได้ใช้โมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อใช้ในการทำนายข้อมูล แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส นั้นมีความเรียบง่ายในการใช้ทำนาย โดยที่นักวิจัยมีหน้าที่เพียงค้นหาฟีเจอร์ที่สำคัญที่ส่งผลในการทำนาย หรือให้ประสิทธิภาพของโมเดลดีขึ้น จุดเด่นของแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ คือการนำมาช่วยในการทำนายคำตอบที่เป็นลำดับ โดยแต่ละลำดับนั้นมีความเกี่ยวข้องกัน (Dependent Tagging) ในงานวิจัยทางด้านภาษาจึงเป็นที่นิยมอย่างยิ่งในการถูกนำมาใช้ในการรู้จำชื่อเฉพาะ (Named-Entity Recognition) เช่นในชื่อคนใช้ในการทำนาย B-PER หรือ ตัวขึ้นต้นชื่อคนที่เป็นชื่อเฉพาะ ดังนั้นคำตอบต่อไปจึงมีความน่าจะเป็นที่จะตอบ I-PER หรือคำที่ยังอยู่ในขอบเบตของ PER มากกว่าการทำนายว่าคำต่อไปเป็น B-PER ซ้ำตัวเดิมนั่นเอง (Jin, 2019)

สำหรับ CRF เลเยอร์นั้นทำการเรียนรู้ได้จากการทำการทำนายคำตอบที่ให้ไป โดยให้ y เป็นเซตของ $\{y_1, y_2, \ldots, y_n\}$ โดยที่การทำนายนั้นมาจากลำดับของ X โดยที่ X เป็นเซตของ $\{x_1, x_2, \ldots, x_n\}$ หลังจากนั้นจึงทำการคำนวณหาคะแนนของแต่ละลำดับได้จาก

$$s(X, y) = \sum_{i=0}^{n} (O_{i,yi} + T_{yi,yi+1}) + T_{y0,y1}$$

โดยที่ $O_{i,yi}$ คือคะแนนของ y ลำดับที่ i ของคำที่ x ลำดับที่ i ในประโยค และ T คือคะแนนแทรนซิชั่นแมททริกซ์ (Transition matrix) ที่บอกคะแนนของการตอบคำถามจาก แท็คชนิดที่ i ไปยัง เท็คชนิดที่ j ซึ่ง y_0 และ y_{n+1} คือแท็คเริ่มต้น และแท็คสุดท้ายของประโยค ดังนั้น แทรนซิชั่นแมททริกซ์ จะมีขนาดเท่ากับ k+2 โดยที่ความน่าจะเป็นของ คำตอบที่แท้จริงของลำดับ y สามารถเขียนได้เป็นสมการดังนี้

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_x} e^{s(X,\tilde{y})}}$$

โดยที่ $\tilde{y}$ แทนค่าลำดับคำตอบของประโยค และ Y_x แทนเซตของความน่าจะเป็นที่จะทำนายลำดับของคำตอบเมื่อส่งลำดับของ X เข้าไป

ในฝั่งของการถอดรหัส ได้ใช้ วิเทอบีแอลกอริทึม (Viterbi algorithm) (G. D. Forney, 1973) เพื่อใช้ในการทำนายลำดับของเส้นทางที่ให้คะแนนของคำตอบที่มีค่ามากที่สุด ซึ่งหาได้จากสมการ

$$\bar{y} = argmax_{\tilde{y} \in Y_x} s(X, \tilde{y})$$

ในการประมาณค่าของพารามิเตอร์ทำได้โดย สมมติว่ามีเซตของคำตอบ N จำนวน $\{(x_i, y_i)\}|_{i=1}^{N}$ ดังนั้นจะได้ตัวอย่างสมการในการหา ลอกไลท์ลิฮูดฟังก์ชั่น ดังนี้

$$L(\theta) = \sum_{i=0}^{N} \log\big(P(x_i, y_i)\big)$$

2.8 โมเดลทางภาษา (Language Model)

ในยุคที่การใช้งานเทคนิคโครงข่ายประสาทเป็นเรื่องที่เข้าถึงได้ง่าย การสร้างโมเดลทางภาษานั้นจึงสามารถทำได้โดยใช้เทคนิคการเรียนรู้ผ่านโครงข่ายประสาทเทียม ไปจนถึงเทคนิคที่มีความซับซ้อนมากขึ้นของการเรียนรู้เชิงลึกอย่างสถาปัตยกรรม แอทเทนชั่น (Attention model) ซึ่งโมเดลทางภาษาที่ผ่านการเรียนรู้โดยใช้เทคนิคโครงข่ายประสาทเทียมนั้นมีข้อดีตรงที่ช่วยลดจำนวนมิติทีเกิดขึ้นจากการแปลงข้อมูลทางภาษาให้อยู่ในรูปที่ใช้คำนวณได้แบบวิธีดั้งเดิม วิธีดั้งเดิมนั้นยกตัวอย่างเช่น การแปลงแต่ละคำให้อยู่ในรูปของ คอลัมน์ของคำนั้นๆว่ามีหรือไม่ ยิ่งจำนวนคำมีเยอะ ก็ยิ่งส่งผลให้จำนวนของมิติมีเยอะมากขึ้น ส่งผลเสียในการประมวลผล และการเก็บข้อมูลที่ไม่มีประสิทธิภาพ แต่กลับกันเทคนิคโครงข่ายประสาทเทียมนั้น นักวิจัยสามารถเลือก

จำนวนของมิติได้เอง และเวกเตอร์ที่นำเสนอแต่ละคำนั้น ยังมีคุณสมบัติของเวกเตอร์ที่ใช้ในการหาคำที่ใกล้เคียงกัน หรือสามารถบวก หรือลบกันเพื่อเกิดเป็นคำใหม่ได้อีกด้วย

สำหรับโมเดลทางภาษายุคเริ่มต้นนั้น ในปี ค.ศ. 1980 อาศัยโมเดลทางสถิติเข้ามาช่วยในการทำนายคำต่อไป โดยโมเดลจะทำนายว่าคำต่อไปนั้นน่าจะเป็นคำอะไรจากการดูคำ หรือประโยคตั้งต้นที่ใส่เข้าไป ให้ประโยคนั้นแทนด้วย s และจำนวนคำในประโยคเป็น N จะได้สมการดังนี้

$$P(s) = P(w_1, w_2, \dots, w_N) = P(w_1)P(w_2|w_1) \dots P(w_N|w_1, w_2, \dots, w_{N-1})$$

โดยที่ w_i คือ คำในประโยคที่ลำดับที่ i ของในประโยค s ซึ่งสามารถคำนวณคำที่น่าจะเป็นต่อไปได้โดยการหาโปรดักซ์ของลำดับ ปัญหาต่อมาของโมเดลทางภาษาแบบนี้คือการเกิดจำนวนมิติที่มีเยอะจนเกินไป เพราะหากในระบบเอกสารของเรามีคำศัพท์มากมายถึง 10,000 คำ จะต้องคำนวณหาค่าถึง $10{,}000^n - 1$ พารามิเตอร์

จากปัญหาดังกล่าวได้มีการแก้ไขปัญหาดังกล่าวโดยใช้โมเดลโครงข่ายประสาทเทียม (Neural network) ที่เปลี่ยนโครงสร้างของขอบเขตการคำนวนของข้อมูลจาก ดิสครีตสเปซ (Discrtete space) ไปเป็น คอนตินิวอัสสเปซ (Continuous space) ด้วยความสามารถของสถาปัตยกรรมโครงข่ายประสาทเทียมทำให้โมเดลสามารถเรียนรู้ความน่าจะเป็นของคำต่อไปที่จะเกิดได้ในรูปแบบปัญหาของ คอนตินิวอัสสเปซ และในปี ค.ศ. 2003 Bengio และคณะได้ทำการสร้างโมเดลทางภาษาเพื่อแก้ปัญหาการจำนวนมิติที่เยอะเกินไปจากการใช้โมเดล ฟีดฟอเวิร์ดนิวรอลเนทเวิค (Feedforward neural network) ต่อมาเรียกวิธีนี้ว่า การทำเอมเบดดิ้ง (Embedding) และในปี ค.ศ. 2010 (Mikolov, 2010) และคณะได้เสนอ โมเดลทางภาษาที่สร้างขึ้นจากสถาปัตยกรรมที่มีหน่วยของความจำในการจำข้อมูลที่เป็นลำดับ หรือโมเดลโครงข่ายประสาทเทียมชนิดรีเคอเรนท์ มาใช้ในการสร้าง เอมเบดดิ้ง ตั้งแต่นั้นมาความนิยมในการสร้างโมเดลทางภาษาจากเทคนิคโครงข่ายประสาทเทียมก็เกิดขึ้นอย่างแพร่หลาย ในปี ค.ศ. 2013 (Mikolov, 2013) และคณะได้ 2 สถาปัตยกรรมที่ใช้ในการทำโมเดลทางภาษาที่สามารถเรียนรู้ข้อมูลบน คอนตินิวอัสสเปซ ผ่านชุดข้อมูลขนาดใหญ่ ที่มีคำอยู่ถึง 1.6 พันล้านคำและยังมีประสิทธิภาพที่ดีที่สุดในยุคนั้น ประกอบกับโมเดลทางภาษาใช้เวลาในการเรียนรู้น้อย ทำให้วิธีของ Mikolov เป็นที่นิยมอย่างมาก โดย 2 วิธีดังกล่างได้แก่ คอนตินิวอัสแบกออฟเวิด (Continuous Bag-of-Words Model, CBOW) และ สคริปแกรม (Continuous Skip-gram Model)

2.8.1 Continuous Bag-of-Words Model

Mikolov ได้เสนอสถาปัตยกรรมของโมเดลทางภาษาที่คล้ายกับ ฟีตฟอเวิร์ดนิวรอลเนทเวิค โดยที่นำชั้น นอนลิเนียร์ฮิดเดนท์ เลเยอร์ (non-linear hidden layer) ออก แล้วทำการเปลี่ยนเป็นชั้นของ โพรเทคชั่น เลเยอร์ (projection layer) เพื่อใช้ในการแชร์ข้อมูลกันระหว่างทุกๆคำ และที่จุดที่ทำการโพรเทคนั่นเองทำให้เกิด เวกเตอร์ที่ได้เป็นค่าเฉลี่ย โดยที่ Mikolov เรียกสถาปัตยกรรมรูปแบบนี้ว่า แบ็คออฟเวิร์ด โมเดล (bag-of-words model) โดยโมเดลทำการเรียนรู้คำที่อยู่รอบข้าง ยกตัวอย่างเช่น 4 คำก่อนหน้าเป็นคำอะไรบ้าง และ 4 คำถัดไปเป็นคำอะไรบ้าง เพื่อใช้ทำนายตัวที่อยู่ตรงกลางว่าคือคำอะไรนั่นเอง (Mikolov, 2013)

2.8.2 Continuous Skip-gram Model

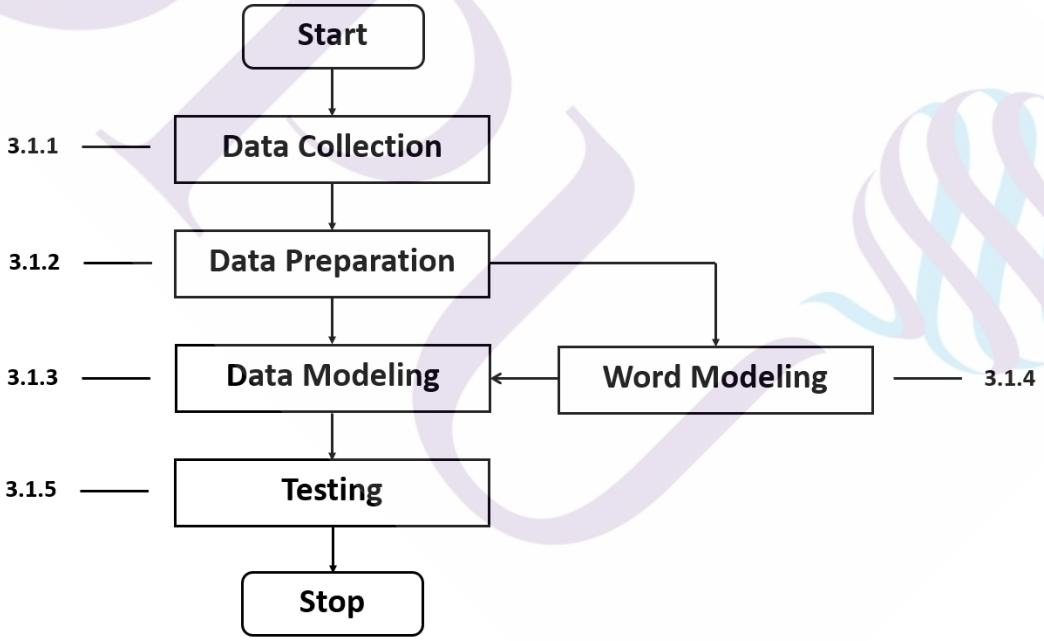
สำหรับสถาปัตยกรรมที่ 2 ของ Mikolov นั้น แทนที่จะทำการทำนายคำที่อยู่ตรงกลาง โมเดลจะทำการทำนายคำที่อยู่รอบๆข้างแทน เช่น 4 คำถัดไปเป็นคำอะไรบ้าง และ 4 คำ ก่อนหน้าเป็นคำอะไรบ้าง จากสถาปัตยกรรมดังกล่าว Mikolov กล่าวว่ายิ่งเพิ่มขอบเขตการทำนายของคำ ยิ่งทำให้เวปเตอร์ของคำที่ได้มีประสิทธิภาพมากขึ้น (Mikolov, 2013)

บทที่ 3
ระเบียบวิจัย

ในบทนี้จะกล่าวถึงงานวิจัยเชิงประจักษ์ (Empirical Research) โดยผู้วิจัยทำการศึกษาถึงการสร้าง และพัฒนาโมเดลเพื่อใช้ในการตัดประโยคภาษาไทย ในบทนี้จะกล่าวถึงการเตรียมชุดข้อมูลเพื่อนำไปใช้ในการเทรนโมเดล กรอบแนวคิดที่ใช้ในการสร้างโมเดลสำหรับการตัดประโยคภาษาไทย การวัดประสิทธิภาพของโมเดล และเครื่องมือที่ใช้ในงานวิจัย

3.1 แนวทางการวิจัย

ทางผู้วิจัยได้แสดงลำดับขั้นตอนของกรอบในงานวิจัยผ่านภาพที่ 3.1 โดยผู้วิจัยจะทำการอธิบายในหัวข้อต่อมา



ภาพที่ 3.1 ระเบียบกรอบแนวคิดที่ใช้ในการสร้างตัวตัดประโยคภาษาไทย

3.1.1 วิธีการเก็บรวบรวมข้อมูล (Data Collection)

ในหัวข้อนี้ทางผู้วิจัยได้อธิบายถึงวิธีการเก็บรวบรวมชุดข้อมูลเพื่อใช้ในการสร้างโมเดลทางภาษา 2 รูปแบบด้วยกันได้แก่ CBOW และ Skip-Gram โดยทางผู้วิจัยได้ใช้ชุดข้อมูลจำนวน 3 ชุดข้อมูล โดยข้อมูลชุดที่หนึ่งคือ ORCHID ซึ่งเป็นชุดข้อมูลที่ปล่อยมาสู่สาธารณชนในปี ค.ศ. 1998, ชุดข้อมูลที่สองคือ scb-mt-en-th-2020 ซึ่งเป็นชุดข้อมูลที่เกี่ยวข้องกับคู่ประโยคแปลภาษาระหว่าง ภาษาไทย-อังกฤษ โดยปล่อยมาเมื่อเดือนสิงหาคม 2020 และสุดท้ายคือชุดข้อมูล LST-20 ที่ปล่อยมาล่าสุดโดยงานวิจัยของเนคเทค และสวทช. สำหรับชุด ข้อมูล scb-mt-en-th-2020 ประกอบด้วยชุดข้อมูลย่อยๆจาก 12 แหล่งข้อมูลรวมกันแสดงผ่านตาราง 3.1

ตารางที่ 3.1 แสดงถึงจำนวนประโยคที่มีของแต่ละชุดข้อมูล

ชุดข้อมูล	จำนวนประโยค
Orchid	30,000
scb-mt-en-th-2020	
generated_review_crowd	24,587
mozilla_common_voice	33,797
task_master_1	222,733
paracrawl	60,039
apdf	13,503
generated_review_translator	133,330
nus_sms	43,750
thai_website	120,280
generate_review_yn	280,208
wikipedia	33,756
msr	10,371
assorted_government	25,398
LST20	527,366

กล่าวถึงชุดข้อมูล ORCHID ซึ่งเป็นชุดข้อมูลที่ประกอบไปด้วยข้อมูลในระดับคำ และระดับ POS ซึ่งข้อมูล POS ของ ORCHID นั้นถูกสร้างขึ้นมาจำนวนทั้งหมด 47 ประเภท ซึ่งแสดงดังตารางที่ 3.2 สำหรับชุดข้อมูลของ ORCHID นั้นไม่มีการจำแนกประเภทของคำเฉพาะมาให้ในชุดข้อมูล

ตารางที่ 3.2 แสดงถึงประเภทของ POS จากชุดข้อมูล ORHCID

ตัวย่อ	POS tags	ตัวอย่าง
NPRP	Proper noun	วินโดวส์ 95, โคโรน่า, โค้ก
NCNM	Cardinal number	หนึ่ง, สอง, สาม, 1, 2, 10
NONM	Ordinal number	ที่หนึ่ง, ที่สอง, ที่สาม, ที่1, ที่2
NLBL	Label noun	1, 2, 3, 4, ก, ข, a, b
NCMN	Common noun	หนังสือ, อาหาร, อาคาร, คน
NTTL	Title noun	ครู, พลเอก
PPRS	Personal pronoun	คุณ, เขา, ฉัน
PDMN	Demonstrative pronoun	นี่, นั้น, ที่นั่น, ที่นี่
PNTR	Interrogative pronoun	ใคร, อะไร, อย่างไร
PREL	Relative pronoun	ที่, ซึ่ง, อัน, ผู้
VACT	Active verb	ทำงาน, ร้องเพลง, กิน
VSTA	Stative verb	เห็น, รู้, คือ
VATT	Attributive verb	อ้วน, ดี, สวย
XVBM	Pre-verb auxiliary, before negator “ไม่”	เกิด, เกือบ, กำลัง
XVAM	Pre-verb auxiliary, after negator “ไม่”	ค่อย, น่า, ได้
XVMM	Pre-verb, before or after negator “ไม่”	ควร, เคย, ต้อง
XVBB	Pre-verb auxiliary, in imperative mood	กรุณา, จง, เชิญ, อย่า, ห้าม
XVAE	Post-verb auxiliary Å	ไป, มา, ขึ้น
DDAN	Definite determiner, after noun without classifier in between	ยี่, นั่น, โน่น, ทั้งหมด

ตารางที่ 3.2 (ต่อ)

DDAC	Definite determiner, allowing classifier in between	นี้, นั้น, โน้น, นู้น
DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression	ทั้ง, อีก, เพียง
DDAQ	Definite determiner, following quantitative expression	พอดี, ถ้วน
DIAC	Indefinite determiner, following noun; allowing classifier in between	ไหน, อื่น, ต่างๆ
DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression	บาง, ประมาณ, เกือบ
DIAQ	Indefinite determiner, following quantitative expression	กว่า, เศษ
DCNM	Determiner, cardinal number expression	หนึ่งคน, เสือ, 2 ตัว
DONM	Determiner, ordinal number expression	ที่หนึ่ง, ที่สอง, ที่สุดท้ัย
ADVN	Adverb with normal form	เก่ง, เร็ว, ช้า, สม่ำเสมอ
ADVI	Adverb with iterative form	เร็วๆ, เสทอๆ, ช้าๆ
ADVP	Adverb with prefixed form	โดยเร็ว
ADVS	Sentential adverb	โดยปกติ, ธรรมดา
CNIT	Unit classifier	ตัว, คน, เล่ม
CLTV	Collective classifier	คู่, กลุ่ม, ฝูง, เชิง, ทาง, ด้าน, แบบ, รุ่น
CMTR	Measurement classifier	กิโลกรัม, แก้ว, ชั่วโมง

ตารางที่ 3.2 (ต่อ)

CFQC	Frequency classifier	ครั้ง, เที่ยว
CVBL	Verbal classifier	ม้วน, มัด
JCRG	Coordinating conjunction	และ, หรือ, แต่
JCMP	Comparative conjunction	กว่า, เหมือนกับ, เท่ากับ
JSBR	Subordinating conjunction	เพราะว่า, เนื่องจาก ที่
RPRE	Preposition	จาก, ละ, ของ, ใต้, บน
INT	Interjection	โอ้บ, โอ้, เออ, เฮ๋, อ๋อ
FIXN	Nominal prefix	การทำงาน, ความสนุนสนาน
FIXV	Adverbial prefix	อย่างเร็ว
EAFF	Ending for affirmative sentence	จ๊ะ, จ้ะ, ค่ะ, ครับ, นะ, น่า, เถอะ
EITT	Ending for interrogative sentence	หรือ, เหรอ, ไหม, มั้ย
NEG	Negator	ไม่, มิได้, ไม่ได้, มิ
PUNC	Punctuation	(,), “, „, ;

สำหรับชุดข้อมูล LST20 นั้นมีแบ่งข้อมูลให้ในระดับคำ และระดับประโยค รวมถึงยังมี POS และ NEs ชนิดใหม่ที่สร้างขึ้นมาโดยเฉพาะในชุดข้อมูล LST20 ซึ่งประกอบไปด้วย POS จำนวน 16 ประเภท และ NEs จำนวน ซึ่งแสดงไว้ในตาราง 3.3 และ 3.4 ตามลำดับ

ตารางที่ 3.3 แสดงถึงประเภทของ POS จากชุดข้อมูล LST20

ตัวย่อ	**POS tags**
AJ	Adjective
AV	Adverb
AX	Auxiliary
CC	Connector
CL	Classifier
FX	Prefix

ตารางที่ 3.3 (ต่อ)

IJ	Interjection
NG	Negator
NN	Noun
NU	Number
PA	Particle
PR	Pronoun
PS	Preposition
PU	Punctuation
VV	Verb
XX	Others

ตารางที่ 3.4 แสดงถึงประเภทของ NEs จากชุดข้อมูล LST20

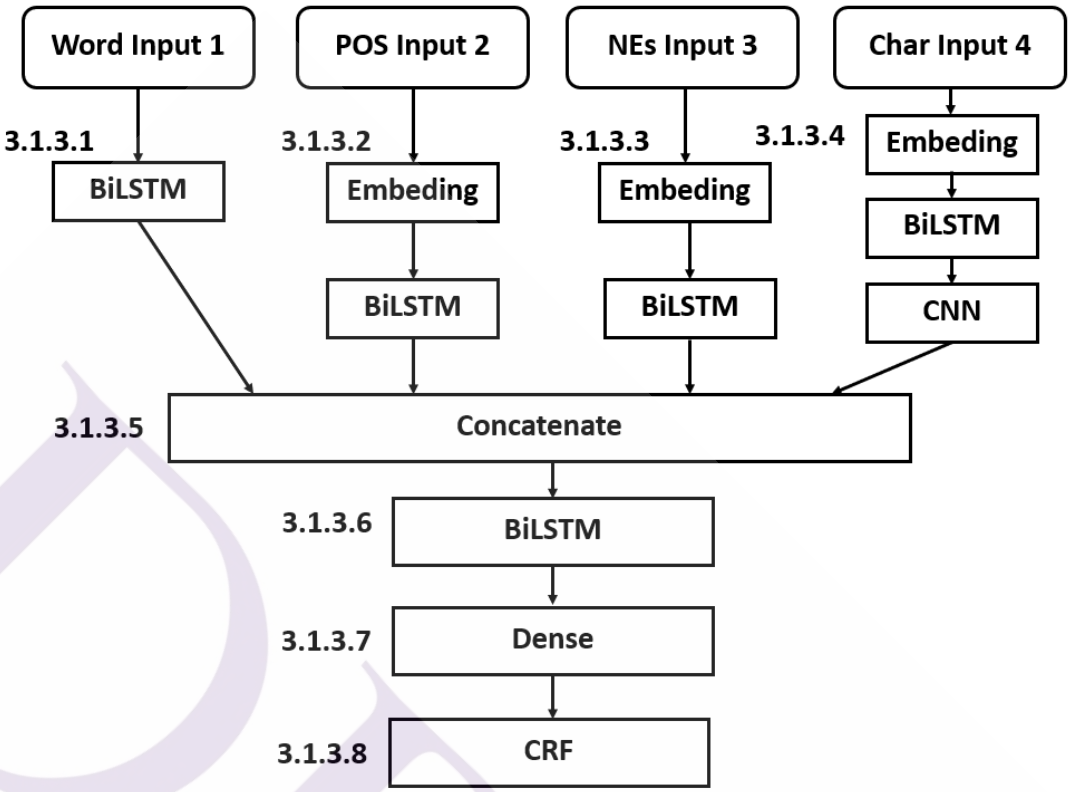
ตัวย่อ	ชื่อของตัวย่อ
TTL	Title
DES	Designation
PER	Person
ORG	Organization
LOC	Location
DTM	Date and time
BRN	Brand
MEA	Measurement
NUM	Number
TRM	Terminology

3.1.2 การเตรียมข้อมูล (Data Preparation)

ในส่วนนี้ผู้วิจัยได้อธิบายถึงวิธีการเตรียมข้อมูลเพื่อใช้ในการสร้างโมเดลตัดประโยคภาษาไทย โดยที่โมเดลดังกล่าวที่สร้างขึ้น นั้นจำเป็นต้องใช้ชุดข้อมูลที่อ้างอิงคล้ายรูปแบบของพารากราฟ (Paragraph) เพื่ออ้างอิงรูปแบบของข้อมูลที่นำไปใช้ในการให้โมเดลเรียนรู้ว่าควรตัดประโยคที่จุดหรือตำแหน่งไหน หรือคำไหนคือคำเริ่มต้นประโยค คำไหนคือคำที่บอกจุดจบประโยค ในงานวิจัยชิ้นนี้นักวิจัยได้ทำการรวมประโยค 4 ประโยคเข้าด้วยกันเพื่อสร้างเป็น 1 พารากราฟ ทั้งนี้เพราะมีข้อจำกัดของ หน่วยประมวลผลด้านกราฟฟิก 3 มิติ (Graphics Processing unit, GPU) ที่ใช้ประมวลผล ที่สามารถรองรับได้เพียง 4 พารากราฟสำหรับเทรนโมเดล หรือ 200 คำต่อ 1 ตัวอย่างพารากราฟ

3.1.3 การสร้างแบบจำลอง (Modeling Techniques)

ในหัวข้อนี้ผู้วิจัยได้อธิบายถึงโมเดลที่ใช้ในการตัดประโยคภาษาไทย โดยปัญหาของงานวิจัยนี้คือการค้นหาว่าคำไหนคือตัวเริ่มต้นประโยค คำไหนคือ คำที่อยู่ระหว่างประโยค และคำไหนที่บอกถึงจุดจบของประโยค โดยโมเดลดังกล่าวถูกสร้างขึ้นมาโดยอาศัยเทคนิคการเรียนรู้เชิงลึก (Deep Learning) ซึ่งโมเดลดังกล่าวถูกออกแบบโดยมีพื้นฐานให้สามารถรองรับทางเข้าของฟีเจอร์ได้หลากหลายชนิด เช่นสามารถรองรับฟีเจอร์ในระดับคำ ซึ่งสามารถใส่ฟีเจอร์เพิ่ม เวิร์ดเอมเบดดิ่ง (Word Embedding) ไวยกรณ์ทางภาษา (Part of Speech, POS) และ ชื่อเฉพาะ (Named Entity, NEs) นอกจากนี้ยังสามารถรองรับฟีเจอร์ในระดับตัวอักษร ทั้งนี้กรอบแนวคิดที่ได้ทำการทดลองผู้วิจัยได้ออกแบบงานวิจัยโดยแบ่งฟีเจอร์ที่นำมาใช้งานในระดำคำในส่วนของ เวิร์ดเอมเบดดิ่ง (Word Embedding) ออกเป็น 2 ส่วนด้วยกัน ส่วนที่หนึ่งเป็นการสร้างโมเดลทางภาษาโดยใช้เทคนิค คอนตินิวอัสแบคออฟเวิร์ด (Continuous Bag of Words, CBOW) และ เทคนิค สคริปแกรม (Skip-grams) โดยทางผู้วิจัยจะทำการอธิบายเพิ่มเติมในส่วนถัดไป สำหรับ ฟีเจอร์ ไวยกรณ์ทางภาษา และ ชื่อเฉพาะ ผ่านกรอบแนวคิดดั่งในภาพที่ 3.2 ด้วยกรอบแนวคิดดังกล่าวทำให้กรอบแนวคิดที่ใช้ในการสร้างโมเดลตัดประโยคภาษาไทยนั้น สามารถหาจุดปรับปรุงให้มีประสิทธิภาพดีขึ้นได้



ภาพที่ 3.2 กรอบแนวคิดสำหรับโมเดล บอยด์คัท เพื่อใช้ตัดประโยคภาษาไทย

3.1.3.1 Word Input layer

เป็นส่วนของชั้นที่นำเวิร์ดเอมเบดดิ้งมาทำการเรียนรู้เพิ่มเติมในชั้นของ Bi-LSTM เพื่อทำการเรียนรู้รูปแบบของข้อมูลที่มีการเรียงเป็นลำดับก่อนหลัง แล้วส่งต่อเวกเตอร์ส่วนที่เรียนรู้ได้ไปยังชั้น 3.1.3.5

3.1.3.2 POS Input layer

เป็นส่วนของชั้นที่นำไวยกรณ์ทางภาษาในระดับคำมาผ่านการสร้างเวกเตอร์เอมเบดดิ้งมาทำการเรียนรู้เพิ่มเติมในชั้นของ Bi-LSTM เพื่อทำการเรียนรู้รูปแบบของข้อมูลที่มีการเรียงเป็นลำดับก่อนหลัง แล้วส่งต่อเวกเตอร์ส่วนที่เรียนรู้ได้ไปยังชั้น 3.1.3.5

3.1.3.3 NEs Input layer

เป็นส่วนของชั้นที่นำคำเฉพาะในระดับคำมาผ่านการสร้างเวกเตอร์เอมเบดดิ้ง มาทำการเรียนรู้เพิ่มเติมในชั้นของ Bi-LSTM เพื่อทำการเรียนรู้รูปแบบของข้อมูลที่มีการเรียงเป็นลำดับก่อนหลัง แล้วส่งต่อเวกเตอร์ส่วนที่เรียนรู้ได้ไปยังชั้น 3.1.3.5

3.1.3.4 Character Input layer

ชั้นนี้ทำการใช้ประโยชน์ของ สถาปัตยกรรมโครงข่ายประสาทเทียมแบคอนโวลูชั่นมาทำการช่วยสกัดข้อมูลที่สำคัญในระดับตัวอักษรของแต่ละคำในประโยค แล้วส่งต่อเวกเตอร์ส่วนที่เรียนรู้ได้มาทำการเรียนรู้เพิ่มเติมในชั้นของ Bi-LSTM เพื่อทำการเรียนรู้รูปแบบของข้อมูลที่มีการเรียงเป็นลำดับก่อนหลัง แล้วส่งต่อเวกเตอร์ส่วนที่เรียนรู้ได้ไปยังชั้น 3.1.3.5

3.1.3.5 Concatenate layer

ชั้นนี้ทำหน้าที่เพื่อนำผลลัพธ์ที่เป็นเวกเตอร์ของแต่ละสายของข้อมูลในชั้นก่อนหน้ามาต่อกันให้เกิดเวกเตอร์ใหม่ ไม่ได้ทำการเรียนรู้อะไรเป็นพิเศษ

3.1.3.6 BiLSTM layer

ชั้นที่นำเวคเตอร์ในชั้นก่อนหน้ามาทำการเรียนรู้คอนเท็กอีกรอบหนึ่ง หลังจากนั้นทำการส่งต่อผลลัพธ์ของเวกเตอร์ที่ได้ในแต่ละชั้นไปยังชั้นถัดไป

3.1.3.7 Dense Layer

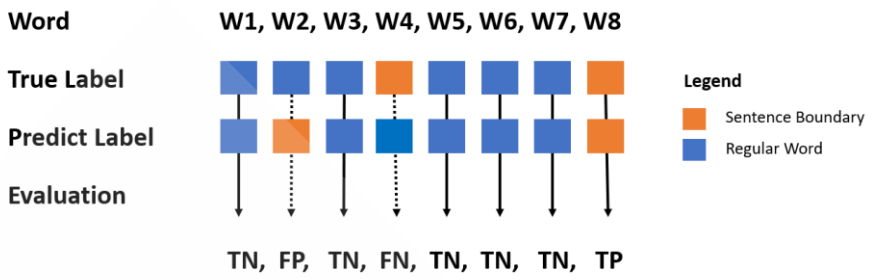
ชั้นนี้เป็นการนำผลลัพธ์ในแต่ละลำดับที่ได้นำมาส่งต่อไปยังชั้นถัดไปให้กับแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อทำการถอดรหัสหรือทำนายว่าแต่ละลำดับเป็นคำที่เป็นตัวตัดประโยคหรือไม่

3.1.3.8 CRF layer

ชั้นนี้เป็นเสมือนชั้นที่ใช้ในการถอดรหัสหรือทำนายว่าแต่ละลำดับเป็นคำที่เป็นตัวตัดประโยคหรือไม่ โดยใช้แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อมาช่วยในการตอบ

3.1.5 วิธีการทดสอบประสิทธิภาพโมเดล

ผู้วิจัยได้ใช้วิธีวัดประสิทธิภาพของโมเดลโดยใช้ค่า เอฟวัน แมคโคร (F1-Macro) เพราะเป็นการวัดประสิทธิภาพแต่ละคลาสของคำตอบโดยวัดคำตอบทีละคำ และ เอฟวันแมคโคร ยังเป็นเมทริกซ์ที่เหมาะสมในการวัดประสิทธิภาพของโมเดลที่จำนวนคลาสแต่ละคลาสนั้นมีจำนวนของข้อมูลที่มีอัตราส่วนไม่เท่ากัน โดยแสดงการวัดผลในภาพที่ 3.3



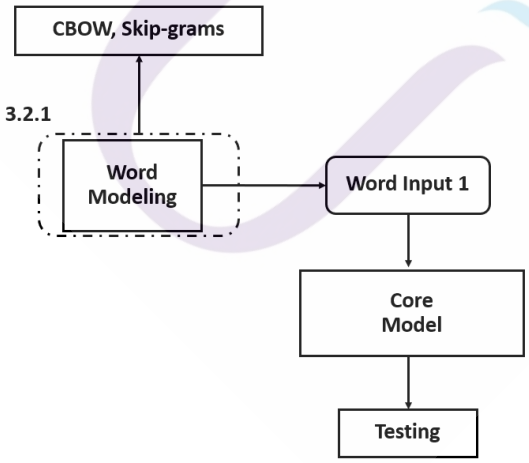
ภาพที่ 3.3 แสดงการคำนวณการวัดประสิทธิภาพโมเดลตัดประโยคภาษาไทย

3.2 วิธีออกแบบการทดลอง (Experimental Design)

ในส่วนนี้ผู้วิจัยได้ทำการอธิบายรายละเอียดของกรอบแนวคิดที่ใช้ในการออกแบบงานวิจัยในแต่ละส่วน เพื่อค้นหาองค์ประกอบที่ให้โมเดลตัดประโยคภาษาไทยที่มีประสิทธิภาพดีที่สุด งานวิจัยได้ถูกแบ่งออกเป็น สามส่วนใหญ่ๆด้วยกัน

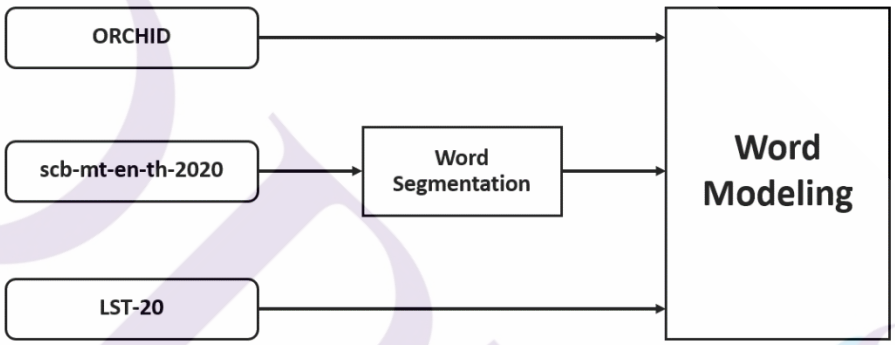
3.2.1 การสร้างโมเดลทางภาษา (Word Modeling)

ในส่วนของฟีเจอร์ระดับคำที่ใช้ในการสร้างโมเดลทางภาษา นักวิจัยได้ทำการทดลองสร้างโมเดลทางภาษาสองเทคนิค เพื่อใช้ในการเปรียบเทียบประสิทธิภาพของสองเทคนิคที่ใช้สร้างโมเดลทางภาษา เพื่อเปรียบเทียบว่าเทคนิคชนิดไหนส่งผลให้ประสิทธิภาพของโมเดลตัดประโยคภาษาไทยส่งผลดีกว่ากัน โดยแสดงดังภาพที่ 3.4



ภาพที่ 3.4 ส่วนของการสร้างโมเดลทางภาษาเพื่อใช้เป็นฟีเจอร์ใน บอยด์คัท โมเดล

สำหรับขั้นตอนการสร้างโมเดลทางภาษา (Word Language Modeling) ทางผู้วิจัยได้ทำการรวบรวมประโยคจากทั้งสามแหล่งข้อมูล ซึ่งในการสร้างโมเดลทางภาษานั้นจำเป็นต้องมีการจัดรูปแบบของข้อมูลเพื่อใช้ในการเรียนรู้ ให้อยู่ในรูปแบบของ ["นั่ง", "ริม", "สระ"] ทั้งนี้จากชุดข้อมูลทั้งสามแหล่ง มีเพียงแหล่งข้อมูลของ scb-mt-en-th-2020 ที่รูปแบบของชุดข้อมูลเป็นรูปแบบประโยคโดยตรง ยกตัวอย่างเช่น "เช่นเดียวกับเมื่อก่อน สภาเป็นผู้ที่จะตัดสินใจทิศทางและกระบวนการ" ไม่ได้มีการตัดคำมาให้ใช้งานได้โดยตรง ดังนั้นทางผู้วิจัยจึงต้องทำการตัดคำจากชุดข้อมูลของ scb-mt-en-th-2020 เสียก่อน โดยผู้วิจัยได้ทำการเลือกวิธีการตัดคำโดยใช้โมเดล AttaCut (P. Chormai, 2019) และแสดงเส้นทางของการสร้างโมเดลทางภาษาดังภาพที่ 3.5

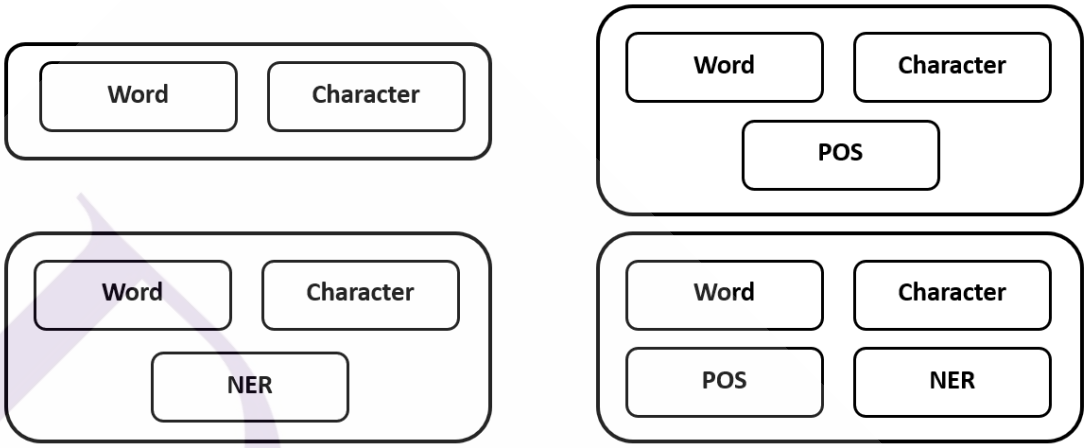


ภาพที่ 3.5 แสดงถึงขั้นตอนการนำชุดข้อมูลมาใช้ทำโมเดลทางภาษา

3.2.2 ฟีเจอร์ที่ใช้ในการเรียนรู้ (Input Features)

ในส่วนของฟีเจอร์ที่ใช้ในการเรียนรู้นั้น ผู้วิจัยแบ่งการทดลองออกเป็น 4 กลุ่มด้วยกัน โดยกลุ่มที่หนึ่งคือใช้ฟีเจอร์เพียงคำ และตัวอักษรเท่านั้น กลุ่มที่สองทำการเพิ่มฟีเจอร์ POS เข้าไป

ในกลุ่มที่หนึ่ง กลุ่มที่สามทำการเพิ่มฟีเจอร์ NEs เข้าไปในกลุ่มที่หนึ่ง และกลุ่มสุดท้ายทำการเพิ่มฟรเจอร์ทั้ง POS และ NEs เข้าไปในกลุ่มที่หนึ่ง โดยแสดงในภาพที่ 3.6



ภาพที่ 3.6 เซตของฟีเจอร์ที่ใช้ในการเทรนโมเดล บอยด์คัท

3.2.3 ขั้นตอนการแปลงฟีเจอร์เพื่อใช้งานโมเดล บอยด์คัท จากฟีเจอร์กลุ่มต่างๆ

ในหัวข้อนี้ผู้วิจัยได้ทำการอธิบายถึงขั้นตอนการใช้งานโมเดลการตัดประโยคภาษาไทย บอยด์คัท ทั้งนี้จากการออกแบบการทดลอง ทางผู้วิจัยได้ทำการออกแบบกลุ่มของฟีเจอร์ที่ใช้ในการเรียนรู้สำหรับโมเดล 4 กลุ่มฟีเจอร์ โดยกลุ่มที่ใช้ฟีเจอร์เฉพาะ คำ และ ตัวอักษร เป็นกลุ่มที่ใช้งานง่ายที่สุดสำหรับการนำไปใช้งานจริง ซึ่งทางผู้วิจัยได้ออกแบบให้ผู้ใช้งานสามารถเพิ่มฟีเจอร์ที่เกี่ยวข้องเพื่อทำการทดสอบตัดประโยคภาษาไทยได้ ซึ่งแสดงดังภาพที่ 3.7

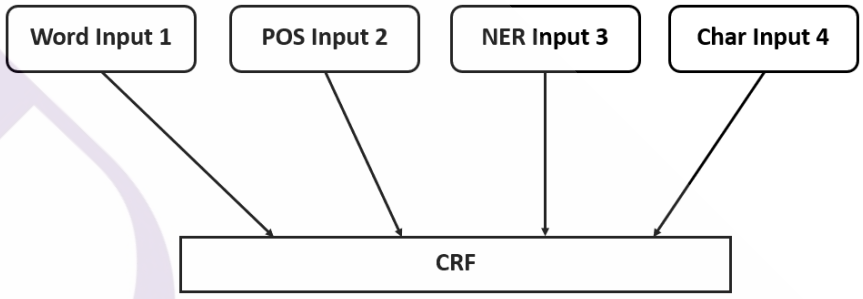
ในส่วนของฟีเจอร์คำ และ ตัวอักษร ผู้ใช้งานจำเป็นต้องทำการแปลงพารากราฟของเอกสารที่ต้องการตัดประโยค เป็นประโยคที่มีความยาวไม่เกิน 200 ตัวอักษร จากนั้นจึงทำการตัดคำโดยใช้โมเดล AttaCut (P. Chormai, 2019) หลังจากนั้นจึงนำกลุ่มคำไปใช้ในการสร้างฟีเจอร์ในระดับตัวอักษรต่ออีกทีหนึ่ง ซึ่งแสดงในภาพที่ 3.8

หลังจากที่ได้คำที่ถูกตัดแล้ว จึงนำไปผ่านโมเดลเพื่อใช้ในการทำนาย POS และ NEs ซึ่งโมเดลที่ใช้ในการทำนายนั้น POS และ NEs ต้องถูกสร้างมาจาก POS และ NEs ที่มาจากชุดข้อมูล LST20 โดยเฉพาะ ซึ่งแสดงไว้ในภาพที่ 3.9

3.2.4 ส่วนของรูปแบบสถาปัตยกรรมที่ใช้ในการตัดประโยค

ในส่วนที่สามคือการทดลองรูปแบบของสถาปัตยกรรมต่างๆที่ใช้ในการสร้างโมเดลตัดประโยคภาษาไทย โมเดลตัดประโยคภาษาไทยประกอบไปด้วยส่วนของ คอนเทกซ์ เอนโคดดิ้ง

(Context Encoding) หรือการออกแบบสถาปัตยกรรมเพื่อนำมาเรียนรู้ข้อมูลที่เป็นลำดับ ส่วนต่อมาคือส่วนของ แท็กดีโคดดิ้ง (Tag Decoder) ที่ใช้ในการตอบคำถามว่าคำไหนเป็นจุดตัดประโยค หรือจุดไหนไม่ใช่คำที่เป็นจุดตัดประโยค โดยการทดลองดังกล่าวผู้วิจัยได้ออกแบบการทดลองเปรียบเทียบกับสถาปัตยกรรมรูปแบบดั้งเดิม นั่นคือการใช้แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส โดยแสดงในภาพที่ 3.7

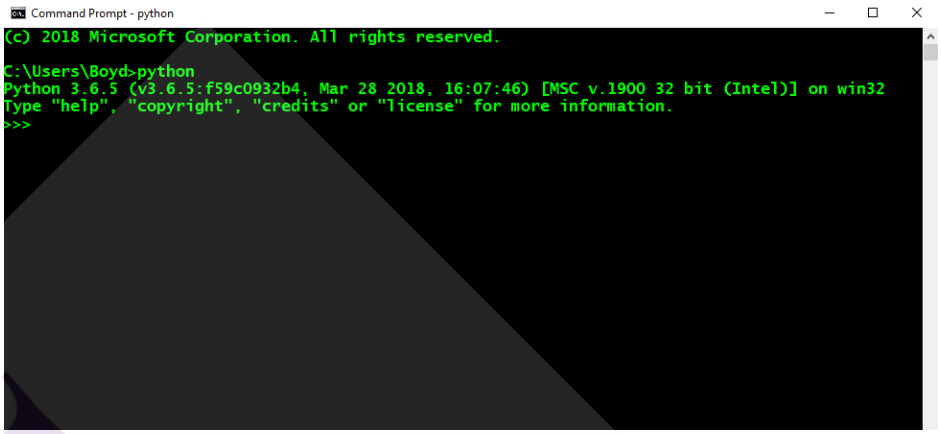


ภาพที่ 3.7 โมเดลที่สร้างจาก แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพื่อใช้ในการเปรียบเทียบการทดลองที่ใช้เทคนิคการเรียนรู้เชิงลึกเข้ามาช่วยในการเพิ่มประสิทธิภาพโมเดล

3.3 เครื่องมือที่ใช้ในงานวิจัย (Research tools)

3.3.1 ภาษาไพทอน (Python language)

ภาษาไพทอนเป็นหนึ่งในภาษาที่ได้รับความนิยมอย่างแพร่หลายในกลุ่มนักวิเคราะห์ข้อมูล เนื่องจากการติดตั้งที่ง่าย รูปแบบไวยกรณ์ของภาษาที่เข้าใจได้ไม่ยาก มีการเรียนรู้ของผู้เรียนที่ไว ด้วยเหตุนี้จึงส่งผลให้เกิดชุมชนที่ใหญ่ขึ้นเรื่อยๆ ส่งผลให้เกิด ไลบรารี ที่เป็นประโยชน์ในงานหลากหลายสาขา ภาษาไพทอนยังมีไลบรารีที่ใช้เพื่อให้การประมวลผลไวใกล้เคียงกับภาษาระดับล่างเช่น ภาษาซี ภาษาฟอแทรน ได้อีกด้วย

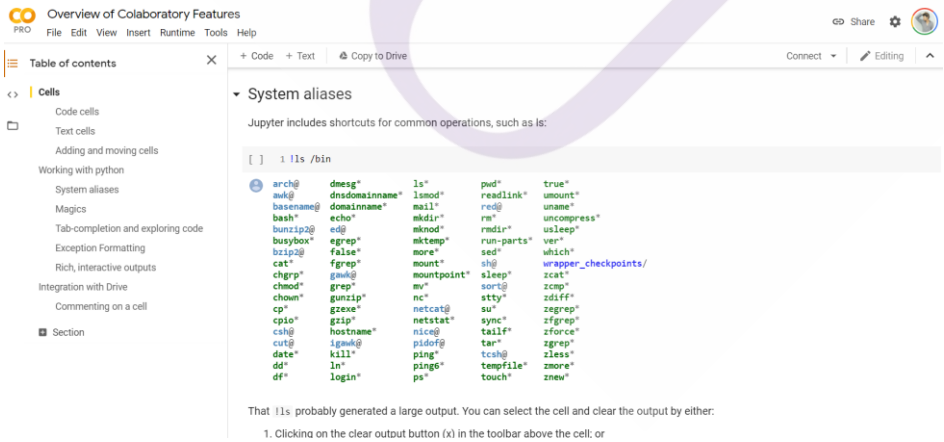


ภาพที่ 3.8 ตัวอย่างการใช้งาน Python ผ่าน Command Prompt

3.3.2 กูเกิลโคแลโบราโทรี (Google Colaboratory)

กูเกิลโคแลโบราโทรี คือหนึ่งในเครื่องมือที่เป็นที่นิยมมากที่สุดของนักวิเคราะห์ข้อมูล หรือนักวิจัยที่ใช้ในการออกแบบทำการทดลองด้านการเรียนรู้ของเครื่อง หรือทำงานวิจัยที่เกี่ยวข้องกับการใช้เทคนิคการเรียนรู้เชิงลึก ทั้งนี้เพราะเป็นเครื่องมือที่ทางผู้พัฒนาให้ใช้งานได้ฟรี และยังมีหน่วยประมวลผลด้านกราฟฟิก 3 มิติ ให้ใช้ฟรีอีกด้วย

กูเกิลโคแลโบราโทรี ยังมี โปรเวอร์ชั่น ที่เสียเงินเดือนละประมาณ 10 ดอลลาห์ แต่สามารถใช้งาน กูเกิลโคแลโบราโทรี ได้อย่างมีประสิทธิภาพมากขึ้น จากเดิมปกติสามารถใช้ได้ฟรีต่อเนื่อง 12 ชั่วโมง กลายเป็น 24 ชั่วโมง และช่วยประหยัดเวลาการประมวลผลจากการที่สามารถเข้าถึง หน่วยประมวลผลด้านกราฟฟิก 3 มิติ ขั้นสูงได้จากเดิมสามารถใช้ K80s กลายเป็น P100 ในโปรเวอร์ชั่น



ภาพที่ 3.9 หน้าใช้งานของ กูเกิลโคแลโบราโทรี

บทที่ 4
ผลการศึกษา

ผลจากการวิจัยการสร้างโมเดลเพื่อใช้ในการตัดประโยคภาษาไทย โดยใช้พื้นฐานโมเดลจากโมเดลการเรียนรู้เชิงลึก (Deep Learning) ผู้วิจัยได้ทำการออกแบบการทดลองโดยใช้ข้อมูลของ LST20 จัดทำขึ้นโดย เนคเทค และสวทช. โดยที่ผู้วิจัยได้ทำการเลือกออกแบบการทดลองโดยแยกกลุ่มของฟีเจอร์เป็นรูปแบบต่างๆ เพื่อใช้ในการทดลองการสร้างโมเดลการตัดประโยคภาษาไทย ทั้งนี้ฟีเจอร์รูปแบบต่างๆประกอบไปด้วย การใช้คำ และตัวอักษรเป็นตัวแทนของฟีเจอร์ การใช้คำ ตัวอักษร และชื่อเฉพาะ (Named-Entity Recognition) เป็นตัวแทนของฟีเจอร์ การใช้คำ ตัวอักษร และหน้าที่ของคำ (Part of Speech) เป็นตัวแทนของฟีเจอร์ และสุดท้ายคือการทดลองใช้คำ ตัวอักษร หน้าที่ของคำ และชื่อเฉพาะ เป็นตัวแทนของฟีเจอร์ โดยประยุกต์ใช้กับสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง (Bidirectional Long Short-Term Memory หรือ BiLSTM) เทียบกับแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ซึ่งเป็นตัวพื้นฐานไว้เทียบผลการทดลอง

จากการวิจัยเพื่อทำโมเดลตัดประโยคภาษาไทย สามารถแบ่งวิเคราะห์ออกมาได้ 4 ส่วนด้วยกันดังนี้

4.1 เป็นการเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย ระหว่างแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส และสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง

4.2 เป็นการเปรียบเทียบการใช้ฟีเจอร์จากการทำโมเดลทางภาษา (Language Model) ผ่านวิธีของ Mikolov อันได้แก่ คอนตินิวอัสแบกออฟเวิด (Continuous Bag-of-Words Model, CBOW) และ สคริปแกรม (Continuous Skip-Gram Model)

4.3 เป็นการเปรียบเทียบการใช้ฟีเจอร์กลุ่มต่างๆ เพื่อเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย

4.4 เป็นการทดสอบประสิทธิภาพของโมเดลที่สร้างจากกลุ่มของฟีเจอร์ระดับคำร่วมกับตัวอักษร และระดับคำ ระดับตัวอักษร ร่วมกับ POS ซึ่งทำการทดสอบกับข้อมูล ORCHID และ scb-mt-en-th-2020

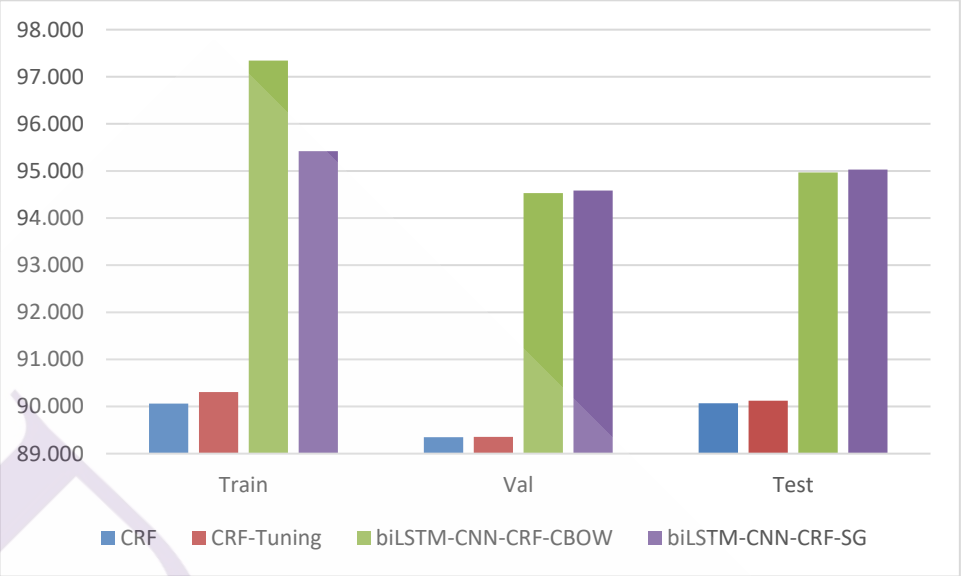
4.1 เปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย ระหว่างแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส และสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง

จากผลการทดลองเปรียบเทียบระหว่างการใช้แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส และสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง โดยเทียบกลุ่มของฟีเจอร์ที่ใช้ทั้งหมดพบว่า ผลการทดลองถูกแสดงในภาพที่ 4.1

จากผลการทดลองของชุดข้อมูลที่แยกไว้เพื่อทำการทดสอบโมเดลขั้นต้น (Validation Data) แสดงให้เห็นว่า ประสิทธิภาพของโมเดลการตัดประโยคภาษาไทยที่ถูกสร้างด้วยกลุ่มของฟีเจอร์ที่มาจาก คำ ตัวอักษร หน้าที่ของคำ และชื่อเฉพาะ พบว่าโมเดลจากสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง ให้ผลความแม่นยำดีที่สุด โดยที่โมเดลที่สร้างจากแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ให้ประสิทธิภาพการตัดประโยคภาษาไทย F1-Macro ที่ 89.349 % ในขณะที่ แบบจำลองคอนดิชันนอลแรนดอมฟิลด์สที่ได้รับการปรับปรุงให้ประสิทธิภาพที่ 89.356 % ซึ่งน้อยกว่า สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง ที่ถูกสร้างโดยใช้ฟีเจอร์ของโมเดลทางภาษาที่สร้างจากเทคนิค CBOW และ Skip-Gram อยู่ที่ 94.530 % และ 94.583 % ตามลำดับ

ทั้งนี้หากเปรียบเทียบผลการทดลองจากชุดข้อมูลที่แยกไว้เพื่อทำการทดสอบสุดท้าย (Testing Data) พบว่าประสิทธิภาพของโมเดลที่ดีที่สุดของการตัดประโยคภาษาไทย เป็นโมเดลที่ถูกสร้างขึ้นจาก สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง F1-Macro ที่ 90.069 % ในขณะที่โมเดลที่สร้างจากแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ที่ได้รับการปรับปรุงให้ประสิทธิภาพการตัดประโยคภาษาไทยที่ 90.123 % ในขณะ โดยที่ทั้งสองโมเดลให้ประสิทธิภาพการตัดประโยคภาษาไทยน้อยกว่า สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง ที่ถูกสร้างโดยใช้ฟีเจอร์ของโมเดลทางภาษาที่สร้างจากเทคนิค CBOW และ Skip-Gram อยู่ที่ 94.970 % และ 95.030 % ตามลำดับ

จากผลการทดลองแสดงให้เห็นว่าโมเดลที่ถูกสร้างขึ้นโดยใช้เทคนิคการเรียนรู้เชิงลึกร่วมกับสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง สามารถสร้างโมเดลเพื่อใช้ในการตัดประโยคภาษาไทยได้มีประสิทธิภาพมากกว่า การใช้แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ซึ่งเป็นเทคนิคดั้งเดิมที่ใช้กัน และเมื่อเปรียบเทียบประสิทธิภาพของโมเดลทั้งจากชุดข้อมูลวาลิเดชั่น และชุดข้อมูลเทสติ้งจาก สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง ที่ถูกสร้างโดยใช้ฟีเจอร์ของโมเดลทางภาษาที่สร้างจากเทคนิค CBOW และ Skip-Gram พบว่า Skip-Gram ให้ผลลัพธ์ที่มีประสิทธิภาพมากที่สุดในการทดสอบโมเดล



ภาพที่ 4.1 กราฟแท่งแสดงการเปรียบเทียบค่าเฉลี่ยประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบเทคนิคระหว่างเทคนิคดั้งเดิม หรือ CRF เปรียบเทียบกับ Bi-LSTM-CNN-CRF

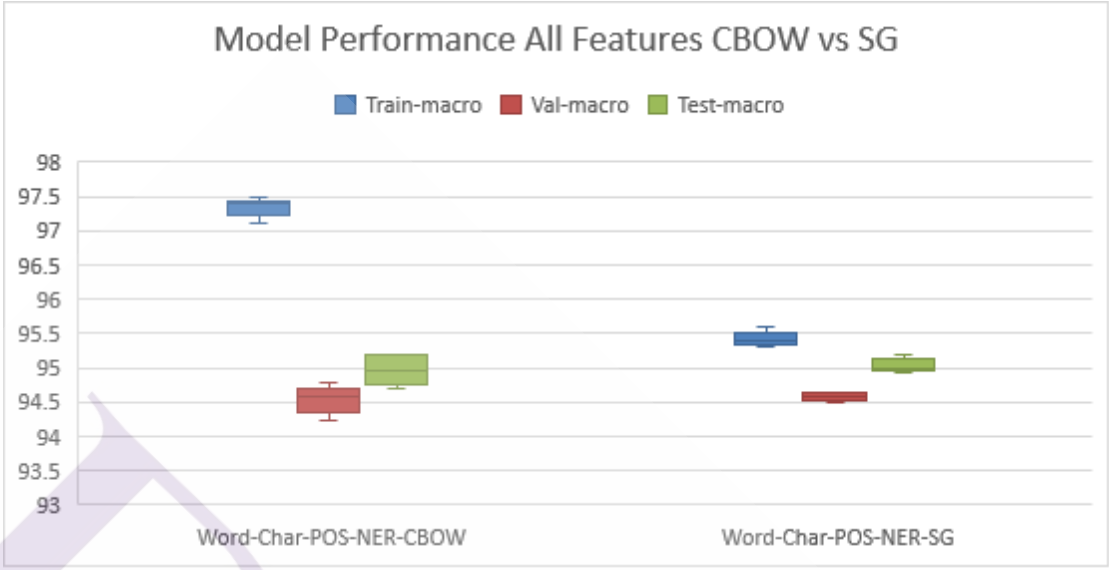
ตารางที่ 4.1 ตารางแสดงการเปรียบเทียบค่าเฉลี่ยประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบเทคนิคระหว่างเทคนิคดั้งเดิม หรือ CRF เปรียบเทียบกับ Bi-LSTM-CRF

Model	Train	Val	Test
CRF	90.061	89.349	90.069
CRF-Tuning	90.308	89.356	90.123
biLSTM-CNN-CRF-CBOW	97.344	94.530	94.970
biLSTM-CNN-CRF-SG	95.422	94.583	95.030

4.2 ผลการเปรียบเทียบการใช้ฟีเจอร์จากการทำโมเดลทางภาษา (Language Model) ผ่านวิธีของ Mikolov อันได้แก่ คอนตินิวอัสแบกออฟเวิด (Continuous Bag-of-Words Model, CBOW) และ สคริปแกรม (Continuous Skip-Gram Model)

จากผลการทดลองในภาพที่ 4.1 พบว่าผลการประสิทธิภาพของโมเดลตัดประโยคภาษาไทยที่สร้างขึ้นจากการผสมผสานกันของสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง ร่วมกับ สถาปัตยกรรมโครงข่ายประสาทแบบคอนโวลูชัน และแบบแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ได้ให้ผลลัพธ์การตัดประโยคที่มีประสิทธิภาพมากกว่าการใช้แบบแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส เพียงโมเดลเดียว

ดังนั้นผู้วิจัยจึงได้ทำการเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย ที่สร้างขึ้นมาจากการใช้โมเดลทางภาษาที่ต่างกัน นั่นคือ CBOW และ Skip-gram รวมถึงการทดลองการใช้กลุ่มของฟีเจอร์ในระดับต่างๆกัน โดยเริ่มจากกลุ่มที่มีจำนวนฟีเจอร์มากที่สุด นั่นคือการใช้ฟีเจอร์ ระดับคำ ระดับตัวอักษร หน้าที่ของคำ (Parts of Speech) และคำเฉพาะ (Name Entities) จากการทดลองพบว่า ประสิทธภาพของโมเดลที่ถูกทดสอบโดยชุดข้อมูลเทรนนิ่ง Training Dataset พบว่าเวิร์ดเอมเบดดิ้ง ที่มาจาก CBOW ดีกว่า Skip-gram โดยความแม่นยำ F1-Macro การตัดประโยคเฉลี่ยอยู่ที่ 97.344 % และ 95.422 % ตามลำดับ หากวัดที่ชุดข้อมูลสำหรับการทดสอบ Validation Dataset ความแม่นยำ F1-Macro เฉลี่ยของการตัดประโยคอยู่ที่ 94.530 % และ 94.580 % ตามลำดับ และ ที่ชุดข้อมูลสำหรับการทดสอบขั้นสุดท้าย Testing Dataset ความแม่นยำ F1-Macro เฉลี่ยของการตัดประโยคอยู่ที่ 94.970 % และ 95.030 % ตามลำดับ ซึ่งหากพิจารณาที่ Validation Dataset แล้วนั้นพบว่า ถึงแม้จะมีบางชุดข้อมูลทดสอบแล้ว CBOW ได้ F1-Macro ที่มากกว่า Skip-gram แต่ค่าเฉลี่ย F1-Macro ของ Skip-gram นั้นสูงกว่าทั้งในชุดข้อมูล Validation Dataset และ Testing Dataset ซึ่งแสดงดังในภาพที่ 4.2



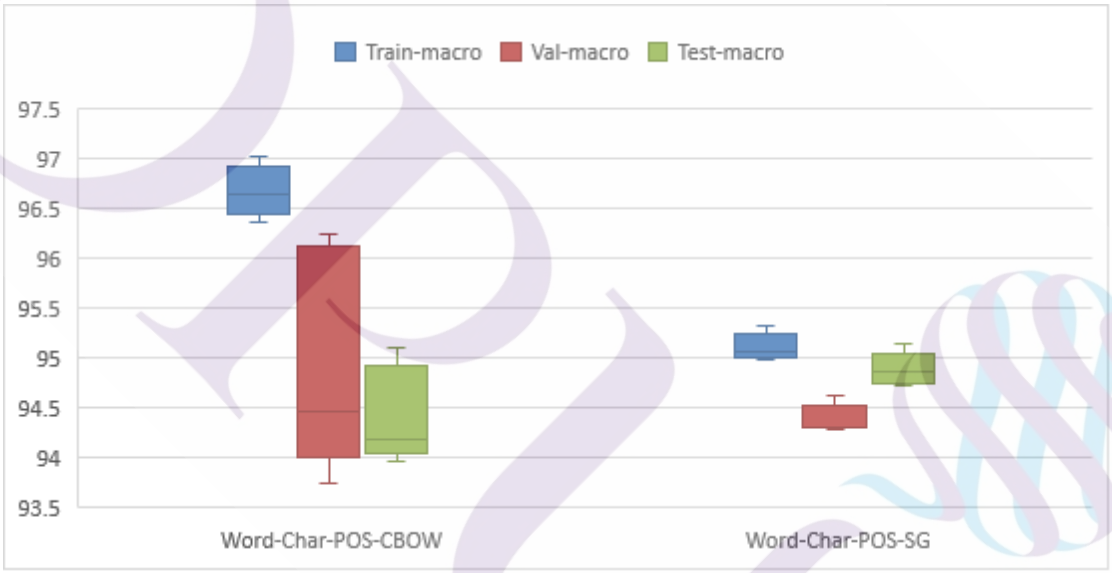
ภาพที่ 4.2 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs

ตารางที่ 4.2 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs ซึ่งเป็นผลลัพธ์ทั้ง 5 รอบที่ทำการทดสอบ

Model BiLSTM-CNN-CRF	Train-macro	Val-macro	Test-macro
Word-Char-POS-NER-CBOW	97.095	94.246	94.690
Word-Char-POS-NER-CBOW	97.388	94.442	94.809
Word-Char-POS-NER-CBOW	97.389	94.570	94.971
Word-Char-POS-NER-CBOW	97.369	94.774	95.198
Word-Char-POS-NER-CBOW	97.478	94.617	95.183
Word-Char-POS-NER-SG	95.435	94.624	94.962
Word-Char-POS-NER-SG	95.378	94.588	94.974
Word-Char-POS-NER-SG	95.603	94.572	95.177
Word-Char-POS-NER-SG	95.381	94.647	95.1
Word-Char-POS-NER-SG	95.312	94.485	94.936

ทางผู้วิจัยได้ทำการทดลองฟีเจอร์ต่ออีกสามกลุ่มโดยที่กลุ่มต่อมาทำการพิจารณาใช้เพียง คำเฉพาะ ร่วมกับฟีเจอร์ระดับคำ และตัวอักษร ได้ผลลัพธ์ดังภาพที่ 4.3 และพิจารณาใช้เพียงหน้าที่ของคำ ร่วมกับฟีเจอร์ระดับคำ และตัวอักษร ได้ผลลัพธ์ดังภาพที่ 4.4

สำหรับฟีเจอร์ที่ใช้ในภาพที่ 4.3 ผู้วิจัยได้ออกแบบเพื่อลดการใช้งานฟีเจอร์จำนวนมาก โดยการใช้ฟีเจอร์เหลือเพียงกลุ่มของระดับคำ ระดับตัวอักษร และ POS โดยที่ประสิทธิภาพเฉลี่ยของโมเดล F1-macro โดยวัดใน Training Dataset พบว่า โมเดลภาษาที่ใช้เทคนิค CBOW ให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 96.684 % และ 95.120 % ตามลำดับ เมื่อวัดใน Validation Dataset พบว่า เทคนิค CBOW ยังคงให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 94.945 % และ 94.392 % ตามลำดับ แต่หากพิจารณาที่ Testing Dataset ประสิทธิภาพของโมเดลจากเทคนิคของ Skip-gram ให้ประสิทธิภาพที่ดีกว่า CBOW ที่ 94.898 %, 94.429 % ตามลำดับ

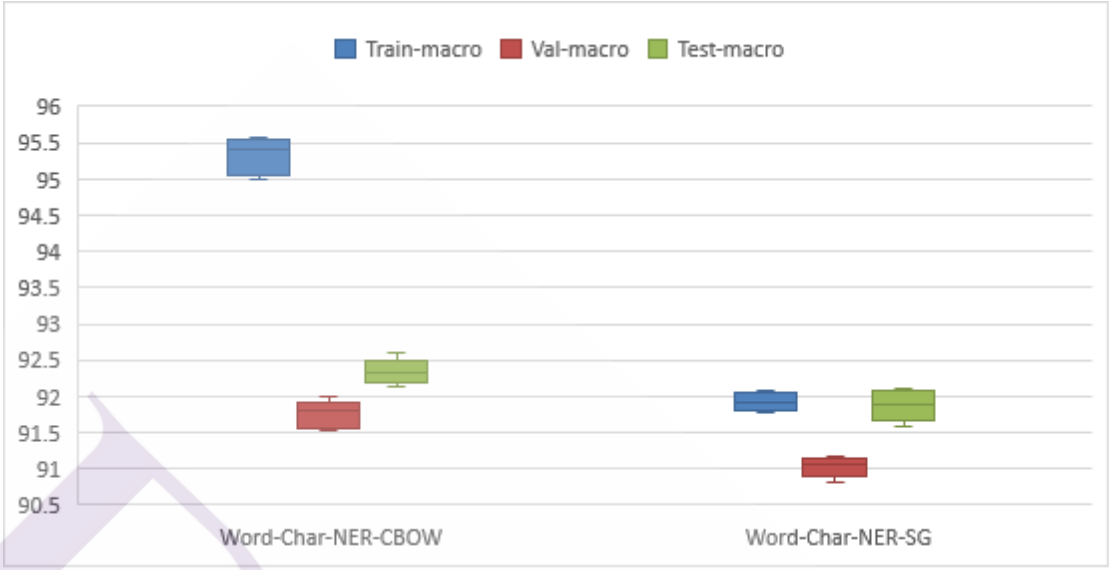


ภาพที่ 4.3 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

ตารางที่ 4.3 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS ซึ่งเป็นผลลัพธ์ทั้ง 5 รอบที่ทำการทดสอบ

Model BiLSTM-CNN-CRF	Train-macro	Val-macro	Test-macro
Word-Char-POS-CBOW	96.522	93.743	94.194
Word-Char-POS-CBOW	96.656	96.255	94.138
Word-Char-POS-CBOW	96.364	95.986	93.963
Word-Char-POS-CBOW	97.032	94.46	95.101
Word-Char-POS-CBOW	96.845	94.282	94.748
Word-Char-POS-SG	95.18	94.417	94.951
Word-Char-POS-SG	94.998	94.296	94.723
Word-Char-POS-SG	95.077	94.302	94.876
Word-Char-POS-SG	95.328	94.634	95.156
Word-Char-POS-NER-SG	95.019	94.31	94.782

ในภาพที่ 4.4 แสดงถึงผลลัพธ์ของฟีเจอร์ระดับคำ ระดับตัวอักษร และ NEs โดยที่ประสิทธิภาพเฉลี่ยของโมเดล F1-macro โดยวัดใน Training Dataset พบว่า โมเดลภาษาที่ใช้เทคนิค CBOW ให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 95.315 % และ 91.920 % ตามลำดับ เมื่อวัดใน Validation Dataset พบว่า เทคนิค CBOW ยังคงให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 91.748 % และ 91.025 %ตามลำดับ และที่ชุดข้อมูล Testing Dataset ประสิทธิภาพของโมเดลจากเทคนิคของ CBOW ยังคงให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 92.335 %, 91.876 % ตามลำดับ จะเห็นว่าที่ Training Dataset พบว่า ประสิทธิภาพการตัดประโยคของโมเดล CBOW มีระยะห่างจากการทดสอบโมเดลด้วยชุดข้อมูล Validation Dataset และ Testing Dataset โดยเมื่อเทียบกับ โมเดลภาษาที่ใช้เทคนิค Skip-gram ด้วยนั้นมีระยะห่างของประสิทธิภาพโมเดลเมื่อทดสอบด้วยชุดข้อมูล Training Dataset, Validation Dataset และ Testing Dataset ที่น้อยกว่า หรือใกล้เคียงกัน



ภาพที่ 4.4 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ NEs

ตารางที่ 4.4 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ NEs ซึ่งเป็นผลลัพธ์ทั้ง 5 รอบที่ทำการทดสอบ

Model BiLSTM-CNN-CRF	Train-macro	Val-macro	Test-macro
Word-Char-NER-CBOW	95.111	91.539	92.135
Word-Char-NER-CBOW	95.484	91.823	92.378
Word-Char-NER-CBOW	95.4	91.807	92.32
Word-Char-NER-CBOW	95.586	91.997	92.614
Word-Char-NER-CBOW	94.996	91.576	92.23
Word-Char-NER-SG	92.066	91.095	92.065
Word-Char-NER-SG	92.028	91.182	92.112
Word-Char-NER-SG	91.817	90.972	91.879
Word-Char-NER-SG	91.766	90.817	91.581
Word-Char-NER-SG	91.923	91.058	91.741

ในภาพที่ 4.5 แสดงถึงผลลัพธ์ของฟีเจอร์ระดับคำ และระดับตัวอักษร ซึ่งเป็นสองฟีเจอร์ที่ไม่ต้องผ่านการสกัดเพื่อนำ POS และ NEs ออกมาใช้เพิ่ม โดยที่ประสิทธิภาพเฉลี่ยของโมเดล F1-macro โดยวัดใน Training Dataset พบว่า โมเดลภาษาที่ใช้เทคนิค CBOW ให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 94.901 % และ 91.660 % ตามลำดับ เมื่อวัดใน Validation Dataset พบว่า เทคนิค CBOW ยังคงให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 92.132 %, 91.013 %ตามลำดับ และที่ชุดข้อมูล Testing Dataset ประสิทธิภาพของโมเดลจากเทคนิคของ CBOW ยังคงให้ประสิทธิภาพที่ดีกว่า Skip-gram ที่ 92.130 %, 91.730 % ตามลำดับ แต่จะเห็นได้ว่าที่ Training Dataset พบว่าประสิทธิภาพการตัดประโยคของโมเดล CBOW มีระยะห่างจากการทดสอบโมเดลด้วยชุดข้อมูล Validation Dataset และ Testing Dataset โดยเมื่อเทียบกับ โมเดลภาษาที่ใช้เทคนิค Skip-gram ด้วยนั้นมีระยะห่างของประสิทธิภาพโมเดลเมื่อทดสอบด้วยชุดข้อมูล Training Dataset, Validation Dataset และ Testing Dataset ที่น้อยกว่า หรือใกล้เคียงกัน และใน Validation Dataset นั้นเทคนิคของ CBOW มีค่าประสิทธิภาพการตัดประโยคที่มีค่าพิสัยกว้างกว่าเทคนิคของ Skip-gram เป็นอย่างมาก

ภาพที่ 4.5 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ และระดับตัวอักษร

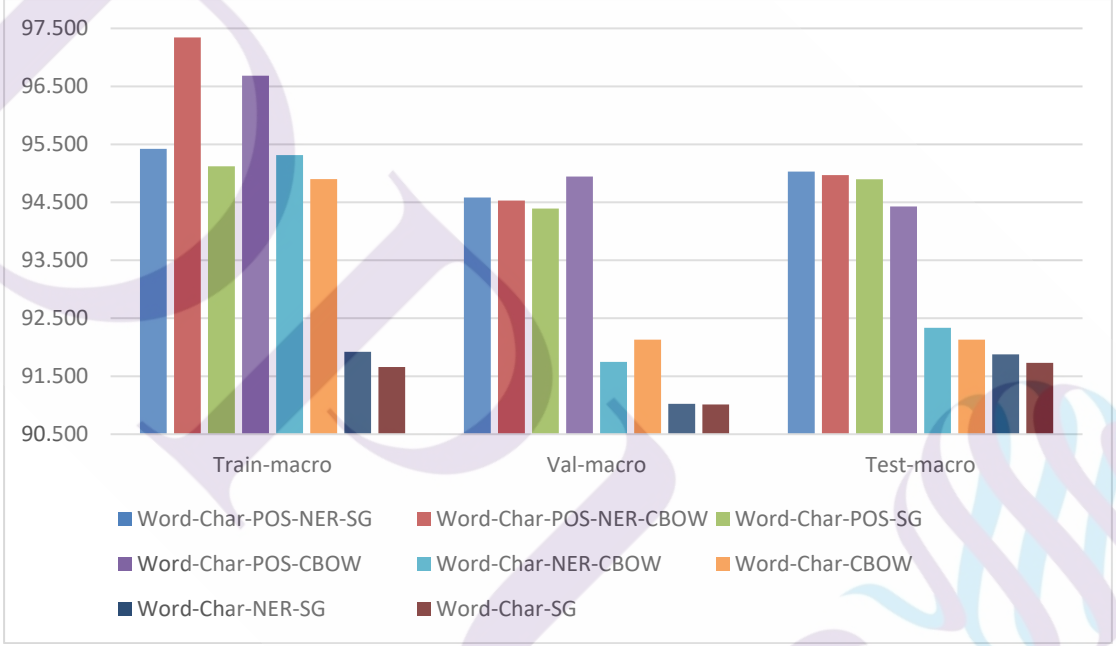
ตารางที่ 4.5 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ และระดับตัวอักษร ซึ่งเป็นผลลัพธ์ทั้ง 5 รอบที่ทำการทดสอบ

Model BiLSTM-CNN-CRF	Train-macro	Val-macro	Test-macro
Word-Char-CBOW	94.878	91.546	92.027
Word-Char-CBOW	95.176	94.615	92.268
Word-Char-CBOW	95.089	91.681	92.206
Word-Char-CBOW	94.800	91.552	92.284
Word-Char-CBOW	94.562	91.264	91.863
Word-Char-SG	91.705	90.812	91.734
Word-Char-SG	91.506	90.819	91.733
Word-Char-SG	91.607	90.815	91.715
Word-Char-SG	91.699	90.899	91.819
Word-Char-SG	91.784	91.719	91.648

4.3 ผลการเปรียบเทียบการใช้ฟีเจอร์กลุ่มต่างๆ เพื่อเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย

จากผลการทดลองสร้างโมเดลตัดประโยคภาษาไทยเมื่อพิจารณาเฉพาะโมเดลที่พัฒนาขึ้นมาจากการเรียนรู้เชิงลึก โดยทำการเรียนรู้จากกลุ่มของฟีเจอร์ที่แตกต่างกัน และการแปลงฟีเจอร์ในระดับคำจากโมเดลทางภาษาที่แตกต่างกัน พบว่าเมื่อวัดประสิทธิภาพของโมเดลด้วยชุดข้อมูล Testing Dataset โมเดลที่ดีที่สุดไปน้อยที่สุดได้ดังนี้ อันดับหนึ่งคือ โมเดลที่ใช้ฟีเจอร์ระดับคำ ตัวอักษร POS และ NEs ซึ่งใช้เวิร์ดเอมเบดดิ้งจาก Skip-gram มีค่า F1-macro เท่ากับ 95.030 % อันดับสองคือโมเดลที่ใช้ฟีเจอร์ระดับคำ ตัวอักษร POS และ NEs ซึ่งใช้เวิร์ดเอมเบดดิ้งจาก CBOW มีค่า F1-macro เท่ากับ 94.970 % อันดับสามคือโมเดลที่ใช้ฟีเจอร์ระดับคำ ตัวอักษร และ POS ซึ่งใช้เวิร์ดเอมเบดดิ้งจาก Skip-gram มีค่า F1-macro เท่ากับ 94.898 % อันดับสี่คือโมเดลที่ใช้ฟีเจอร์ระดับคำ ตัวอักษร และ POS ซึ่งใช้เวิร์ดเอมเบดดิ้งจาก CBOW มีค่า F1-macro เท่ากับ 94.429 % อันดับห้าคือโมเดลที่ใช้ฟีเจอร์ระดับคำ ตัวอักษร และ NEs ซึ่งใช้เวิร์ดเอมเบดดิ้งจาก CBOW มีค่า F1-macro

เท่ากับ 92.335 % ซึ่งในอันดับห้านั้นผลที่ได้ต่างออกไป โดยที่ ฟีเจอร์ 2 กลุ่มแรกนั้น Skip-gram ส่งผลให้ประสิทธิภาพของโมเดลมีค่าที่ดีกว่า CBOW แต่ในฟีเจอร์ NEs นั้น CBOW กลับให้ค่า F1-macro ที่ดีกว่า และในอันดับที่หก เป็นโมเดลที่ใช้เพียงระดับคำ และระดับตัวอักษร โดยฟีเจอร์ของคำนั้นมาจาก CBOW อีกเช่นกัน โดยที่ประสิทธิภาพของโมเดลทำคะแนนไปได้ 92.130 % และในอันดับเจ็ด และอันดับแปด ค่า F1-macro อยู่ที่ 91.876 % และ 91.730 % ตามลำดับ ซึ่งอันดับที่เจ็ดมาจากการใช้ฟีเจอร์ในระดับคำ ระดับตัวอักษร และ NEs เข้าร่วมด้วยกันกับ Skip-gram ท้ายที่สุดอันดับที่แปด มาจากการใช้ฟีเจอร์ระดับคำ และตัวอักษร เข้าร่วมกับ Skip-gram

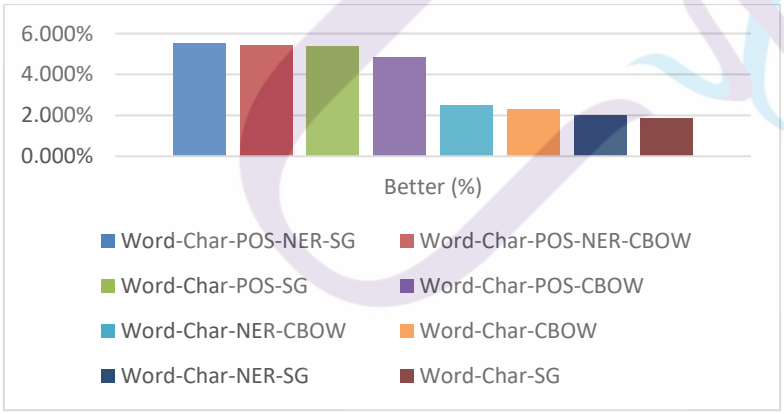


ภาพที่ 4.6 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพค่าเฉลี่ยของ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs

ตารางที่ 4.6 ตารางแสดงการเปรียบเทียบประสิทธิภาพค่าเฉลี่ยของ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs

Model BiLSTM-CNN-CRF	Train-macro	Val-macro	Test-macro
Word-Char-POS-NER-SG	95.422	94.583	95.030
Word-Char-POS-NER-CBOW	97.344	94.530	94.970
Word-Char-POS-SG	95.120	94.392	94.898
Word-Char-POS-CBOW	96.684	94.945	94.429
Word-Char-NER-CBOW	95.315	91.748	92.335
Word-Char-CBOW	94.901	92.132	92.130
Word-Char-NER-SG	91.920	91.025	91.876
Word-Char-SG	91.660	91.013	91.730

ผลการทดลองสร้างโมเดลตัดประโยคภาษาไทยเมื่อทำการเปรียบเทียบประสิทธิภาพของโมเดลที่สร้างขึ้นจากการเรียนรู้เชิงลึกร่วมกับฟีเจอร์ในรูปแบบต่างๆ เปรียบเทียบกับ โมเดลพื้นฐานที่สร้างขึ้นมาจากแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส ถูกแสดงในภาพที่ 4.7 และตารางที่ 4.7



ภาพที่ 4.7 กราฟกล่องแสดงการเปรียบเทียบประสิทธิภาพค่าเฉลี่ยของ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อทำการเปรียบเทียบประสิทธิภาพของโมเดลที่สร้างขึ้นจากการเรียนรู้เชิงลึกร่วมกับฟีเจอร์ในรูปแบบต่างๆ เปรียบเทียบกับ โมเดลพื้นฐานที่สร้างขึ้นมาจากแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส

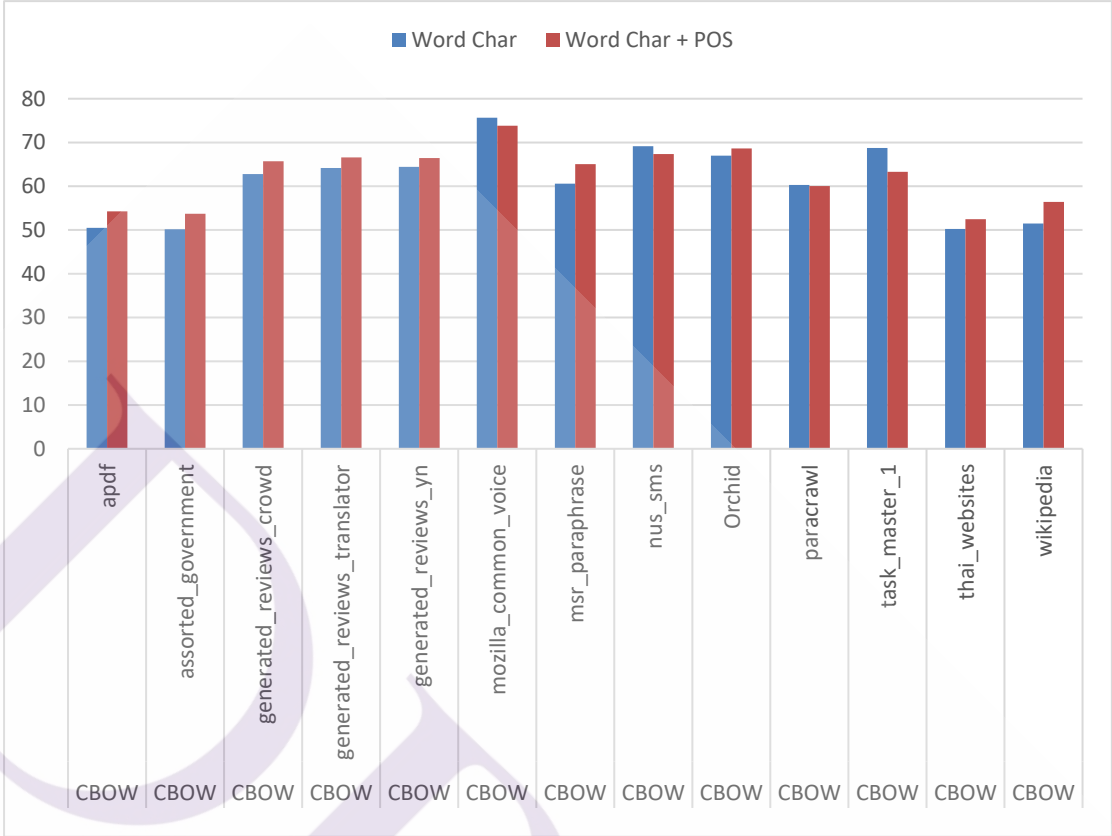
ตารางที่ 4.7 ตารางแสดงการเปรียบเทียบประสิทธิภาพค่าเฉลี่ยของ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา ระหว่าง CBOW และ Skip-gram โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทั้งสองกลุ่มใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs

Model BiLSTM-CNN-CRF	**% ที่ดีขึ้น**
Word-Char-POS-NER-SG	5.508
Word-Char-POS-NER-CBOW	5.441
Word-Char-POS-SG	5.361
Word-Char-POS-CBOW	4.841
Word-Char-NER-CBOW	2.516
Word-Char-CBOW	2.288
Word-Char-NER-SG	2.006
Word-Char-SG	1.844

4.4 เปรียบเทียบผลการทดสอบประสิทธิภาพของโมเดลที่สร้างจากกลุ่มของฟีเจอร์ระดับคำ ร่วมกับตัวอักษร และระดับคำ ระดับตัวอักษร ร่วมกับ POS ซึ่งทำการทดสอบกับชุดข้อมูล ORCHID และ scb-mt-en-th-2020

ภายหลังจากการเปรียบเทียบประสิทธิภาพการตัดประโยคภาษาไทยด้วยชุดข้อมูลทดสอบ Testing data และ Validation data ในหัวข้อที่ 4.1 ถึง 4.3 แล้ว ตึงได้ทำการนำโมเดลที่สร้างขึ้นจาก Bi-LSTM-CNN-CRF นำมาทดสอบกับชุดข้อมูลอื่นๆที่ไม่เคยเรียนรู้มาก่อน 2 ชุดข้อมูลนั่นคือชุดข้อมูล ORCHID และชุดข้อมูล scb-mt-en-th-2020 โดยทำการทดสอบบนกลุ่มของฟีเจอร์ระหว่าง กลุ่มของระดับคำ และตัวอักษร กับกุล่มระดับคำ ระดับตัวอักษร และ POS

เมื่อทำการพิจารณาในกลุ่มของโมเดลภาษาที่สร้างขึ้นมาจาก CBOW โดยเปรียบเทียบระหว่าง 2 กลุ่มฟีเจอร์ พบว่าชุดข้อมูล ORCHID นั้นการเพิ่ม POS ฟีเจอร์ส่งผลให้ประสิทธิภาพการตัดประโยคดีขึ้นจากเดิม ซึ่งเป็นไปตามผลการทดลองกับชุดข้อมูลของ LST20 ในขณะที่กลุ่มชุดข้อมูล scb-mt-en-th-2020 พบว่ามีจำนวน 4 ใน 12 ชุดข้อมูลที่พบว่าการใช้ฟีเจอร์เพียงระดับคำ และระดับตัวอักษรนั้นสามารถตัดประโยคภาษาไทยได้ดีกว่าการเพิ่ม POS เข้าไป ซึ่งได้แก่ชุดข้อมูล mozilla_common_voice, nus_sms, task_master_1 และ paracrawl โดยแสดงผลไว้ในภาพที่ 4.8

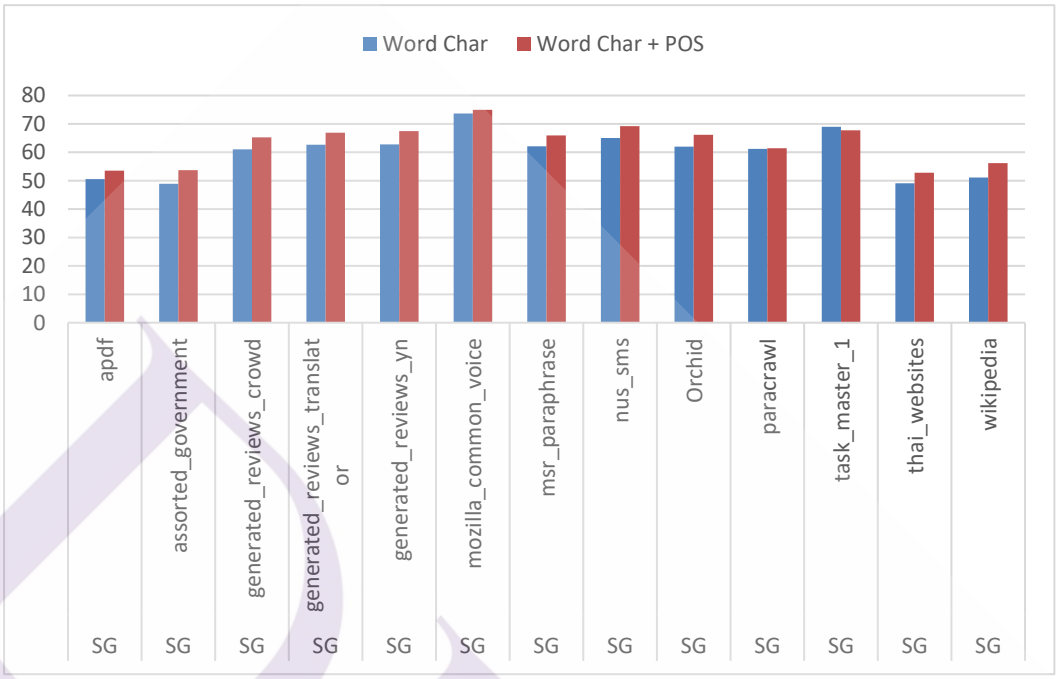


ภาพที่ 4.8 กราฟแท่งแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา CBOW โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ระหว่างทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร เปรียบเทียบกับ การใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

ตารางที่ 4.8 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา CBOW โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ระหว่างทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร เปรียบเทียบกับ การใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

Dataset	Word Char	Word Char + POS
apdf	50.489	54.262
assorted_government	50.156	53.699
generated_reviews_crowd	62.776	65.699
generated_reviews_translator	64.168	66.58
generated_reviews_yn	64.431	66.428
mozilla_common_voice	75.668	73.843
msr_paraphrase	60.6	65.051
nus_sms	69.148	67.349
Orchid	66.974	68.621
paracrawl	60.3	60.057
task_master_1	68.735	63.311
thai_websites	50.231	52.45
wikipedia	51.491	56.406

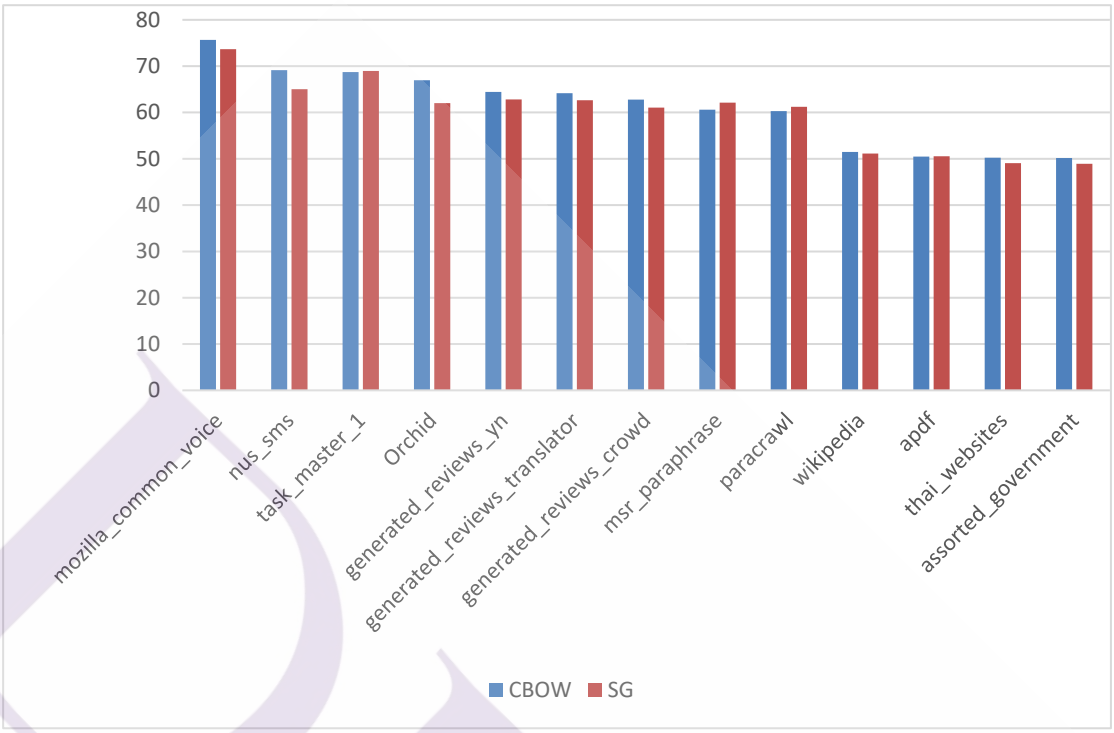
ในอีกมุมหนึ่งเมื่อทำการพิจารณาในกลุ่มของโมเดลภาษาที่สร้างขึ้นมาจาก Skip-Gram โดยเปรียบเทียบระหว่าง 2 กลุ่มฟีเจอร์ พบว่าชุดข้อมูล ORCHID นั้นการเพิ่ม POS ฟีเจอร์ส่งผลให้ประสิทธิภาพการตัดประโยคดีขึ้นจากเดิม ซึ่งเป็นไปตามผลการทดลองกับชุดข้อมูลของ LST20 ในขณะที่กลุ่มชุดข้อมูล scb-mt-en-th-2020 พบว่ามีจำนวนเพียง 1 ใน 12 ชุดข้อมูลที่พบว่าการใช้ฟีเจอร์เพียงระดับคำ และระดับตัวอักษรนั้นสามารถตัดประโยคภาษาไทยได้ดีกว่าการเพิ่ม POS เข้าไป ซึ่งได้แก่ชุดข้อมูล task_master_1 โดยแสดงผลไว้ในภาพที่ 4.9



ภาพที่ 4.9 กราฟแท่งแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ระหว่างทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร เปรียบเทียบกับ การใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

ตารางที่ 4.9 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษา SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ระหว่างทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร เปรียบเทียบกับการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

Dataset	Word Char	Word Char + POS
apdf	50.561	53.523
assorted_government	48.919	53.718
generated_reviews_crowd	61.067	65.247
generated_reviews_translator	62.648	66.902
generated_reviews_yn	62.801	67.45
mozilla_common_voice	73.666	74.94
msr_paraphrase	62.121	65.936
nus_sms	65.03	69.212
Orchid	62.005	66.159
paracrawl	61.231	61.451
task_master_1	68.964	67.733
thai_websites	49.068	52.812
wikipedia	51.127	56.218



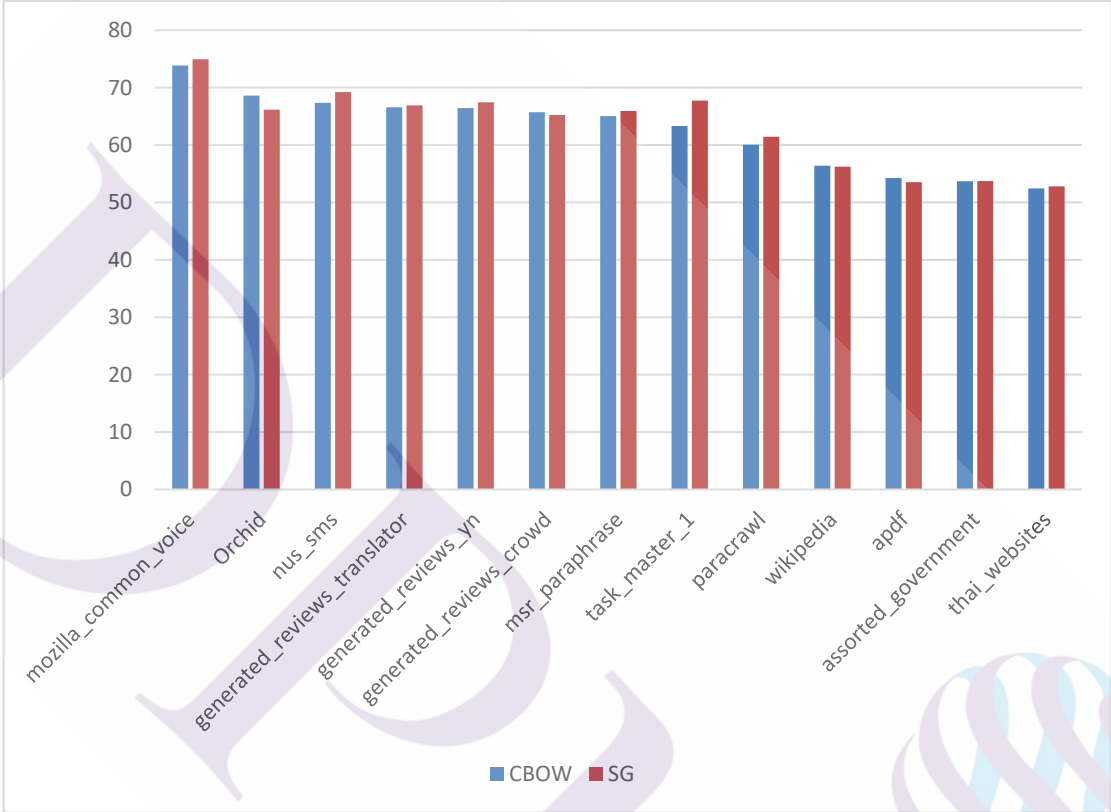
ภาพที่ 4.10 กราฟแท่งแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษาระหว่าง CBOW และ SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทำการใช้ฟีเจอร์ระดับคำ และระดับตัวอักษร

ตารางที่ 4.10 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษาระหว่าง CBOW และ SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทำการใช้ฟีเจอร์ระดับคำและระดับตัวอักษร

Dataset	CBOW	SG
mozilla_common_voice	73.843	74.94
nus_sms	68.621	66.159
task_master_1	67.349	69.212
Orchid	66.58	66.902
generated_reviews_yn	66.428	67.45
generated_reviews_translator	65.699	65.247
generated_reviews_crowd	65.051	65.936
msr_paraphrase	63.311	67.733
paracrawl	60.057	61.451
wikipedia	56.406	56.218
apdf	54.262	53.523
thai_websites	53.699	53.718
assorted_government	52.45	52.812

เมื่อทำการพิจารณาประสิทธิภาพของโมเดลโดยเปรียบเทียบ โมเดลทางภาษาที่ใช้ระหว่าง ORCHID และ Skip-gram ในกลุ่มของฟีเจอร์ระดับคำ และตัวอักษร พบว่าชุดข้อมูล ORCHID นั้นการใช้ CBOW มีประสิทธิภาพการตัดประโยคที่ดีกว่า Skip-gram ซึ่งเป็นไปตามผลการทดลองกับชุดข้อมูลของ LST20 ในขณะที่กลุ่มชุดข้อมูล scb-mt-en-th-2020 พบว่ามีจำนวน 3 ใน 12 ชุดข้อมูลที่พบว่าการใช้ Skip-gram ส่งผลดีกว่า ซึ่งได้แก่ชุดข้อมูล msr_paraphrase, paracrawl และ apdf โดยสำหรับชุดข้อมูล apdf มีความต่างกันเพียงเล็กน้อย โดยผลการทดลองแสดงไว้ในภาพที่ 4.10 และเมื่อทำการพิจารณาประสิทธิภาพของโมเดลโดยเปรียบเทียบ โมเดลทางภาษาที่ใช้ระหว่าง ORCHID และ Skip-gram ในกลุ่มของฟีเจอร์ระดับคำ ตัวอักษร และ POS พบว่าชุดข้อมูล ORCHID นั้นการใช้ CBOW มีประสิทธิภาพการตัดประโยคที่ดีกว่า Skip-gram ซึ่งต่างจากผลลัพธ์

ของการทดลองกับชุดข้อมูลของ LST20 ในขณะที่กลุ่มชุดข้อมูล scb-mt-en-th-2020 พบว่ามีจำนวน 3 ใน 12 ชุดข้อมูลที่พบว่าการใช้ CBOW ส่งผลดีกว่า ซึ่งได้แก่ชุดข้อมูล generated_reviews_crowd, Wikipedia และ apdf โดยผลการทดลองแสดงไว้ในภาพที่ 4.11



ภาพที่ 4.11 กราฟแท่งแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษาระหว่าง CBOW และ SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

ตารางที่ 4.11 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษาระหว่าง CBOW และ SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

Dataset	CBOW	SG
mozilla_common_voice	75.668	73.666
nus_sms	69.148	65.03
task_master_1	68.735	68.964
Orchid	66.974	62.005
generated_reviews_yn	64.431	62.801
generated_reviews_translator	64.168	62.648
generated_reviews_crowd	62.776	61.067
msr_paraphrase	60.6	62.121
paracrawl	60.3	61.231
wikipedia	51.491	51.127
apdf	50.489	50.561
thai_websites	50.231	49.068
assorted_government	50.156	48.919

บทที่ 5
บทสรุปและข้อเสนอแนะ

งานวิจัยในวิทยานิพนธ์เล่มนี้ได้กล่าวถึงการวิจัยการสร้างโมเดลตัดประโยคภาษาไทยด้วยเทคนิคการเรียนรู้เชิงลึก โดยใช้ข้อมูลจากทั้งหมด 3 แหล่งข้อมูล ORCHID, scb-mt-en-th-2020 และ LST20 ซึ่งทั้งสามแหล่งข้อมูลถูกรวบรวมนำมาสร้างโมเดลทางภาษาเพื่อใช้เป็นฟีเจอร์เวิร์ดเอมเบดดิ้งที่มีการสร้างจากเทคนิค CBOW และ Skip-gram ในการทดลองการสร้างโมเดลตัดประโยคภาษาไทย ผู้วิจัยได้ทำการทดลองวัดประสิทธิภาพบนชุดข้อมูล LST20 โดยทำการเปรียบเทียบประสิทธิภาพของโมเดลการเรียนรู้เชิงลึกกับ โมเดลที่เป็นพื้นฐาน หรือ แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส โดยมีการแบ่งฟีเจอร์ที่ใช้ในการเรียนรู้เป็น 4 กลุ่ม นั่นคือ กลุ่มของระดับคำ ระดับตัวอักษร POS และ NEs กลุ่มที่สอง คือ กลุ่มของระดับคำ ระดับตัวอักษร และ POS กลุ่มที่สาม คือ คือ กลุ่มของระดับคำ ระดับตัวอักษร และ NEs กลุ่มสุดท้าย คือ กลุ่มของระดับคำและระดับตัวอักษร โดยโมเดลเชิงลึกที่ใช้ในการสร้างโมเดลตัดประโยคภาษาไทยนั้น มีการออกแบบมาจากการใช้สถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง เพื่อทำการเรียนรู้ข้อมูลที่มีความสัมพันธ์กันจากคำหนึ่งสู่อีกคำหนึ่ง โดยนำมาใช้ร่วมกับสถาปัตยกรรมโครงข่ายประสาทเทียมแบคอนโวลูชั่น เพื่อใช้ในการสกัดฟีเจอร์ที่สำคัญออกมา และสุดท้ายนำมาประยุกต์กับ แบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส หรือผู้วิจัยตั้งชื่อว่า บอยด์คัท หรือเรียกว่า Bi-LSTM-CNN-CRF (Bi-Directional Convolutional Neural Network with Conditional Random Field) หลังจากนั้นจึงทำการวัดประสิทธิภาพของโมเดลด้วย F1-macro

5.1 สรุปผลการศึกษา

ในส่วนของการสรุปผลการทดลอง ผู้ทดลองได้ทำการแบ่งออกเป็นสองส่วน ในส่วนแรกคือการสรุปถึงผลลัพธ์ของโมเดลที่ใช้ในการตัดประโยคภาษาไทยที่ถูกสร้างขึ้น และทดสอบด้วยชุดข้อมูล LST20 และส่วนที่สอง ทำการสรุปผลจากการนำโมเดลที่ถูกสร้างขึ้นไปทดสอบประสิทธิภาพกับชุดข้อมูลที่ไม่ได้ทำการเรียนรู้มาก่อน ด้วยชุดข้อมูล ORCHID และ scb-mt-en-th-2020

5.1.1 สรุปจากโมเดลที่ถูกสร้าง และทดสอบด้วยชุดข้อมูล LST20

จากผลการทดลองการสร้างโมเดลตัดประโยคภาษาไทยด้วยโมเดล Bi-LSTM-CNN-CRF พบว่ามีประสิทธิภาพการตัดประโยคที่สูงกว่าโมเดลแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส โดยเมื่อเทียบประสิทธิภาพของโมเดลจากฟีเจอร์ที่มีการใช้ร่วมกันของระดับคำ ระดับตัวอักษร POS และ NEs พบว่า ประสิทธิภาพของโมเดล Bi-LSTM-CNN-CRF สูงกว่า CRF ร้อยละ 5.441 และ 5.508 ของ F1-macro เมื่อเทียบกับเทคนิคระหว่าง CBOW และ Skip-gram ดังนั้น การสร้างโมเดลตัดประโยคทางภาษาไทย การใช้โมเดลที่ประกอบด้วยเทคนิคการเรียนรู้เชิงลึก หรือ การประยุกต์ใช้ CRF เข้าร่วมกับสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง และสถาปัตยกรรมโครงข่ายประสาทเทียมแบคอนโวลูชั่น นั้นส่งผลให้ประสิทธิภาพการตัดประโยคของโมเดลมีประสิทธิภาพที่ดียิ่งขึ้น

ในขณะเดียวกันหากผู้ใช้งานต้องการสร้างโมเดลตัดประโยคภาษาไทย จาก Bi-LSTM-CNN-CRF การใช้ฟีเจอร์ทั้งระดับคำ ระดับตัวอักษร POS และ NEs เข้าร่วมในการสอนโมเดลประกอบกับการใช้เทคนิค Skip-gram จะส่งผลให้โมเดลมีประสิทธิภาพสูงที่สุดในการทดลอง แต่ถ้าหากผู้ใช้ต้องการทำโมเดลที่ใช้เพียงฟีเจอร์ระดับคำ และตัวอักษร สามารถเลือกใช้ โมเดลที่สร้างจาก Bi-LSTM-CNN-CRF ประกอบกับการใช้เทคนิค CBOW จะส่งผลให้ได้โมเดลตัดประโยคภาษาไทยที่มีประสิทธิภาพสูงสุด เมื่อเทียบกับ โมเดลที่ใช้ฟีเจอร์เดียวกันนั้น CBOW มีประสิทธิภาพสูงกว่า Skip-gram ร้อยละ 0.436 และสูงกว่าโมเดลที่สร้างจาก CRF แต่ทำการใช้ฟีเจอร์ทั้งในระดับคำ ระดับตัวอักษร POS และ NEs ถึงร้อยละ 2.288

ทั้งนี้หากผู้ใช้ต้องการสร้างโมเดลตัดประโยคภาษาไทยด้วยฟีเจอร์ที่ประกอบด้วย POS และ NEs โดยใช้ชุดข้อมูลจากแหล่งอื่นๆประกอบด้วยจำเป็นต้องสร้างโมเดลที่ใช้สำหรับการสกัด POS และ NEs เพิ่มขึ้นอีก 2 โมเดลเพื่อใช้ในการสกัดฟีเจอร์ และนำฟีเจอร์ที่ได้นั้นมาเป็นชุดข้อมูลเพื่อใช้ในการสอนโมเดลตัดประโยคภาษาไทยอีกถอดหนึ่ง ในขณะเดียวกันหากจำเป็นต้องใช้งานโมเดลตัดประโยคภาษาไทยที่ต้องใช้ฟีเจอร์ POS และ NEs ประกอบด้วยนั้น ทางผู้ใช้จะต้องสร้างอีก 2 โมเดลเพื่อใช้ในการสกัดฟีเจอร์เช่นเดียวกัน ทำให้ในการใช้งานโมเดลตัดประโยคภาษาไทยนั้นมีความยากลำบากในการใช้งานเพิ่มขึ้น หากแต่ผู้ใช้งานต้องการความสะดวกสบาย และเลือกที่จะไม่ทำโมเดลสกัด POS และ NEs เพิ่มแล้วนั้น สามารถเลือกสร้างโมเดลจาก Bi-LSTM-CNN-CRF ประกอบกับการใช้เทคนิค CBOW เพื่อสร้างโมเดลตัดประโยคภาษาไทยที่มีประสิทธิภาพสูงขึ้นมาได้ แม้มีฟีเจอร์ที่ใช้ประกอบการสร้างเพียง ระดับคำ และระดับตัวอักษรเท่านั้น

การเลือกใช้เทคนิค Skip-gram ในการสร้างโมเดลทางภาษา จะส่งผลให้ประสิทธิภาพของโมเดลตัดประโยคภาษาไทยสูงยิ่งขึ้น ถ้ามีการใช้งานฟีเจอร์กลุ่มของทั้งในระดับคำ ระดับ

ตัวอักษร POS และ NEs เข้าประกอบกัน และในกลุ่มของระดับคำ ระดับตัวอักษร และ POS ในทางกลับกันหากใช้ฟีเจอร์ในกลุ่มของระดับ ระดับคำ ระดับตัวอักษร และ NEs เข้าประกอบกัน และในกลุ่มของระดับคำ และระดับตัวอักษร นั้น การเลือกใช้เทคนิค CBOW เป็นทางเลือกที่เหมาะสมกว่า

5.1.2 สรุปผลจากการนำโมเดลที่ถูกสร้างขึ้นไปทดสอบประสิทธิภาพกับชุดข้อมูลที่ไม่ได้ทำการเรียนรู้มาก่อน ด้วยชุดข้อมูล ORCHID และ scb-mt-en-th-2020

จากผลการทดลองทดสอบประสิทธิภาพการตัดประโยคของโมเดลที่ถูกสร้างขึ้นจาก Bi-LSTM-CNN-CRF ที่มีการใช้ฟีเจอร์ที่แตกต่างกันสองเซต นั่นคือ ฟีเจอร์ในกลุ่มของคำ และตัวอักษร และเซตของกลุ่มคำ ตัวอักษร และ POS ทำการทดสอบบนชุดข้อมูลที่ไม่เคยเรียนรู้มาก่อน นั่นคือ ORCHID และ scb-mt-en-th-2020 พบว่าการเพิ่มฟีเจอร์ POS มีส่วนช่วยเพิ่มประสิทธิภาพการตัดประโยคถึง 21 ตัวอย่างชุดข้อมูลจากทั้งหมด 26 ชุดข้อมูล แสดงดังตารางที่ 5.1 และหากแบ่งเป็นกลุ่มของโมเดลที่สร้างจาก CBOW และ Skip-Gram แล้วพบว่าโมเดลที่สร้างจาก Skip-Gram นั้นหากทำการเพิ่ม POS เข้าไปในฟีเจอร์สามารถส่งผลให้ประสิทธิภาพการตัดประโยคดีขึ้นถึง 12 ชุดข้อมูลจาก 13 ชุดข้อมูล ในขณะเดียวกันการใช้ CBOW พบว่ามีจำนวนชุดข้อมูลที่ไม่ได้มีประสิทธิภาพที่ดีขึ้นหลังการเพิ่มฟีเจอร์ POS ถึง 4 ชุดข้อมูล จาก 13 ชุดข้อมูล

ตารางที่ 5.1 ตารางแสดงการเปรียบเทียบประสิทธิภาพ F1-macro ของโมเดลตัดประโยคภาษาไทย เมื่อเทียบด้วยเทคนิคการใช้โมเดลทางภาษาระหว่าง CBOW และ SG โดยทำการเปรียบเทียบจากโมเดล Bi-LSTM-CNN-CRF ซึ่งทำการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS

Dataset	Word	Word Char + POS	% ที่ดีขึ้น
	Char		
CBOW_mozilla_common_voice	75.668	73.843	-2.412
SG_mozilla_common_voice	73.666	74.94	1.729
CBOW_nus_sms	69.148	67.349	-2.602
SG_task_master_1	68.964	67.733	-1.785
CBOW_task_master_1	68.735	63.311	-7.891
CBOW_Orchid	66.974	68.621	2.459
SG_nus_sms	65.03	69.212	6.431
CBOW_generated_reviews_yn	64.431	66.428	3.099
CBOW_generated_reviews_translator	64.168	66.58	3.759
SG_generated_reviews_yn	62.801	67.45	7.403
CBOW_generated_reviews_crowd	62.776	65.699	4.656
SG_generated_reviews_translator	62.648	66.902	6.790
SG_msr_paraphrase	62.121	65.936	6.141
SG_Orchid	62.005	66.159	6.699
SG_paracrawl	61.231	61.451	0.359
SG_generated_reviews_crowd	61.067	65.247	6.845
CBOW_msr_paraphrase	60.6	65.051	7.345
CBOW_paracrawl	60.3	60.057	-0.403
CBOW_wikipedia	51.491	56.406	9.545
SG_wikipedia	51.127	56.218	9.958
SG_apdf	50.561	53.523	5.858
CBOW_apdf	50.489	54.262	7.473
CBOW_thai_websites	50.231	52.45	4.418
CBOW_assorted_government	50.156	53.699	7.064
SG_thai_websites	49.068	52.812	7.630
SG_assorted_government	48.919	53.718	9.810

สรุปจากผลการทดลอง การนำโมเดลตัดประโยคภาษาไทยที่ถูกสร้างขึ้นจากชุดข้อมูล LST20 นำไปใช้กับชุดข้อมูลที่ไม่เคยเรียนรู้มาก่อนผ่านกลุ่มฟีเจอร์ 2 กลุ่ม นั่นคือกลุ่มของระดับคำและตัวอักษร และกลุ่มของระดับคำ ตัวอักษร และ POS พบว่า มีส่วนช่วยให้การตัดประโยคภาษาไทยมีประสิทธิภาพมากขึ้นเมื่อเทียบกับการใช้ข้อมูลฟีเจอร์เพียงระดับคำ และตัวอักษร โดยที่โมเดลที่ถูกสร้างขึ้นด้วยเทคนิคของ Skip-gram มีแนวโน้มที่ให้ประสิทธิภาพการตัดประโยคที่ดีกว่าการใช้เทคนิค CBOW

5.2 อภิปรายผลการศึกษา

จากการศึกษา และทดลองสร้างโมเดลตัดประโยคภาษาไทยโดยใช้เทคนิคการเรียนรู้เชิงลึก ร่วมกับแบบจำลองโมเดลคอนดิชันนอลแรนดอมฟิลด์ส จนเกิดเป็นโมเดล บอยด์คัท หรือเรียกว่า Bi-LSTM-CNN-CRF (Bi-Directional Convolutional Neural Network with Conditional Random Field) ทางผู้วิจัยได้ทำการสรุปหัวข้อที่น่าสนใจไว้ดังนี้

5.2.1 จากการเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย ระหว่างแบบจำลองคอนดิชันนอลแรนดอมฟิลด์ส และสถาปัตยกรรมโครงข่ายลองชอตเทอมเมมโมรี่ชนิดสองทาง พบว่าประโมเดลตัดประโยคภาษาไทยที่ออกแบบด้วยเทคนิคการเรียนรู้เชิงลึก มีประสิทธิภาพ F1-macro ที่สูงกว่าโมเดลพื้นฐานที่ใช้ แบบจำลองโมเดลคอนดิชันนอลแรนดอมฟิลด์ส ซึ่งโมเดล Bi-LSTM-CNN-CRF ที่ใช้ Skip-gram และ CBOW เทคนิคมีค่า เฉลี่ย F1-macro สูงกว่า แบบจำลองโมเดลคอนดิชันนอลแรนดอมฟิลด์ส 4.961 หน่วยและ 4.901 หน่วยตามลำดับ หรือดีขึ้นจากเดิมคิดเป็นร้อยละ 5.51 และ 5.44 ตามลำดับ ดังนั้นในการเริ่มทำโมเดลตัดประโยคภาษาไทย การประยุกต์ใช้ หรือออกแบบสถาปัจยกรรมโมเดลโดยมีพื้นฐานจากเทคนิคการเรียนรู้เชิงลึกสามารถสร้างประสิทธิภาพโมเดลที่สูงได้มากกว่าการใช้เทคนิคการใช้โมเดล คอนดิชันนอลแรนดอมฟิลด์ส เพียงอย่างเดียว

5.2.2 จากการเปรียบเทียบการใช้ฟีเจอร์จากการทำโมเดลทางภาษา (Language Model) ผ่านวิธีของ Mikolov อันได้แก่ คอนตินิวอัสแบกออฟเวิด (Continuous Bag-of-Words Model, CBOW) และ สคริปแกรม (Continuous Skip-Gram Model) พบว่าในการใช้กลุ่มของฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs ร่วมกับการใช้ฟีเจอร์โมเดลทางภาษาด้วยเทคนิค Skip-gram ให้ประสิทธิภาพการตัดคำที่ดีที่สุด เมื่อเปรียบเทียบกับทุกกลุ่มการทดลอง และเมื่อทำการลดฟีเจอร์ที่ใช้ในการสอนโมเดลลงมาเหลือเพียง ระดับคำ ระดับตัวอักษร และ POS เทคนิคของ Skip-gram ยังคงมีประสิทธิภาพการตัดประโยคภาษาไทยที่สูงกว่าการใช้เทคนิค CBOW แต่เมื่อเปลี่ยนกลุ่มของฟีเจอร์ลงมาเหลือเพียงกลุ่มของ ระดับคำ ระดับตัวอักษร และ NER กับอีกกลุ่มหนึ่งคือ ระดับคำ

ระดับตัวอักษร พบว่าการใช้เทคนิค CBOW ส่งผลให้ประสิทธิภาพการตัดประโยคภาษาไทยที่สูงกว่าการใช้เทคนิค Skip-gram ดังนั้นหากต้องการสร้างโมเดลตัดประโยคภาษาไทยโดยใช้ฟีเจอร์เพียงระดับคำ และตัวอักษร ให้เลือกใช้ CBOW ในการสร้างโมเดลแทน แต่หากผู้สร้างโมเดลต้องการใช้ฟีเจอร์ทั้งในระดับคำ ระดับตัวอักษร POS และ NEs ร่วมกันสร้างโมเดลตัดประโยคภาษาไทย การเลือก Skip-gram เทคนิคมาใช้ จะส่งผลให้โมเดลมีประสิทธิภาพการตัดประโยคที่แม่นยำกว่าเทคนิค CBOW

จากผลการทดลอง เมื่อพิจารณาประสิทธิภาพการตัดประโยคภาษาทไทยที่สร้างด้วยเทคนิค Bi-LSTM-CNN-CRF จะเห็นได้ว่าการเพิ่มฟีเจอร์ระหว่าง POS และ NEs นั้นการเพิ่มฟีเจอร์ POS ส่งผลให้ประสิทธิภาพของโมเดลสูงกว่าการเพิ่ม NEs ถึงร้อยละ 3.289 และ 2.942 เมื่อเปรียบเทียบระหว่างสองเทคนิค Skip-gram และ CBOW ตามลำดับ ซึ่งแสดงให้เห็นว่า หากผู้ใช้งานต้องการสร้างโมเดลตัดประโยคภาษาไทยด้วยตัวเองนั้น การใช้ฟีเจอร์ที่นอกเหนือจากระดับคำ และระดับตัวอักษร คือการเพิ่มฟีเจอร์ POS เข้าไปช่วย ขณะเดียวกันหากทำการเพิ่ม NEs เข้าไปแทนฟีเจอร์ POS ประสิทธิภาพของโมเดล ยังคงสามารถเพิ่มประสิทธิภาพของโมเดลได้สูงกว่าการใช้ฟีเจอร์ในกลุ่มของระดับคำ และตัวอักษร เพียงอย่างเดียวร้อยละ 0.159 และ 0.223 เมื่อเปรียบเทียบระหว่างสองเทคนิค Skip-gram และ CBOW ตามลำดับ ดังนั้นหากต้องการเพิ่มประสิทธิภาพของโมเดลตัดประโยคภาษาไทย การใส่ฟีเจอร์ POS เพิ่มก็เพียงพอ และเป็นตัวเลือกที่เหมาะสมกว่าการเพิ่มฟีเจอร์ NEs

5.2.3 จากการเปรียบเทียบการใช้ฟีเจอร์กลุ่มต่างๆ เพื่อเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย พบว่ากลุ่มของฟีเจอร์ที่ส่งผลให้ประสิทธิภาพการตัดประโยคภาษาไทย Bi-LSTM-CNN-CRF มีประสิทธิภาพมากที่สุดคือการใช้ฟีเจอร์ระดับคำ ระดับตัวอักษร POS และ NEs ร่วมกับการใช้โมเดลทางภาษาด้วยเทคนิค Skip-gram หากใช้กลุ่มของฟีเจอร์ระดับคำ ระดับตัวอักษร และ POS การเลือกใช้โมเดลทางภาษาด้วยเทคนิค Skip-gram ให้ประสิทธิภาพการตัดประโยคภาษาไทยที่สูงกว่า CBOW เช่นเดียวกัน ในขณะเดียวกันหากเปลี่ยนมาใช้ฟีเจอร์ในกลุ่มของ ระดับคำ ระดับตัวอักษร NEs หรือ กลุ่มของ ระดับคำ ระดับตัวอักษร การเลือกใช้โมเดลทางภาษาด้วยเทคนิค CBOW ส่งผลให้ประสิทธิภาพของโมเดลสูงกว่าการเลือกใช้เทคนิค Skip-gram ทั้งนี้การเพิ่มการใช้ฟีเจอร์ POS และ NEs เข้าร่วมกับฟีเจอร์ระดับของคำ และตัวอักษร ส่งผลให้ประสิทธิภาพของโมเดลเพิ่มขึ้นจากการใช้ฟีเจอร์เพียงระดับคำ และตัวอักษรร้อยละ 3.598 และ 3.083 เมื่อเทียบระหว่างเทคนิค Skip-gram และ CBOW ตามลำดับ

และเมื่อพิจารณา เปรียบเทียบประสิทธิภาพของโมเดล Bi-LSTM-CNN-CRF ที่ใช้เพียงฟีเจอร์ระดับคำ และตัวอักษร เทียบกับโมเดล แบบจำลองโมเดลคอนดิชันนอลแรนดอมฟิลด์ส ที่ใช้

ฟีเจอร์ทั้งในระดับคำ ระดับตัวอักษร POS และ NEs มีประสิทธิภาพที่สูงกว่าถึงร้อยละ 1.844 และ 2.888 เมื่อเทียบระหว่างเทคนิค Skip-gram และ CBOW ตามลำดับ และเมื่อทำการเปรียบเทียบกับโมเดลแบบจำลองโมเดลคอนดิชันนอลแรนดอมฟิลด์ส ที่ได้รับการปรับปรุง พบว่า ประสิทธิภาพของโมเดล Bi-LSTM-CNN-CRF ที่ใช้เพียงฟีเจอร์ระดับคำ และตัวอักษร ยังคงมีประสิทธิภาพที่สูงกว่าถึงร้อยละ 1.783 และ 2.227 เมื่อเทียบระหว่างเทคนิค Skip-gram และ CBOW ตามลำดับ

5.2.4 จากการเปรียบเทียบประสิทธิภาพของโมเดลตัดประโยคภาษาไทย โดยนำโมเดลที่ไปทดสอบประสิทธิภาพกับชุดข้อมูลที่ไม่ได้ทำการเรียนรู้มาก่อน ด้วยชุดข้อมูล ORCHID และ scb-mt-en-th-2020 พบว่าการเพิ่ม POS ฟีเจอร์ลงไปในกลุ่มฟีเจอร์ระดับคำ และตัวอักษรช่วยเพิ่มประสิทธิภาพการตัดประโยคภาษาไทย ภึงแม้ว่า ชุดข้อมูลเหล่านั้นจะไม่เคยถูกเรียนรู้มาก่อนเลย โดยที่จาก 26 ชุดข้อมูลที่ทำการทดลองโดยแบ่งเป็นทดสอบด้วยโมเดลที่ถูกสร้างขึ้นจากเทิคนิค CBOW จำนวน 13 ชุดข้อมูล และ Skip-gram อีก 13 ชุดข้อมูล พบว่าการเพิ่ม POS ฟีเจอร์ส่งผลให้ประสิทธิภาพการตัดประโยคเพิ่มขึ้น 9 จาก 13 ชุดข้อมูลใน CBOW เทคนิค และ 12 จาก 13 ชุดข้อมูลในเทคนิค Skip-gram โดยที่การใช้โมเดลที่สร้างขึ้นจากเทคนิค Skip-gram ส่งผลต่อการตัดประโยคที่ดีกว่าในมุมของการเพิ่มฟีเจอร์ด้วย POS

หากผู้ใช้งานต้องการสร้างโมเดลตัดประโยคภาษาไทย ผู้วิจัยแนะนำให้สร้างขึ้นจากโมเดล Bi-LSTM-CNN-CRF ที่ใช้เทคนิคของ Skip-gram ด้วยใช้ฟีเจอร์ที่มาจากทั้งระดับคำ ตัวอักษร POS และ NEs ซึ่งในปัจจุบันการสกัดฟีเจอร์ POS นั้นสามารถทำได้โดยใช้โมเดลจากทาง PythaiNLP ซึ่งช่วยสกัด POS ออกมา แต่ไม่สามารถทราบถึงประสิทธิภาพความถูกต้องของโมเดลในการสกัด POS นอกจากนี้ยังไม่มีโมเดลที่ใช้ในการสกัด NEs ที่มีจากชุดข้อมูล LST20 ดังนั้นทางผู้วิจัยจึงแนะนำให้ใช้เพียงฟีเจอร์ระดับคำ ตัวอักษร และ POS ก็เพียงพอต่อการสร้างโมเดลตัดประโยคภาษาไทยที่มีประสิทธิภาพแล้ว

5.3 ข้อเสนอแนะ

5.3.1 เพิ่มการใช้เทคนิคการทำโมเดลด้วยเทคนิคอื่นๆ ที่ช่วยให้ประสิทธิภาพการตัดประโยคภาษาไทย มีความแม่นยำมากขึ้น

5.3.2 เพิ่มการใช้เทคนิคการทำโมเดลภาษาในรูปแบบใหม่ๆ ที่ช่วยให้ประสิทธิภาพการตัดประโยคภาษาไทย มีความแม่นยำมากขึ้น

5.3.3 เพิ่มการปรับปรุงสถาปัตยกรรมของโมเดลให้มีความหลากหลายมากขึ้น เช่นการเพิ่มจำนวนชั้นของเลเยอร์โมเดล

5.3.4 เพิ่มฟีเจอร์ที่สามารถสกัดได้จากระดับคำ และในระดับตัวอักษร หรือฟีเจอร์อื่นๆที่เกี่ยวข้อง โดยที่ไม่จำเป็นต้องสร้างโมเดลอื่นๆ มาทำการสกัดฟีเจอร์เพิ่ม

5.3.5 การนำโมเดลตัดประโยคภาษาไทยไปใช้งานในระดับแอพพลิเคชั่นของการประมวลผลภาษาธรรมชาติ เช่นการวิเคราะห์ความรู้สึก ในระดับเอกสาร (Document Sentiment Analysis) หรือการต่อยอดทำการสรุปบทความ (Text Summarization) รวมถึงการสร้างองค์ความรู้เชิงกราฟ (Knowledge Graph)

บรรณานุกรม

บรรณานุกรม

ภาษาต่างประเทศ

R. Kittinaradorn, T. Achakulvisut, K. Chaovavanich, K. Srithaworn, P. Chormai, C. Kaewkasi, T. Ruangrong and K. Oparad, "DeepCut: A Thai word tokenization library using Deep Neural Network," 2019.

L. Sertis Co., "Thai word segmentation with bi-directional RNN," 2017.

P. Chormai, P. Prasertsom and A. T. Rutherford, "AttaCut: A Fast and Accurate Neural Thai Word Segmenter," in NeurIPS, 2019.

J. P. Chiu and E. Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs," in Transactions of the Association for Computational Linguistics, 2015.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural Architectures for Named Entity Recognition," in NAACL, 2016.

B. Y. Lin, F. Xu, Z. Luo and K. Zhu, "Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media," in Proceedings of the 3rd Workshop on Noisy User-generated Text, 2017.

X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in Annual Meeting of the Association for Computational Linguistics, 2016.

S. Sirirattanajakarin and B. Suntisrivaraporn, "Annotation Intent Identification toward Enhancement of Marketing Campaign Performance," in International Conference on Knowledge and Systems Engineering, Danang, 2019.

N. Tongtep and T. Theeramunkong, “Multi-stage automatic NE and pos annotation using pattern-based and statistical-based techniques for thai corpus construction,” IEICE Transactions, vol.96-D, no.10, pp.2245–2256, 2013.

N Tongtep, F Coenen and T Theeramunkong. Content-based readability assessment: A study using a syllabic alphabetic language (in Thai). In: Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence. Gold Coast, Australia, 2014, p. 863-70.

N Tongtep, F Coenen and T Theeramunkong. “Discovery of predicate-oriented relations among named entities extracted from thai texts, IEICE Trans. Inf. & Syst., vol. E95-D, no.7, pp.1932-1946, July 2012.

Aroonmanakun, Wirote (2007). “Thoughts on Word and Sentence Segmentation in Thai”. In: Proceedings of the SNLP2007–Symposium on Natural Language Processing, pp. 85–90.

Thairatananond, Y. 1981. Towards the Design of a Thai Text Syllable Analyzer. Master thesis, Asian Institute of Technology.

Poowarawan, Y. 1986. Dictionary-based Thai Syllable Separation, In Proceeding of the Ninth Electronics Engineering Conference.

Sornlertlamvanich, V. 1993. Word Segmentation for Thai in a Machine Translation System NECTEC. (in Thai).

Asanee Kawtrakul, Supapas Kumtanode, Thitima Jamjanya, and Chanvit Jewriyavech. 1995. A Lexibase Model for Writing Production Assistant System. In Proceedings of the Symposium on Natural Language Processing in Thailand '95.

Meknavin, S., Charoenpornsawat, P. and Kijsirikul, B. (1997) Feature-based Thai Word Segmentation. Proceeding of the Natural Language Processing Pacific Rim Symposium 1997, pp. 35-46

D.D. Palmer, Tokenisation and sentence segmentation, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, Marcel Dekker, New York, 2000, pp. 11–36.

Longchupole, Sungkornsarun, 1995, Thai Syntactical Analysis system by method of splitting sentences from paragraph for machine translation, Master Thesis, King Mongkut's Institute of technology Ladkrabang (in Thai)

Riley, M. D. (1989). Some applications of tree-based modelling to speech and language. In Proceedings of the DARPA Speech and Natural Language Workshop, page 339 – 352.

Palmer, David D. and Marti A. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. Computational Linguistics, 23(2):241–67.

Mittrapiyanuruk, P. and V. Sornlertlamvanich. 2000. The Automatic Thai Sentence Extraction. The Fourth Symposium on Natural Language Processing (SNLP2000), Chiang Mai, Thailand, pp 23-28.

J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," 2018, Xiv:1812.09449. [Online]. Available: http://arxiv.org/abs/1812.09449

N. Tongtep and T. Theeramunkong, "Multi-stage automatic NE and pos annotation using pattern-based and statistical-based techniques for thai corpus construction," IEICE Transactions, vol.96-D, no.10, pp.2245–2256, 2013.

Tirasaroj, N. and W. Aroonmanakun. 2009. Thai Named Entity Recognition Based on Conditional Random Fields. Proceedings of the Eighth International Symposium on Natural Language Processing. Bangkok.

W. S. McCulloch, W. A. Pitts, Bull. Math. Biophys. 5, 115 (1943).

D. O. Hebb, The Organization of Behavior, Wiley, New York (1949).

Minsky, M., and Papert, S. Perceptrons. Cambridge, MA: MIT Press (1969).

David E. Rumelhart, Geoffrey E. Hinton, and R. J. Williams 1985, "Learning internal representations by error propagation", In Rumelhart et al.

Rosenblatt, F. The Perceptron — A Perceiving and Recognizing Automaton. Tech. Rep. 85-460-1 (Cornell Aeronautical Laboratory, 1957).

Fukushima, K. & Miyake, S. Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognition 15, 455–469 (1982).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998a). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 2278–2324.

Hopfield JJ. 1982. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79:2554–58.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Jin, Y.; Xie, J.; Guo, W.; Luo, C.; Wu, D.; Wang, R. LSTM-CRF Neural Network with Gated Self Attention for Chinese NER. IEEE Access 2019, 7, 136694–136709.

G. D. Forney, "The Viterbi algorithm," Proc. IEEE, vol. 61, pp. 268–278, Mar. 1973.

Yoshua Bengio, Jean Ducharme, Pascal Vincent, and Christian Janvin. March 2003. A neural probabilistic language model,

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernock ´ y, and Sanjeev Khudanpur. 2010. Recurrent neural ` network based language model. In INTERSPEECH, pages 1045–1048.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013c. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย สรทรรศน์ ศิริรัตนจักริน
ประวัติการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมข้อมูลขนาดใหญ่ มหาวิทยาลัยธุรกิจบัณฑิตย์ ปีการศึกษา 2561
ตำแหน่งและสถานที่ทำงานปัจจุบัน	นักวิทยาศาสตร์ข้อมูล, Freshket

ผลงานทางวิชาการ

- Movie Genre in Multi-label Classification Using Semantic Extraction from Only Movie Poster **S. Sirattanajakarin**, P. Thusaranon Jul 7, 2019 Proceedings of the 2019 7th International Conference on Computer and Communications Management

- Annotation Intent Identification toward Enhancement of Marketing Campaign Performance **S. Sirirattanajakarin**, B. Suntisrivaraporn Oct 10, 2019 11th International Conference on Knowledge and Systems Engineering (KSE)

- Improved Identification of Imbalanced Multiple Annotation Intent Labels with a Hybrid BLSTM and CNN model and Hybrid Loss Function

S. Vatathanavaro, K. Pasupa, **S. Sirirattanajakarin**, and B. Suntisrivaraporn 2020 The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)

- BoydCut: Bidirectional LSTM-CNN Model for Thai Sentence Segmenter

S. Sirirattanajakarin, D. Jitkongchuen, P. Intarapaiboon 2020 1st International Conference on Big Data Analytics and Practices (IBDAP)