# การฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นในการจัดประเภทข้อความ

นายธีรพงศ์ แซ่ลิ้ม

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาสถิติ ภาควิชาสถิติ
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

# VIRTUAL ADVERSARIAL TRAINING WITH WEIGHTED TOKEN PERTURBATION IN TEXT CLASSIFICATION

Mr. Teerapong Sae-lim

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Statistics

Department of Statistics

FACULTY OF COMMERCE AND ACCOUNTANCY

Chulalongkorn University

Academic Year 2021

Copyright of Chulalongkorn University

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นในการจัดประเภทข้อความ |
| โดย | นายธีรพงศ์ แซ่ลิ้ม |
| สาขาวิชา | สถิติ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | ผู้ช่วยศาสตราจารย์ ดร.สุรณพีร์ ภูมิวุฒิสาร |

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะพาณิชยศาสตร์และการบัญชี

(รองศาสตราจารย์ ดร.วิเลิศ ภูริวัชร)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.วธนน์ วิริยสิทธาวัฒน์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.สุรณพีร์ ภูมิวุฒิสาร)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ภูริพันธุ์ รุจิขจร)

..... กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร.ชาลี ธรรมรัตน์)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ธีรพงศ์ แซ่ลิ้ม : การฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นในการจัดประเภทข้อความ. ( VIRTUAL ADVERSARIAL TRAINING WITH WEIGHTED TOKEN PERTURBATION IN TEXT CLASSIFICATION) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุรณพีร์ ภูมิวุฒิสาร

การจัดประเภทข้อความ (Text classification) เป็นกระบวนการคัดแยกข้อความให้เป็นหมวดหมู่อย่างถูกต้อง ตัวแบบจำลองการฝึกอบรมล่วงหน้าโดยใช้ตัวเข้ารหัสแบบสองทิศจากทรานฟอร์เมอร์ หรือเรียกว่า BERT ช่วยทำให้ตัวแบบจำลองเรียนรู้บริบทของคำแบบสองทิศทาง ส่งผลให้สามารถจัดประเภทข้อความได้อย่างมีประสิทธิภาพและแม่นยำ ถึงแม้ว่าตัวแบบจำลอง BERT และตัวแบบจำลองที่เกิดขึ้นจากสถาปัตยกรรมนี้ จะสามารถจัดการงานด้านการประมวลผลทางธรรมชาติได้อย่างยอดเยี่ยม แต่กลับพบว่าตัวแบบจำลองนี้ยังพบเจอปัญหา Overfitting กล่าวคือ เมื่ออยู่ในสถานการณ์ที่ชุดข้อมูลในการฝึกอบรมมีจำนวนตัวอย่างน้อย ตัวแบบจำลอง BERT จะให้ความสนใจไปที่คำบางคำมากเกินไปจนไม่สนใจบริบทของประโยค จนทำให้ตัวแบบจำลองไม่สามารถทำนายข้อมูลในชุดการทดสอบได้ถูกต้อง ซึ่งส่งผลในประสิทธิของตัวแบบจำลองลดลง ดังนั้นในงานวิทยานิพนธ์ฉบับนี้จึงเสนอแนวทาง วิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น ซึ่งรวมการรบกวนสองระดับเข้าด้วยกัน ได้แก่ การรบกวนระดับประโยค และการรบกวนแบบถ่วงน้ำหนักโทเค็น เพื่อสร้างการรบกวนที่มีความละเอียดกว่าการฝึกปรปักษ์เสมือนแบบดั้งเดิม ที่อาศัยเพียงการรบกวนระดับประโยคเท่านั้น วิธีการนี้จะช่วยให้ตัวแบบจำลองสามารถเรียนรู้โทเค็นที่สำคัญในประโยค จากการทดลองบนเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (GLUE) แสดงให้เห็นว่าวิธีการที่นำเสนอสามารถเพิ่มประสิทธิภาพของตัวแบบจำลองโดยได้คะแนนเฉลี่ยร้อยละ 79.5 ซึ่งมีประสิทธิภาพเหนือกว่าตัวแบบจำลอง BERT และสามารถแก้ไขปัญหา Overfitting ในชุดข้อมูลขนาดเล็ก

สาขาวิชา สถิติ ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

# # 6380157926 : MAJOR STATISTICS

KEYWORD: Natural language processing, Virtual adversarial training, Transformers, TEXT CLASSIFICATION

Teerapong Sae-lim : VIRTUAL ADVERSARIAL TRAINING WITH WEIGHTED TOKEN PERTURBATION IN TEXT CLASSIFICATION. Advisor: Asst. Prof. SURONAPEE PHOOMVUTHISARN, Ph.D.

Text Classification is the process of classifying text into categories. Among its contextualized architecture proposed, pretraining Bidirectional Encoder Representations from Transformers (BERT) helps models learn the bidirectional context of words, making it possible to classify text much more efficiently and accurately. Although BERT and its variance have led to impressive gains on many natural language processing (NLP) tasks, one of the problems of BERT is the overfitting problem. When training data is limited, BERT model overemphasizes certain words and ignores the context of the sentence. This makes it difficult for the model to make accurate predictions on the test data. We propose virtual adversarial training with the weighted token perturbation, which combines two-level perturbations: (1) sentence-level perturbation and (2) the weighted token perturbation to create a more granular perturbation than traditional virtual adversarial training with only sentence-level perturbation. Our approach can help models learn more about the key and important tokens in sentences when trained with virtual adversarial examples. The experiments in the General Language Understanding Evaluation (GLUE) benchmark showed that our approach can achieve the average score of 79.5%, which outperforms $\text{BERT}_{\text{base}}$ model and reduce the overfitting problem on small datasets.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Field of Study: Statistics Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้รับความช่วยเหลือและคำแนะนำจาก ผศ.ดร.สุรณพีร์ ภูมิวุฒิสาร อาจารย์ที่ปรึกษาวิทยานิพนธ์ จึงสามารถสำเร็จลุล่วงไปได้ด้วยดี ผู้ทำวิจัยขอกราบขอบพระคุณอาจารย์ที่คอยช่วยเหลือ สอนสั่งและแนะนำตั้งแต่เริ่มทำวิทยานิพนธ์จนสำเร็จลุล่วง อีกทั้งยังค่อยเอาใจใส่ ติดตามและให้เวลาอันมีค่าแก่ผู้วิจัยทุกครั้ง ในวันที่ผู้ทำวิจัยท้อแท้กับงาน อาจารย์ยังค่อยเป็นกำลังใจและดูแลเป็นอย่างดี สุดท้ายขอขอบพระคุณอาจารย์อย่างใจจริงและดีใจที่ได้มาเป็นผู้วิจัยภายใต้การดูแลของอาจารย์

ผู้วิจัยขอกราบขอบพระคุณ รองศาสตรจารย์ ดร.วธนน์ วิริยสิทธาวัฒน์ ประธานกรรมการ ผู้ช่วยศาสตราจารย์ ดร.ภูริพันธุ์ รุจิขจร และ อาจารย์ ดร.ชาลี ธรรมรัตน์ กรรมการสอบวิทยานิพนธ์ เป็นอย่างสูงที่สละเวลาอันมีค่าเพื่อให้คำแนะนำ ตรวจสอบและแก้ไขงานวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วง และขอกราบขอบพระคุณอาจารย์ทุกท่านจากภาควิชาสถิติ คณะพานิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ที่ให้โอกาสแก่ผู้วิจัยได้เข้ารับการศึกษาในภาควิชานี้และค่อยให้ความรู้ตลอดมา

สุดท้ายขอขอบพระคุณบิดา มารดา และครอบครัวที่ค่อยสนับสนุนและส่งเสริมผู้วิจัยมาตลอดตั้งแต่เริ่มต้นเรียนปริญญาโทจนถึงการทำวิจัยในช่วงสุดท้าย ขอบคุณที่ค่อยให้กำลังใจและให้อิสระในการตัดสินใจเลือกเรียนในสิ่งที่ผู้วิจัยชอบเสมอ

ธีรพงศ์ แซ่ลิ้ม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# สารบัญ

**หน้า**




จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# สารบัญตาราง

**หน้าที่**



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# สารบัญภาพ

**หน้าที่**



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

การจัดประเภทข้อความ (Text Classification) เป็นสาขาหนึ่งของการประมวลผลภาษาธรรมชาติ (Natural Language Processing) ที่อาศัยความสัมพันธ์เชิงบริบทระหว่างคำหรืออักขระเพื่อจัดประเภทข้อความเป็นหมวดหมู่ ในการเอาชนะปัญหาการขาดแคลนข้อมูลของงานด้านภาษา จึงได้มีการนำเสนอตัวแบบจำลองที่อาศัยตัวเข้ารหัสแบบสองทิศทางจากสถาปัตยกรรม transformers หรือเรียกว่า BERT (Devlin, Chang, Lee, & Toutanova, 2019) ซึ่งได้รับการออกแบบมาเพื่อการฝึกอบรบล่วงหน้า (Pre-trained) ด้วยข้อมูลจำนวนมากและสามารถนำมาปรับใช้ (Fine-tuning) ให้เข้ากับงานอื่น ๆ ที่มีความคล้ายคลึงกันได้ นอกจากจะได้รับการฝึกอบรมด้วยข้อมูลขนาดใหญ่แล้ว ตัวแบบจำลอง BERT ยังอาศัยการเข้ารหัสแบบสองทิศทางเพื่อให้เข้าใจรูปแบบความสัมพันธ์เชิงลึกอีกด้วย โดยคุณลักษณะนี้ช่วยให้ตัวแบบจำลองเรียนรู้บริบทของคำหนึ่ง ๆ ได้อย่างมีประสิทธิภาพโดยพิจารณาทั้งด้านขวาและซ้ายของรูปแบบประโยค การทำให้ตัวแบบจำลองเห็นบริบทของประโยคทั้งทางซ้ายและทางขวาทำให้การเรียนรู้มีประสิทธิภาพและแม่นยำมากขึ้น แม้ว่าในช่วงไม่กี่ปีที่ผ่านมา ตัวแบบจำลอง BERT ได้มีการปรับปรุงและต่อยอดเป็นตัวแบบจำลองอื่นๆ อีกมากมาย เช่น RoBERT (Y. Liu et al., 2019), DistrilBERT (Sanh, Debut, Chaumond, & Wolf, 2019), และ ALBERT (Lan et al., 2020) ซึ่งตัวแบบจำลองเหล่านี้ประสบความสำเร็จในการทำงานด้านการประมวลผลธรรมชาติ แต่ก็ไม่สามารถบรรลุความแม่นยำระดับสูงสำหรับงานบางอย่างที่มีความซับซ้อนสูงได้ (Dagan, Glickman, & Magnini, 2005), (Dolan & Brockett, 2005), (Warstadt, Singh, & Bowman, 2019) และเมื่อมีข้อมูลการฝึกอบรมที่จำกัดทำให้ตัวแบบจำลอง BERT ที่ถูกฝึกอบรมล่วงหน้าด้วยข้อมูลขนาดใหญ่จะความสนใจไปยังคำบางคำหรือคำที่ทำให้เกิดความเข้าใจผิดมากเกินไปโดยไม่สนใจบริบทรอบข้างของประโยคส่งผลให้เกิดปัญหา Overfitting การที่ตัวแบบจำลองให้ความสนใจไปยังคำที่ไม่เกี่ยวข้องหรือคำที่ทำให้เข้าใจผิดมากเกินไปของตัวแบบจำลองส่งผลให้ความแม่นยำลดลงเมื่อนำตัวแบบจำลองไปปรับใช้ในชุดข้อมูลสำหรับทดสอบ (Test Data)

ปัจจุบันมีหลายวิธีการในการแก้ไขปัญหา Overfitting วิธีการแรกคือวิธีการเสริมข้อมูล (Data Augmentation) โดยมีหลายงานวิจัย (Kobayashi, 2018), (Wei & Zou, 2019), และ (Wu, Lv, Zang, Han, & Hu, 2019) ที่ใช้เทคนิคนี้เพื่อสร้างข้อมูลใหม่เพื่อมาฝึกอบรมตัวแบบจำลอง

อย่างไรก็ตามวิธีการเสริมข้อมูลในงานด้านภาษาศาสตร์มีความท้าทายเนื่องจากการเปลี่ยนคำในประโยคอาจจะส่งผลให้ความหมายหรือบริบทของประโยคเปลี่ยนไปจากประโยคเดิมและความยากในการตรวจสอบความถูกต้องของประโยคที่ถูกสร้างขึ้นใหม่ วิธีการที่สองคือการสร้างงานเสริม (Auxiliary Task) มีหลายงานวิจัยเช่น (Zhou, Li, & Xie, 2021), (Kou et al., 2021), และ (Gunel, Du, Conneau, & Stoyanov, 2021) โดยให้ตัวแบบจำลองเรียนรู้และแก้ไขงานเสริมที่ถูกสร้างขึ้นเพื่อแก้ไขปัญหา Overfitting ในงานจัดประเภทข้อความ ส่งผลให้ตัวแบบจำลองมีการปรับปรุงประสิทธิภาพ อย่างไรก็ดีการสร้างงานเสริมนั้นมีความเฉพาะเจาะจงอย่างมากและใช้เวลาในการสร้างป้ายกำกับ (Label) นอกจากนี้ยังมีอีกแนวทางหนึ่งที่น่าสนใจซึ่งนำมาจากคอมพิวเตอร์วิทัศน์ (Computer Vision) ที่เรียกว่าการฝึกปรปักษ์ (Adversarial Training) (Goodfellow, Shlens, & Szegedy, 2015) วิธีการนี้เพิ่มการรบกวนเล็กน้อย (Small Perturbation) ให้กับข้อมูลเดิมเพื่อเสริมความแข็งแกร่งให้กับตัวแบบจำลอง (Jiang et al., 2020), (X. Liu, Cheng, et al., 2020), (Pereira, Liu, Cheng, Asahara, & Kobayashi, 2020), และ (Zhu et al., 2020) การเพิ่มสิ่งรบกวนให้กับข้อมูลเดิมจะทำให้ตัวแบบจำลองทำนายผลผิดพลาดแต่ตัวแบบจำลองจะสามารถเรียนรู้ข้อมูลอื่นเพิ่มเติมเพื่อให้สามารถทำนายผลให้กลับมาถูกต้องได้ แม้ว่าจะสามารถแก้ไขปัญหา Overfitting แต่อาศัยเพียง Normalization Ball ระดับประโยคเพื่อสร้างการรบกวนให้กับข้อมูลเดิม กล่าวอีกนัยหนึ่งก็คือ Normalization Ball ระดับประโยคซึ่งเป็นข้อจำกัดของ $L_p$ normalization ทำหน้าที่ในการคำนวณค่า Normalization โดยรวมของโทเค็นทั้งประโยค ทำให้ค่า Normalization ของทุกโทเค็นมีค่าเท่ากัน จึงขาดความละเอียดในการรับรู้ความแตกต่างของโทเค็นในประโยค (Li & Qiu, 2021) ด้วยวิธีการนี้อาจจะทำให้ตัวแบบจำลองไม่มีการเรียนรู้โทเค็นที่มีความสำคัญและทำให้การทำนายผลลัพธ์ไม่ถูกต้อง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ในงานวิจัยนี้เสนอการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) เพื่อปรับปรุงประสิทธิภาพของตัวแบบจำลอง BERT วิธีการของงานวิจัยนี้รวมการรบกวนระดับประโยคของฝึกปรปักษ์เสมือนแบบดั้งเดิมและการรบกวนแบบถ่วงน้ำหนักระดับโทเค็นเพื่อสร้างตัวอย่างการฝึกปรปักษ์เสมือนแบบละเอียด การรบกวนแบบถ่วงน้ำหนักโทเค็นใช้ Normalization Ball ระดับโทเค็น ซึ่งทำหน้าที่ในการคำนวณค่า normalization ของโทเค็นแต่ละโทเค็นเพื่อใช้สร้างตัวรบกวนระดับโทเค็น อีกทั้งยังถ่วงน้ำหนักความสำคัญของโทเค็นแต่ละตัวตามผลกระทบของการไล่ระดับ (Gradient) ของ Kullback-Leibler Divergence (Joyce, 2011) โดยสามารถเพิ่มประโยชน์ในการเรียนรู้โทเค็นที่เกี่ยวข้องและ

ลดผลกระทบของโทเค็นที่ไม่เกี่ยวข้องหรือโทเค็นที่ทำให้เข้าใจผิด ตัวแบบจำลองจะเรียนรู้ข้อมูลอื่น ๆ ที่เกี่ยวกับโทเค็นที่สำคัญสำหรับการทำนายประเภทข้อความเมื่อฝึกด้วยตัวอย่างปรปักษ์แบบเสมือน (Virtual Adversarial Examples) ดังนั้นวิธีการของงานวิจัยนี้จึงสามารถแก้ไขปัญหา Overfitting และสามารถปรับปรุงประสิทธิภาพของตัวแบบจำลอง

งานวิจัยนี้ได้รับการประเมินโดยใช้เกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE) ซึ่งจะมีชุดข้อมูลสำหรับการฝึกอบรม การประเมินผลและการวิเคราะห์ความเข้าใจของภาษาธรรมชาติของตัวแบบจำลอง (Wang et al., 2018) จากผลของ GLUE แสดงให้เห็นว่าวิธีการของงานวิจัยนี้สามารถปรับปรุงตัวแบบจำลอง BERT ให้ประสิทธิภาพที่สูงขึ้นและสามารถแก้ไขปัญหา Overfitting บนชุดการทดสอบที่มีข้อมูลอยู่อย่างจำกัด

### 1.2 วัตถุประสงค์

1. นำเสนอแนวทางการแก้ไขปัญหา Overfitting ที่เกิดจากตัวแบบจำลอง Bidirectional Encoder Representations from Transformers (BERT)
2. ศึกษาและเปรียบเทียบประสิทธิภาพในด้านการจัดประเภทข้อความระหว่างวิธีการที่นำเสนอในงานวิจัยกับตัวแบบจำลอง BERT

### 1.3 ขอบเขตของการศึกษา

1. วิธีการของงานวิจัยนี้ถูกออกแบบมาเพื่อตัวแบบจำลอง BERT สำหรับงานจัดประเภทข้อความเท่านั้น
2. วิธีการของงานวิจัยนี้จะถูกวัดผลโดยใช้เกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป ซึ่งใช้ทั้งหมด 8 ชุดข้อมูลในการทดสอบและวัดผล
3. วิธีการของงานวิจัยนี้จะถูกเปรียบเทียบประสิทธิภาพกับตัวแบบจำลอง BERT ดั้งเดิมและตัวแบบจำลอง SMART
4. วิธีการของงานวิจัยนี้ดำเนินการสร้าง ทดลองและวัดผลประสิทธิภาพโดยใช้ภาษา Python

### 1.4 วิธีดำเนินการศึกษา

1. ศึกษาตัวแบบจำลอง Transformer และตัวแบบจำลอง BERT
2. ศึกษางานวิจัยที่เกี่ยวข้องกับวิธีการแก้ไขปัญหา Overfitting บนตัวแบบจำลอง BERT หรือตัวแบบจำลองอื่น ๆ ที่เกี่ยวข้อง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3. วิเคราะห์และสรุปข้อจำกัดในงานวิจัยที่ศึกษาในการแก้ไขปัญหา Overfitting
4. ออกแบบและนำเสนอวิธีการที่สามารถปรับปรุงข้อจำกัดเพื่อแก้ไขปัญหา Overfitting
5. ทำการเปรียบเทียบประสิทธิภาพของวิธีการที่นำเสนอกับตัวแบบจำลอง BERT แบบดั้งเดิม และตัวแบบจำลอง SMART โดยใช้เกณฑ์ GLUE ที่มีทั้งหมด 8 ชุดข้อมูลซึ่งมีตัวชีวัดที่แตกต่างกัน
6. วิเคราะห์ผลจากวิธีการที่นำเสนอและสรุปผลการทดลอง

**1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย**

1. สามารถแก้ไขปัญหา Overfitting ที่เกิดขึ้นกับชุดข้อมูลที่มีจำนวนจำกัดโดยใช้วิธีการที่นำเสนอ
2. สามารถปรับปรุงประสิทธิภาพตัวแบบจำลอง BERT ในงานการจัดประเภทข้อความ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เพื่อจะแก้ไขปัญหา Overfitting และปรับปรุงประสิทธิภาพของตัวแบบจำลอง BERT ผู้วิจัยได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อแก้ไขข้อจำกัดต่าง ๆ ซึ่งทฤษฎีที่เกี่ยวข้องได้แก่ ตัวแบบจำลอง Transformer ตัวแบบจำลอง Bidirectional Encoder Representations จากสถาปัตยกรรม Transformers วิธีการฝึกปรปักษ์ (Adversarial Training) วิธีการฝึกปรปักษ์เสมือน (Virtual Adversarial Training) และวิธีการ Kullback-Leibler Divergence

## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 ตัวแบบจำลอง Transformers

ตัวแบบจำลอง Transformer เป็นสถาปัตยกรรมแบบจำลองที่ไม่ใช้กลไกโครงข่ายประสาทเทียมแบบเกิดซ้ำ (Recurrent Neural Network) แต่จะอาศัยกลไก Attention เพื่อวิเคราะห์ความสัมพันธ์ของคำและบริบทของประโยค อีกทั้งตัวแบบจำลอง Transformers สามารถประมวลผลแบบคู่ขนานกันซึ่งเพิ่มคำเร็วในการฝึกตัวแบบจำลองและสามารถบรรลุประสิทธิภาพในงานการแปลภาษาหลังจากได้รับการฝึกเป็นเวลาเพียง 12 ชั่วโมงบนการ์ดแสดงผลรุ่น Nvidia P100 ทั้งหมด 8 ตัว สถาปัตยกรรมตัวแบบจำลอง Transformers จะมีองค์ประกอบใหญ่ที่สำคัญคือ ตัวเข้ารหัส (encoder) และตัวถอดรหัส (Decoder) ซึ่งทั้งสองส่วนอาศัยหลักการเรียงซ้อนกันของชั้น Self-Attention และชั้นเชื่อมต่ออย่างสมบูรณ์ (Fully-Connected Layer) ในส่วนประกอบของตัวเข้ารหัสจะมีด้วยกันทั้งหมด N = 6 ชั้นเรียงต่อกัน โดยในแต่ละชั้นจะมีสองชั้นย่อย ชั้นย่อยแรกคือ Multi-head self-Attention และชั้นย่อยที่สองคือ Position-wise Fully-Connected Feed-Forward Network สุดท้ายจะมีชั้น Normalization ตามหลัง 2 ชั้นย่อยแรก ผลลัพธ์ของแต่ละชั้นย่อยสามารถเขียนเป็นสมการดังนี้

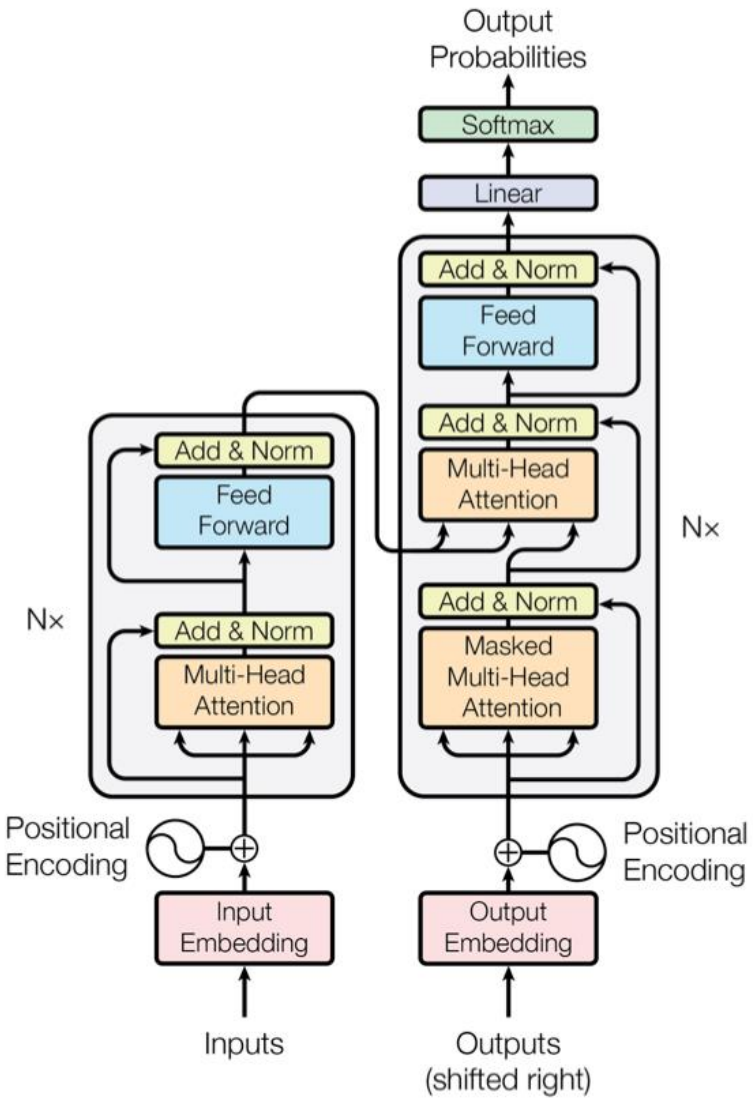
$$LayerNorm(x + Sublayer(x))$$

เมื่อ $Sublayer(x)$ คือ ฟังก์ชันของชั้นย่อย

ในส่วนของตัวถอดรหัสจะมีทั้งหมด N = 6 เรียงต่อกันที่เหมือนกับตัวเข้ารหัสแต่มีการเพิ่มชั้นย่อยที่สามซึ่งเป็นชั้น Multi-Head Attention ต่อเข้ากับตัวเข้ารหัส สุดท้ายต่อเข้ากับชั้น Normalization นอกจากนี้ยังมีการแก้ไขชั้นย่อย Multi-Head Attention ในตัวถอดรหัสเพื่อป้องกันไม่ให้เข้าถึงข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สำหรับทำนายผล ซึ่งทำให้แน่ใจได้ว่าตัวแบบจำลองจะคาดคะเนตำแหน่งที่ $i$ โดยตำแหน่งที่ $i-1$ หรือน้อยกว่า $i$ เท่านั้น โดยแสดงครึ่งซ้ายและครึ่งขวาตามรูปที่ 1




**ภาพที่ 1 แสดงโครงสร้างของ Transformers (ซ้าย) ตัวเข้ารหัส (Encoder) (ขวา) ตัวถอดรหัส (Decoder)**

**(ที่มา: Attention is All You Need)**

ส่วนประกอบย่อยของตัวแบบจำลอง Transformers ที่สำคัญอีกส่วนหนึ่งคือ ฟังก์ชัน Attention ซึ่งทำหน้าที่เป็นการจับคู่คิวรี่ (Query) และชุดของคู่คีย์ (Key) กับค่า (Value) โดยทั้งหมดที่กล่าวมาเป็นเวกเตอร์ทั้งหมด (Vector) ผลลัพธ์จะถูกคำนวณเป็นผลรวมการถ่วงน้ำหนักของค่าทั้งหมด โดยน้ำหนักที่กำหนดให้กับแต่ละค่าจะคำนวณโดยใช้ฟังก์ชันความเข้ากันได้ระหว่างคิวรี่และคีย์ ภายในฟังก์ชัน Attention จะมีกระบวนการพิเศษที่เรียกว่า Scaled Dot-Product Attention ซึ่งจะทำหน้าที่หาความเข้ากันระหว่างคิวรี่และคีย์ โดยอินพุต (Input) สำหรับกระบวนการนี้จะประกอบไปด้วยคิวรี่ คีย์และค่า โดยคิวรี่และคีย์จะมีมิติ $d_k$ และค่าจะมีมิติ $d_v$ จากนั้นกระบวนการนี้จะคำนวณผลคูณเชิงสเกลลาร์ (Dot Product) ระหว่างคิวรี่และคีย์ จากนั้นหารด้วย $\sqrt{d_k}$ สุดท้ายดำเนินการด้วยฟังก์ชัน Softmax เพื่อคำนวณถ่วงน้ำหนักให้กับแต่ละค่า โดยปกติจะมีการคำนวณระหว่างคิวรี่ คีย์และค่า โดยใช้อาศัยหลักการของเมตริก ดังนั้นจึงสามารถเขียนสมการของกระบวนการดังกล่าวได้ดังนี้

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

เมื่อ $Q$ คือ คิวรี่

$K$ คือ คีย์

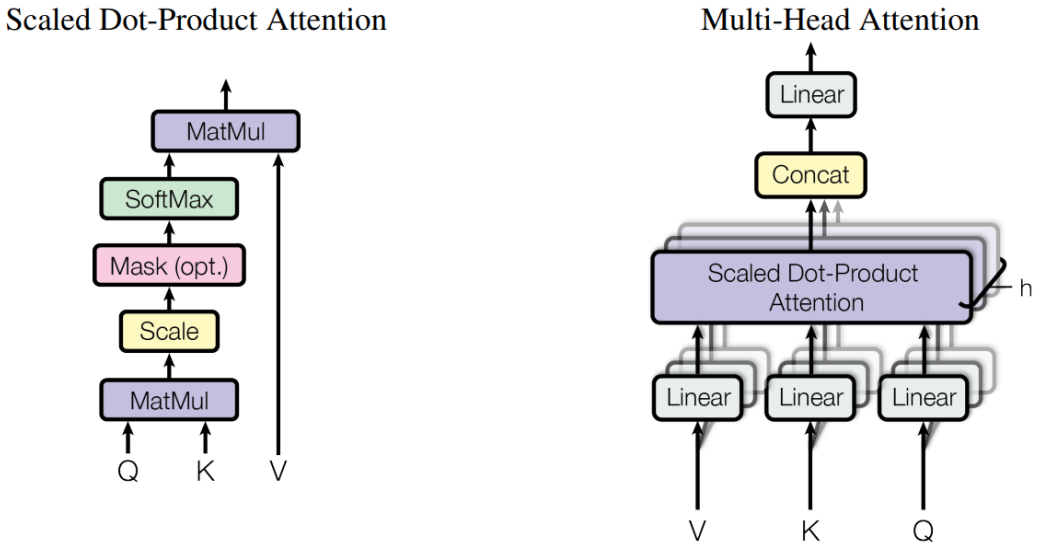
$V$ คือ ค่า

แทนที่จะใช้เพียงการกระบวนการ Attention เพียงอันเดียว ทีมนักวิจัยพบว่าการดำเนินการฟังก์ชัน Attention หลาย ๆ อันจะทำให้ตัวแบบจำลองสามารถมีการเรียนรู้ที่แตกต่างกันซึ่งนั้นเป็นประโยชน์ต่อการหาความสัมพันธ์ของคีย์และคิวรี่ ดังนั้นจึงพัฒนาโดยการรวม Attention เข้าด้วยกันเป็น Multi-Head Attention เพื่อจะทำหน้าที่รับและประมวลผลข้อมูลที่แตกต่างกันได้ สามารถเขียนเป็นสมการได้ดังนี้

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

เมื่อ $head_i = attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$

โดยรายละเอียดของ Scaled Dot-Product Attention และ Multi-Head Attention สามารถตามดูได้จากรูปที่ 2




**ภาพที่ 2 (ซ้าย) Scaled Dot-Product Attention (ขวา) Multi-Head Attention (ที่มา: Attention is All You Need)**

นอกจากกระบวนการ Self-Attention แล้ว แต่ละชั้นในตัวเข้ารหัสและตัวถอดรหัสยังตัวส่วนประกอบสำคัญนั้นคือ Position-Wise Feed-Forward Networks โดยที่ทำหน้าที่คำนวณผลลัพธ์ในแต่ละตำแหน่งแยกจากกัน ซึ่งประกอบไปด้วยสองการแปลงเชิงเส้น (Linear Transformation) โดยใช้ฟังก์ชัน ReLU เขียนเป็นสมการได้ดังนี้

$$FFN(x) = \max(0, xW_1 + b_1)\,W_2 + b_2$$

แม้ว่าสมการแปลงเชิงเส้นจะดูคล้ายกันในทุกตำแหน่งแต่พารามิเตอร์ที่ใช้ในการคำนวณจะแตกต่างกันในแต่ชั้นการคำนวณหาความสัมพันธ์หรือผลลัพธ์ต่าง ๆ

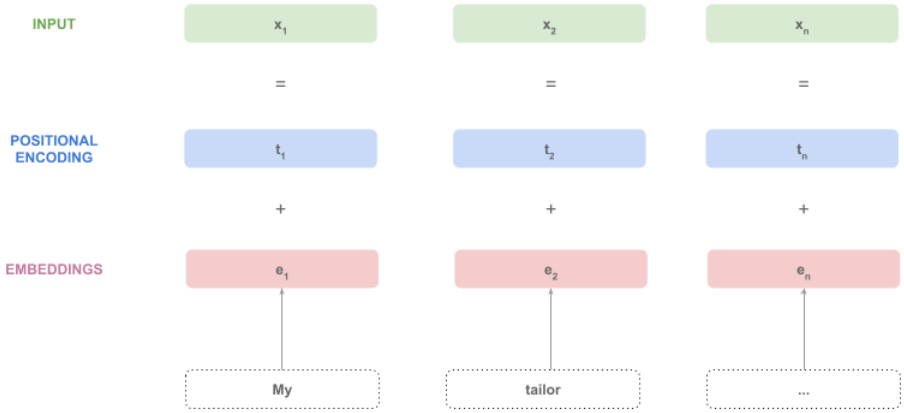
เนื่องจากตัวแบบจำลอง Transformers มีการนำเข้าข้อมูลพร้อมกันไม่เหมือนสถาปัตยกรรมอื่น ๆ ดังนั้นการระบุตำแหน่งและลำดับของอินพุต เพื่อจะให้ตัวแบบจำลองสามารถเข้าลำดับและตำแหน่ง ทีมนักวิจัยใช้กระบวนการ Positional Encoding รวมเข้ากับ Input Embedding เพื่อสร้างเป็นอินพุตสำหรับตัวแบบจำลอง Transformers ดังรูปที่ 3 สำหรับระบุตำแหน่งของโทเค็นในประโยคโดยใช้ฟังก์ชันโคไซน์ (Cosine Function) อาศัยความถี่ที่แตกต่างกันของโคไซน์ในการระบุตำแหน่งตามรูปที่ 4 โดยถ้ายิ่งมีความถี่มากจะทำให้การระบุตำแหน่งละเอียดมากขึ้น โดยมีสมการดังนี้

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

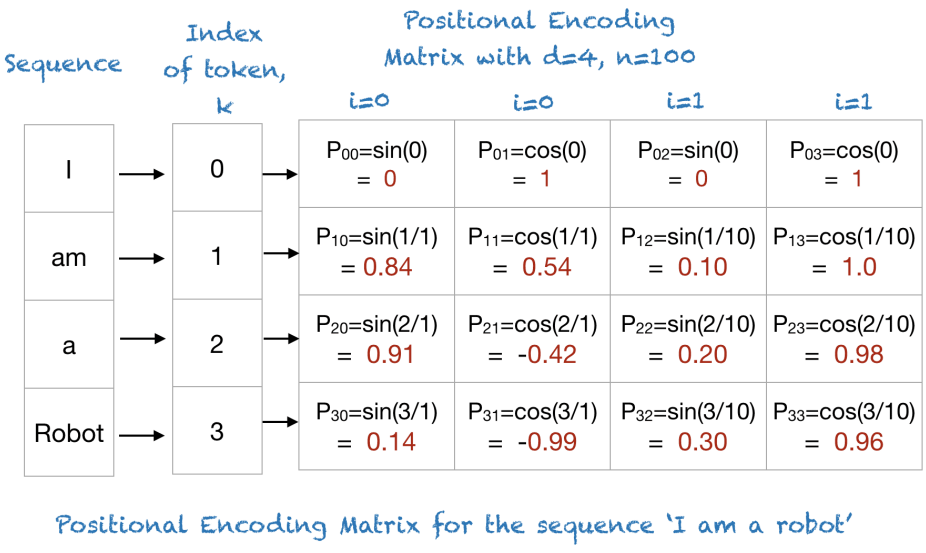
$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

เมื่อ $pos$ คือ ตำแหน่ง

$i$ คือ มิติ



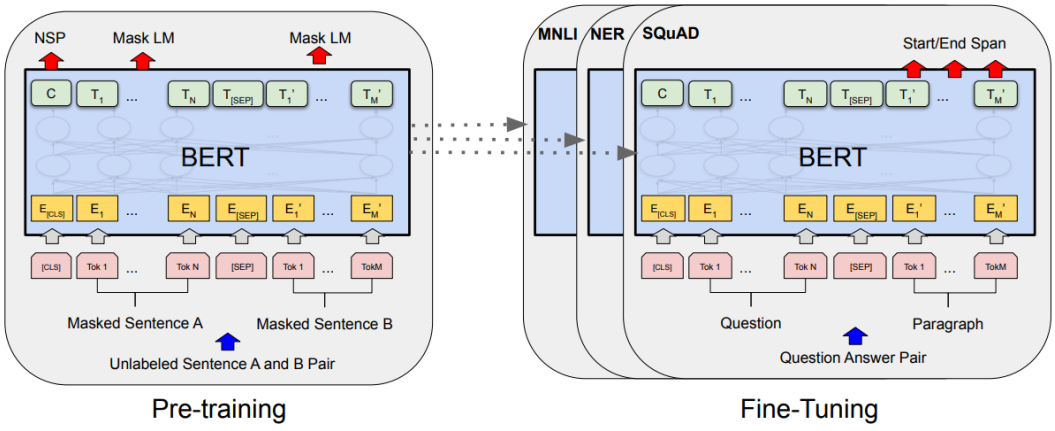

**ภาพที่ 3 อธิบายเกี่ยวกับกระบวนการสร้างอินพุตเพื่อเข้าสู่ตัวแบบจำลอง BERT (ที่มา: https://www.baeldung.com/cs/transformer-text-embeddings)**




**ภาพที่ 4 อธิบายเกี่ยวกับกระบวนการ Positional Encoding (ที่มา: https://machinelearningmastery.com/wp-content/uploads/2022/01/PE3.png)**

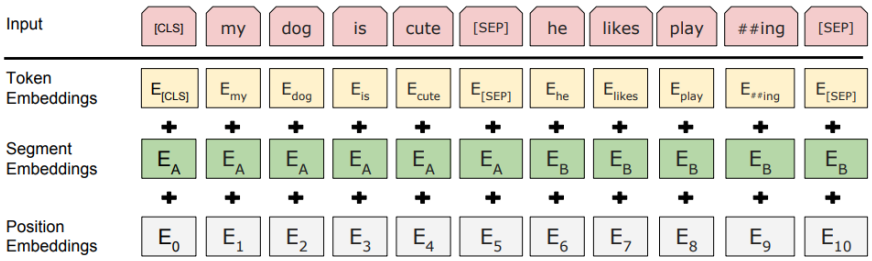
### 2.1.2 ตัวแบบจำลอง Bidirectional Encoder Representations จาก Transformers

ตัวแบบจำลอง Bidirectional Encoder Representations from Transformers หรือเรียกว่า BERT เป็นตัวแบบจำลองที่ล้ำสมัยสำหรับงานด้านภาษาศาสตร์ โดยถูกออกแบบมาเพื่อเป็นตัวแบบจำลองที่ถูกฝึกล่วงหน้า (Pre-Trained model) จากคลังข้อมูลขนาดใหญ่ โดยอาศัยการเรียนรู้แบบสองทิศทาง กล่าวคือตัวแบบจำลองสามารถเรียนรู้บริบทของประโยคทั้งทางซ้ายและทางขวาของประโยคซึ่งทำให้มีประสิทธิภาพที่ดีกว่าการเรียนรู้แบบทิศทางเดียว จากนั้นสามารถนำตัวแบบจำลองที่ถูกฝึกล่วงหน้าไปปรับแต่ง (Fine-Tuning) กับงานอื่น ๆ เช่น งานการจัดประเภทข้อความ งานตอบคำถาม โดยไม่จำเป็นต้องปรับเปลี่ยนสถาปัตยกรรมของตัวแบบจำลองเดิม ดังรูปที่ 5




**ภาพที่ 5 ขั้นตอนการฝึกอบรมล่วงหน้า (Pre-Train) และการปรับแต่ง (Fine-Tune) (ที่มา: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding)**

เพื่อให้ตัวแบบจำลอง BERT สามารถจัดการกับงานได้หลากหลาย อินพุตที่ใช้สำหรับตัวแบบจำลอง BERT ถูกออกแบบสามารถแสดงรูปแบบของประโยคทั้งแบบเดียวและประโยคแบบคู่ เช่น คำถาม “วันนี้สภาพอากาศเป็นอย่างไรบ้าง” คำตอบ “ท้องฟ้าสดใส” ซึ่งแบบนี้คือประโยคคู่ที่มีทั้งคำถามและคำตอบ ทางผู้สร้างใช้ WordPiece Embedding ด้วยโทเค็นทั้งหมด 30,000 คำ ซึ่งจะมีโทเค็นพิเศษ [CLS] ซึ่งทำหน้าที่เป็นโทเค็นแรกของประโยคแรก โคเทนพิเศษ [SEP] ใช้สำหรับแบ่งแยกประโยคระหว่างประโยคที่หนึ่งและประโยคที่สอง อีกทั้งยังมี Segmentatioin Embedding ทำหน้าที่ระบุโทเค็นในประโยคว่าเป็นของประโยคที่หนึ่งหรือประโยคที่สอง ดังรูปที่ 6



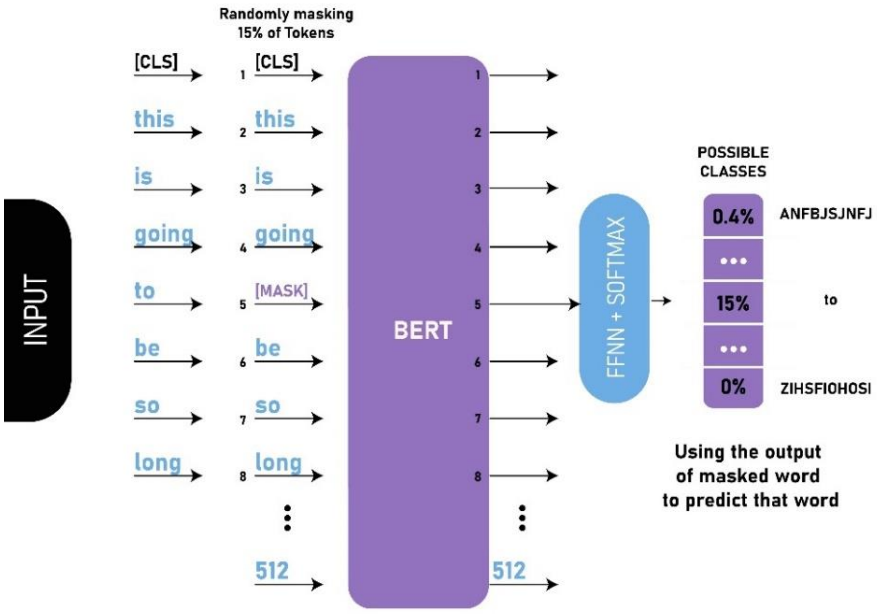

**ภาพที่ 6 แสดงอินพุตของตัวแบบจำลอง BERT**
**(ที่มา: Pre-Training of Deep Bidirectional Transformers for Language Understanding)**

ขั้นตอนสำหรับการฝึกอบรมล่วงหน้าของตัวแบบจำลอง BERT จะอาศัยงานที่ไม่มีมนุษย์ในการกำหนดป้ายกำกับในการเรียนรู้ (Unsupervised Learning) โดยใช้ 2 คลังข้อมูลขนาดใหญ่ได้แก่ BooksCorpus ซึ่งมีคำทั้งหมด 800 ล้านคำและ Wikipedia ภาษาอังกฤษที่มีคำทั้งหมด 2,500 ล้านคำ ซึ่งถูกนำมาฝึกอบรมด้วยงาน 2 งานคือ งานทำนายคำที่ถูกปิดบังไว้ (Masked LM) และงานการทำนายประโยคถัดไป (Next Sentence Prediction) ในท้ายที่สุดค่าสูญเสีย (Loss Function) ของทั้งสองงานจะถูกรวมและนำมาปรับให้เหมาะสมเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด

งานที่ 1 การทำนายคำที่ถูกปิดบังไว้ (Masked LM) งานนี้มีจุดประสงค์เพื่อฝึกการเรียนรู้ให้ตัวแบบจำลองเข้าใจบริบทของประโยคทั้งทางซ้ายและขวา วิธีการนี้จึงอาศัยการสุ่มปิดบังคำในประโยคร้อยละ 15 ของอินพุตโคเทน เพื่อให้ตัวแบบจำลองเรียนรู้จากคำที่ไม่ถูกปิดบังไว้และทำนายคำที่ถูกปิดบังไว้ให้ถูกต้อง โดยค่าความสูญเสียจะคิดเฉพาะคำทำนายที่เกิดจากคำที่ปิดบังไว้เท่านั้น ดังรูปที่ 7

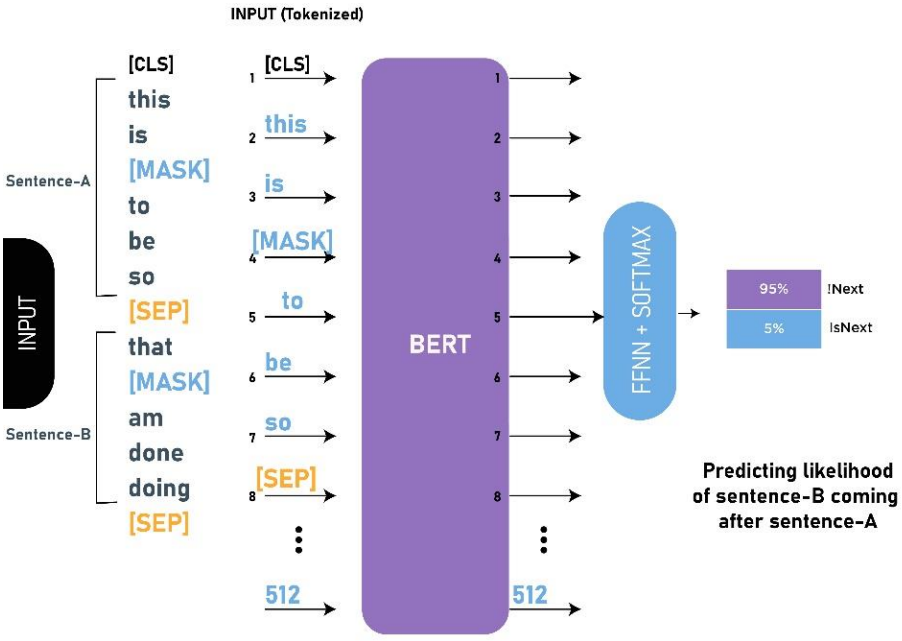
งานที่ 2 การทำนายประโยคถัดไป (Next Sentence Prediction) เนื่องจากมีความหลายงานจำเป็นต้องเข้าใจความสัมพันธ์ระหว่างสองประโยค เช่น งานการตอบคำถามและงานการอนุมานทางภาษา ดังนั้นจึงจำเป็นจะต้องฝึกอบรมงานที่ 2 โดยอินพุตจะมีประโยคเริ่มต้นสองประโยคและให้ตัวแบบจำลองทำนายว่าประโยคที่สองเป็นประโยคถัดไปของประโยคที่หนึ่งหรือไม่ โดยจะแบ่งชุดข้อมูลเป็นสัดส่วนร้อยละ 50 คือประโยคที่สองจะเป็นประโยคถัดไปของประโยคที่หนึ่งและส่วนที่เหลือจะไม่เกี่ยวข้องกัน เพื่อให้ตัวแบบจำลองเข้าใจบริบทในประโยคที่หนึ่งและประโยคที่สองจนสามารถทำนายได้อย่างถูกต้องแม่นยำได้ ดังรูปที่ 8

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University




ภาพที่ 7 วิธีการทำนายคำที่ถูกปิดบังไว้ (Masked LM)

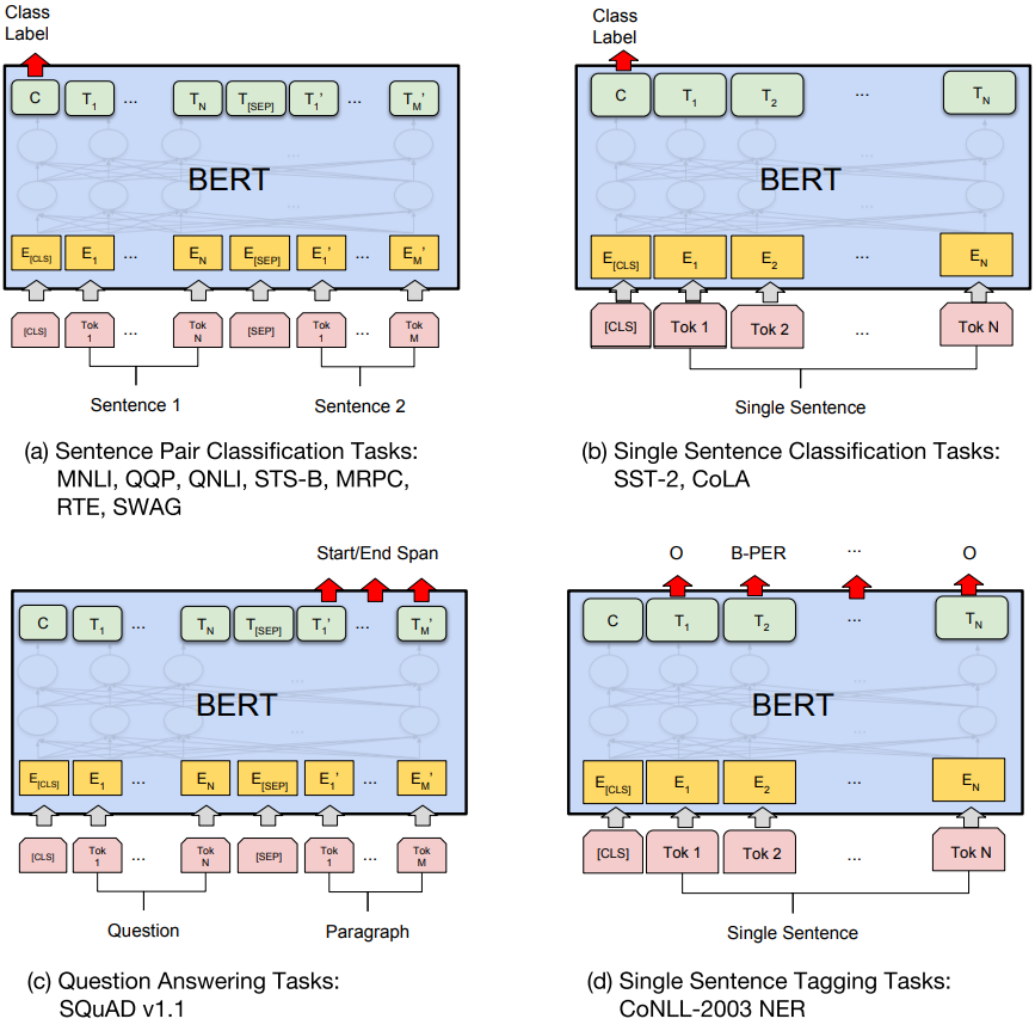
(ที่มา: https://www.geeksforgeeks.org/understanding-bert-nlp/)




ภาพที่ 8 วิธีการทำนายประโยคถัดไป (Next Sentence Prediction)

(ที่มา: https://www.geeksforgeeks.org/understanding-bert-nlp/)

หลังจากการฝึกอบรมล่วงหน้าด้วยวิธีการทั้งสองแล้ว การนำตัวแบบจำลอง BERT ไปใช้กับงานอื่นจะอาศัยวิธีการที่เรียกว่า การปรับจูนละเอียด (Fine-Tuning) ซึ่งมีความตรงไปตรงมา กล่าวคือสามารถนำอินพุตและเอาต์พุตของงานที่ต้องการมาใช้ในการปรับแต่งพารามิเตอร์ของตัวแบบจำลอง BERT โดยการฝึกอบรมล่วงหน้าด้วยวิธีการทำนายประโยคถัดไปมีความคล้ายคลึงกับงานด้านการถอดความแบบประโยคคู่ คู่สมมติฐาน-ข้อความที่เกี่ยวข้อง งานการถามตอบและงานการแบ่งประเภทประโยคโดยโทเค็นพิเศษ [CLS] จะถูกนำมาใช้ในการแบ่งประเภท (Classification) เพื่อทำนายผลลัพธ์ ดังรูปที่ 9

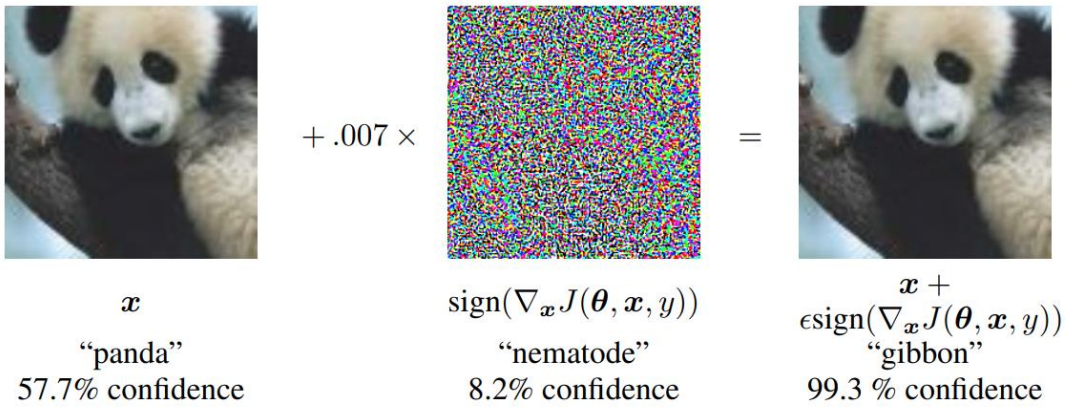



**ภาพที่ 9 Fine-Tuning BERT ในงานที่แตกต่างกัน**

**(ที่มา: Pre-Training of Deep Bidirectional Transformers for Language Understanding)**

### 2.1.3 วิธีการฝึกอบรมปรปักษ์ (Adversarial Training)

Goodfellow นำเสนอวิธีการฝึกอบรมปรปักษ์ (Adversarial Training) ใช้ในงานคอมพิวเตอร์วิทัศน์ โดยสร้างอินพุตสำหรับตัวแบบจำลองโดยอาศัยการใช้การรบกวนเพียงเล็กน้อยแต่ตั้งใจให้เกิดความเข้าใจผิดกับตัวอย่างของชุดข้อมูลในมากที่สุด ซึ่งส่งผลให้เมื่อตัวแบบจำลองเรียนรู้อินพุตที่เกิดจากการรบกวนจะทำให้ทำนายด้วยความเชื่อมั่นที่สูงแต่ผลลัพธ์ไม่ถูกต้องตามป้ายกำกับ ยกตัวอย่างเช่น ตัวแบบจำลองทายอินพุตดั้งเดิมว่ารูปนี้คือ “หมีแพนด้า” ที่ความเชื่อมั่นร้อยละ 57.7 แต่เมื่อเพิ่มการรบกวนเข้าไปยังรูปเดิมแล้วให้ตัวแบบจำลองทำนายอีกครั้ง ตัวแบบจำลองจะทำนายอินพุตที่ถูกการรบกวนเป็น “ชะนี” ที่ความเชื่อมั่นร้อยละ 99.3 ดังรูปที่ 10 จากวิธีนี้จะทำให้ตัวแบบจำลองพยายามเรียนรู้จากข้อมูลอื่นที่เกี่ยวข้องเพื่อทำนายผลลัพธ์ให้ออกมาถูกต้องตามป้ายกำกับและทำให้ปัญหาเรื่อง Overfitting ได้รับการบรรเทา




จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**ภาพที่ 10 อธิบายการทำงานของ Adversarial Training**
**(ที่มา: Explaining and Harnessing adversarial examples)**

เมื่อนำตัวอย่าง Adversarial มาปรับใช้กับตัวแบบจำลองสำหรับงานจำแนกข้อมูล ค่าความสูญเสียของ Adversarial สามารถแสดงได้ดังสมการนี้

$$Loss\ Function = -\log p(y|x + r_{adv}; \theta)$$

$$r_{adv} = \underset{r,\|r\| \le \epsilon}{\operatorname{argmin}} \log p\left(y\middle|x + r; \hat{\theta}\right)$$

เมื่อ

$\boldsymbol{x}$ คือ อินพุตข้อมูลดั้งเดิม

$r$ คือ ตัวรบกวนเพียงเล็กน้อย

$\hat{\theta}$ คือ ค่าประมาณพารามิเตอร์

$r_{adv}$ คือ ตัวรบกวนที่เลวร้ายที่สุด

อย่างไรก็ตามการคำนวนเพื่อสร้างตัวรบกวนที่เลวร้ายที่สุดไม่สามารถคำนวณได้โดยวิธีทั่วไป ดังนั้น Goodfellow เสนอให้ใช้วิธีการประมาณค่าโดยใช้วิธี backpropagation สำหรับโครงข่ายประสาทเทียม (Neural Networks) ท้ายที่สุดการคำนวณตัวรบกวนที่เลวร้ายที่สุดได้ดังสมการนี้

$$r_{adv} = -\varepsilon\, g/\|g\|_2$$
$$g = \nabla_x \log p(y|x;\hat{\theta})$$

**2.1.4 วิธีการฝึกปรปักษ์เสมือน (Virtual Adversarial Training)**

วิธีการ Virtual Adversarial Training เป็นวิธีการสร้างตัวรบกวนแบบเสมือน (Virtual Adversarial Example) ซึ่งมีความแตกต่างจากวิธีการ Adversarial Training โดยไม่อาศัยป้ายกำกับในการสร้างการรบกวนที่เลวร้ายที่สุดและสามารถนำไปประยุกต์ใช้กับการเรียนรู้แบบกึ่งมีผู้สอน (Semi-Supervised Learning) โดยให้ตัวแบบจำลองทำนายอินพุตดั้งเดิมเพื่อให้ได้ผลลัพธ์ปกติและนำไปเปรียบเทียบกับผลลัพธ์ของตัวรบกวนที่เกิดจากตัวแบบจำลองทำนายอินพุตที่รวมกับตัวรบกวนแบบสุ่ม สุดท้ายสมการความสูญเสียจะไปเป็นตามสมการดังนี้

$$L_{v-adv}(\theta) = {}^{1}/_{N'} \sum_{n'}^{N'} KL\left[q(y|x_*,\hat{\theta}), p(y|x_* + r_{v-adv},\theta)\right]$$

$$r_{v-adv} = \underset{r,\|r\|\leq\epsilon}{arg\max}\, KL\left[q(y|x_*,\hat{\theta}), p(y|x_* + r,\theta)\right]$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โดยที่ $x_*$ คือ อินพุตของข้อมูล

$\theta$ คือ พารามิเตอร์ของตัวแบบจำลอง

$N'$ คือ จำนวนของข้อมูลที่มีป้ายกำกับและไม่มีป้ายกำกับ

$r$ คือ ตัวรบกวบตั้งต้นจากการสุ่ม

$r_{v-adv}$ คือ ตัวรบกวนที่เลวร้ายที่สุด

$KL[p\|q]$ คือ Kullback-Leibler Divergence ระหว่าง p และ q

เนื่องจากการคำนวณตัวรบกวนที่เลวร้ายที่สุดไม่สามารถทำได้ ดังนั้นจึงใช้วิธีการประมาณตัวรบกวนที่เลวร้ายที่สุดของ Virtual Adversarial Training โดยใช้

$$r_{v-adv} = \varepsilon\, g/\|g\|_2$$

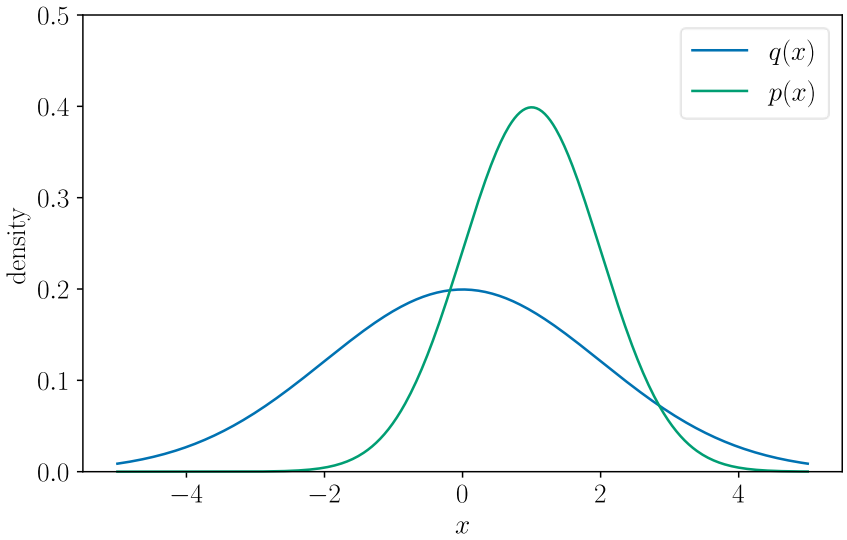
$$g = \nabla_x KL\left[q\left(y|x_*, \hat{\theta}\right), p(y|x_* + r, \theta)\right]$$

### 2.1.5 วิธีการ Kullback-Leibler Divergence

Kullback-Leibler Divergence เป็นวิธีการหาความแตกต่างระหว่างสองการแจกแจงความน่าจะเป็น ยกตัวอย่างเช่น จากรูปที่ 11 ต้องการวัดความแตกต่างระหว่างการแจกแจงความน่าจะเป็น q และการแจงแจกความน่าจะเป็น p สามารถเขียนสัญลักษณ์ได้ดังนี้

$$KL[p(x)\|q(x)]$$

โดยที่ || คือ เครื่องหมายดำเนินการความแตกต่าง (divergence)




**ภาพที่ 11 ตัวอย่างการแจกแจงความน่าจะเป็น p(x) และ q(X)**
**(ที่มา: https://tiao.io/post/density-ratio-estimation-for-kl-divergence-minimization-between-implicit-distributions/)**

จากสมการข้างต้นหมายความว่าปริมาณข้อมูลที่สูญเสียไปเมื่อใช้ $q(x)$ เพื่อประมาณค่า $p(x)$ หรือก็คือความแตกต่างระหว่างสองการแจกแจงความน่าจะเป็นนั่นเอง ในกรณีที่เป็นการแจกแจงความน่าจะเป็นแบบไม่ต่อเนื่องสามารถคำนวณได้จากสมการดังนี้

$$KL[p(x)||q(x)] = \sum_{x \in X} p(x) \ln p(x)/q(x)$$

โดยที่ $p(x)$ คือ การแจกแจงความน่าจะเป็นแบบไม่ต่อเนื่อง

$q(x)$ คือการแจกแจงความน่าจะเป็นแบบไม่ต่อเนื่อง

$x$ คือ ตัวแปรสุ่ม

กรณีการแจกแจงความน่าจะเป็นแบบต่อเนื่องสามารถคำนวณได้จากสมการนี้

$$KL[p(x)||q(x)] = \int_{-\infty}^{\infty} p(x) \ln p(x)/q(x)\, dx$$

โดยที่ $p(x)$ คือ การแจกแจงความน่าจะเป็นแบบต่อเนื่อง

$q(x)$ คือการแจกแจงความน่าจะเป็นแบบต่อเนื่อง

$x$ คือ ตัวแปรสุ่ม

แม้ว่า KL-Divergence จะเป็นการวัด “ระยะทาง” ของสองการแจกแจงความน่าจะเป็นแต่ไม่ใช่วัดระยะทาง เพราะว่าฟังก์ชัน KL-Divergence ไม่สมมาตร กล่าวคือ KL ระหว่าง $p(x)$ ไป $q(x)$ ไม่เท่ากับ KL จาก $p(x)$ ไป $q(x)$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**2.2 งานวิจัยที่เกี่ยวข้อง**

มีความพยายามหลายครั้งในการแก้ไขปัญหา Overfitting บนตัวแบบจำลอง วิธีการแรกคือการเพิ่มข้อมูลการฝึกอบรมโดยใช้วิธีการเสริมข้อมูล (Data Augmentation) ซึ่งเป็นเทคนิคที่สร้างข้อมูลสำหรับการฝึกอบรมขึ้นมาใหม่จากข้อมูลเดิมที่มีอยู่ (Kobayashi, 2018) เสนอวิธีการใหม่สำหรับการเสริมข้อมูลที่เรียกว่า การเสริมข้อมูลตามบริบท (Contextual Augmentation) ซึ่งจะเพิ่มข้อมูลตามบริบทของประโยคเดิม ในทำนองเดียวกัน (Wu et al., 2019) นำเสนอ Condition BERT Contextual ซึ่งมีการดัดแปลงตัวแบบจำลอง BERT ด้วยสถาปัตยกรรมการเงื่อนไขของป้ายกำกับ (Label-Condition) เพื่ออนุญาตให้ตัวแบบจำลองสามารถเสริมข้อมูลโดยไม่ทำลายความเข้ากันได้ของประโยคที่สร้างใหม่และป้ายกำกับ (Wei & Zou, 2019) นำเสนอวิธี Easy Data Augmentation (EDA) ที่เสริมข้อมูลเพื่อเพิ่มประสิทธิภาพการทำงานของตัวแบบจำลอง อย่างไรก็ตามข้อจำกัดหลักของวิธีการเสริมข้อความคือ ข้อความที่ถูกสร้างขึ้นมาอาจจะมีบริบทที่ขัดแย้งกับป้ายกำกับและสามารถตรวจสอบประโยคเหล่านั้นได้ยาก

วิธีการที่สองคือการสร้างงานเสริม (Auxiliary Task) เพื่อให้ตัวแบบจำลองเรียนรู้และแก้ไขงานเสริม (Zhou et al., 2021) เสนอวิธีการใช้ Data-Dependent Regularization โดยสร้างงานการเรียนรู้แบบเรียนรู้ด้วยตนเอง (Self-supervised Learning) เพื่อแก้ไขปัญหา Overfitting (Kou et al., 2021) เสนอวิธีการ Self-supervised attention (SSA) และเพิ่มชั้น SSA ลงในตัวแบบจำลอง BERT โดย SSA จะสร้างป้ายกำกับความสำคัญสำหรับระดับโทเค็นที่มีความอ่อนไหวต่อการทำนายผล (Gunel et al., 2021) นำเสนอการรวม Contrastive Learning เข้ากับ Cross Entropy วิธีการนี้สามารถเพิ่มประสิทธิภาพของ RoBERTa-large บนการวัดประสิทธิภาพของ GLUE จากงานที่กล่าวมาแนวคิดพื้นฐานที่อยู่เบื้องงานเหล่านี้คือ การเพิ่มงานเสริมที่เกี่ยวข้องให้กับงานหลักเพื่อปรับปรุงประสิทธิภาพ แม้ว่างานเสริมเหล่านี้จะสามารถช่วยแก้ไขปัญหา Overfitting แต่การสร้างงานเสริมค่อนข้างมีความเฉพาะและทางเหลือที่หลากหลายซึ่งนำไปสู่ผลลัพธ์ที่แตกต่างกัน นอกจากนี้งานเสริมบางงานยังมีความท้าทายในการสร้างป้ายกำกับและใช้เวลานานอีกด้วย

เพื่อจะแก้ไขปัญหา Overfitting มีอีกแนวทางหนึ่งที่นำมาจากสาขาคอมพิวเตอร์วิทัศน์ (Goodfellow et al., 2015) ที่เรียกว่า Adversarial Training ในแนวทางนี้ ตัวอย่างจะถูกสร้างขึ้นโดยใช้การรบกวนเล็กน้อยกับข้อมูลอินพุต เมื่อเพิ่มการรบกวนเล็ก ๆ น้อย ๆ ให้กับข้อมูลดั้งเดิมในกระบวนการฝึกอบรม ตัวอย่าง Adversarial สามารถหลอกตัวแบบจำลองให้คาดการณ์ผลลัพธ์ที่ไม่

ถูกต้องได้ การพยายามที่จะเรียนรู้ตัวอย่าง Adversarial ทำให้การเรียนรู้ของแบบจำลองมีประสิทธิภาพมากขึ้น นอกจากนี้ยังสามารถแก้ไขปัญหา Overfitting อีกด้วย

มีการศึกษามากมายในสาขาภาษาศาสตร์ (NLP) ที่ใช้แนวทางวิธี Adversarial Training เพื่อปรับใช้กับข้อมูลที่ไม่ต่อเนื่องเช่น ข้อความ (Miyato, Dai, & Goodfellow, 2017) ขยายวิธีการ Adversarial Training และ Virtual Adversarial Training กับงานด้านภาษาศาสตร์โดยปรับใช้กับการฝังคำ (Word embedding) วิธีการนี้จะเพิ่มประสิทธิภาพในการจัดประเภทข้อความแบบการเรียนรู้แบบกึ่งมีผู้สอน (Semi-supervised Learning) และสามารถบรรเทาปัญหา Overfitting (Zhu et al., 2020) นำเสนอ FreeLB ซึ่งเป็นอัลกอริทึมสำหรับเร่งความเร็วให้กับวิธีการฝึกปรปักษ์ (Adversarial Training) โดยใช้ขั้นตอนการแพร่ย้อนกลับ (Backward Step) (Jiang et al., 2020) เสนอเทคนิค Regularization ที่เรียกว่า SMART ซึ่งเน้นที่ความราบเรียบรื่นในท้องถิ่น (Local Smoothness) แนวคิดพื้นฐานเบื้องหลังเกี่ยวกับวิธีการนี้คือ หากมีการรบกวนเล็กน้อยในการฝังคำในประโยคอินพุต (Sentence Embedding) จะทำให้ขอบเขตการตัดสินใจ (Decision Boundary) ของตัวแบบจำลองจะไม่เปลี่ยนแปลงมากนัก ดังนั้น SMART จึงสามารถช่วยให้ตัวแบบจำลองมีความราบเรียบภายในการฝังคำในประโยคอินพุต SMART มีความคล้ายคลึงกับวิธีการฝึกปรปักษ์เสมือน (Virtual Adversarial Training) ของ (Miyato et al., 2017) แต่มีภารกิจเป้าหมายที่แตกต่างกัน (X. Liu, Cheng, et al., 2020) เสนออัลกอริทึม ALUM เพื่อเพิ่มความสูญเสียของ การฝึกปรปักษ์ (Adversarial Training) ให้มากที่สุด (Pereira et al., 2021) นำเสนอ Target Adversarial Training โดยเน้นไปที่อินพุตที่ตัวแบบจำลองทำนายผลลัพธ์ผิด (Li & Qiu, 2021) ใช้การรบกวนในระดับโทเค็นบนอัลกอริทึม FreeLB (Karimi, Rossi, Prati, & Full, 2021) นำเสนอ BERT Adversarial Training (BAT) ใช้วิธีการ Adversarial Training กับงานการจัดประเภทข้อความ (Aspect Extraction) และการจัดประเภทความรู้สึก (Aspect Sentiment Classification) (Pereira et al., 2020) เสนออัลกอริทึม ALICE ซึ่งเป็นการผสมระหว่างวิธีฝึกปรปักษ์และวิธีฝึกปรปักษ์เสมือนเพื่อพัฒนาประสิทธิภาพของตัวแบบจำลอง

แม้ว่างานที่มีอยู่จะเป็นวิธีที่มีแนวโน้มในการลดปัญหา Overfitting แต่ก็ประสบปัญหาขาดการรบกวนที่ละเอียดในระดับโทเค็น เนื่องจากงานส่วนใหญ่อาศัยการฝึกปรปักษ์เสมือนแบบดั้งเดิม ซึ่งใช้ Normalization Ball ระดับประโยคที่ทำหน้าที่คำนวณค่าแบบทั้งประโยค ส่งผลให้โทเค็นทั้งหมดมีค่า Normalization Ball เท่ากัน ดังนั้นวิธีการดังกล่าวจึงไม่ละเอียดพอที่จะสร้างการรบกวนระดับโทเค็น การขาดการรบกวนอย่างละเอียดในระดับโทเค็นทำให้ยากสำหรับตัวแบบจำลองในการ

ตีความหมายของประโยคได้อย่างถูกต้อง ยกตัวอย่างเช่น ให้พิจารณาประโยคเชิงบวก “A whole lot foul, freaky and funny” จากชุดข้อมูล SST (Socher et al., 2013) ในประโยคนี้หากตัวแบบจำลองให้ความสำคัญกับกับคำว่า “Foul” แบบจำลองจะทำนายว่าทั้งประโยคเป็นประโยคเชิงลบ ในทางกลับกันถ้าตัวแบบจำลองให้ความสนใจกับคำว่า “Funny” การทำนายของตัวแบบจำลองจะบ่งบอกว่าเป็นประโยคเชิงบวก ดังนั้นตัวแบบจำลองที่มีการฝึกปรปักษ์เสมือนแบบดั้งเดิมจึงไม่สามารถให้ค่าประมาณที่แม่นยำเมื่อโทเค็นที่สำคัญสำหรับการทำนายไม่ได้ถูกรบกวนเป็นพิเศษ

เพื่อเอาชนะข้อจำกัดนี้ (Kou et al., 2021) แนะนำให้พิจารณาข้อมูลเพิ่มเติมเกี่ยวกับคำหลัก (keyword) ในประโยคเพื่อให้ตัวแบบได้เรียนรู้คำอื่น ๆ มากขึ้น โดยผู้เขียนแสดงให้เห็นว่าตัวแบบจำลอง BERT ที่ปรับแต่งแบบละเอียดสามารถบรรลุความถูกต้องแม่นยำยิ่งขึ้นเมื่อมีการปิดบังโทเค็นที่ทำให้เข้าใจผิด (Masked Misleading Token) นอกจากนี้งานของพวกเขายังแสดงให้เห็นว่าโทเค็นเพียงอย่างเดียวไม่เพียงพอที่จะช่วยให้ตัวแบบจำลองจำแนกความขัดแย้งทางอารมณ์ของประโยคทั้งหมด ซึ่งอาจจะขึ้นกับบริบทของประโยคด้วย ดังนั้นโทเค็นและบริบทของประโยคจึงมีความสำคัญสำหรับงานด้านภาษาศาสตร์ อย่างไรก็ตามวิธีการของ (Kou et al., 2021) ไม่เหมาะสมกับตัวแบบจำลอง BERT เนื่องจากการสร้างฉลากของงานเสริมอาจจะใช้เวลานาน

แนวทางของงานวิจัยนี้ ขอนำเสนอการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) ซึ่งรวมการรบกวนสองระดับได้แก่ (1) การรบระดับประโยคของการฝึกปรปักษ์เสมือนแบบดั้งเดิม (2) การรบกวนโทเค็นแบบถ่วงน้ำหนัก ในการประมาณการว่าการเปลี่ยนของอินพุตใดสร้างความสูญเสียโดยประมาณสูงสุดเพื่อสร้างการรบกวนที่เลวร้ายที่สุด วิธีการของเราใช้วิธี Gradient-base (Goodfellow et al., 2015) (Zhu et al., 2020) โดยเฉพาะกับ Kullbakck-Leibler Divergence (Jiang et al., 2020), (Miyato et al., 2017) ในการรบกวนระดับประโยคจะอาศัย Normalization Ball ระดับประโยคคล้ายกับงานก่อนหน้านี้ (Miyato et al., 2017), (Jiang et al., 2020) แต่รวมกับการรบกวนโทเค็นแบบถ่วงน้ำหนัก การรบกวนโทเค็นแบบถ่วงน้ำหนักนี้ทำหน้าที่เป็นองค์ประกอบหลักที่ทำให้แต่ละโทเค็นมีค่า Normalization Ball แตกต่างกันโดยขึ้นอยู่กับระดับความสำคัญของโทเค็นนั้น การรบกวนโทเค็นแบบถ่วงน้ำหนักแบบใหม่สามารถบังคับให้ตัวแบบจำลองขยายการรบกวนสำหรับโทเค็นที่มีความสำคัญที่มีการไล่ระดับสีที่ใหญ่ (Larger Gradient) และลดการรบกวนบนโทเค็นที่ไม่มีความสำคัญที่มีการไล่ระดับสีขนาดเล็ก (Smaller Gradient)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 3
# วิธีการดำเนินการวิจัย

บทนี้จะอธิบายถึงเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE) และวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) สำหรับงานจัดประเภทข้อความ โดยวิธีการนี้อาศัยวิธีการฝึกปรปักษ์เสมือนแบบดั้งเดิมเป็นหลัก โดยจะรวมการรบกวนระดับประโยคและระดับโทเค็นเข้าด้วยกันเพื่อจะจัดการกับโทเค็นที่มีความสำคัญต่อการทำนายและบริบทของประโยค โดยวิธีการของเรามุ่งเน้นที่ตัวแบบจำลอง BERT เป็นหลัก

## 3.1 เกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE)

เกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (GLUE) (Wang et al., 2018) ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อให้เทคโนโลยีการเข้าใจภาษาธรรมชาติ (Natural language understanding: NLU) เกิดประโยชน์สูงสุดโดยจะต้องสามารถประมวลผลภาษาไม่เฉพาะแค่งานประเภทเดียวหรือชุดข้อมูลเดียว เกณฑ์มาตรฐานการประเมินความเข้าใจ (GLUE) ซึ่งเป็นชุดเครื่องมือสำหรับการประเมินประสิทธิภาพของแบบจำลองในงาน NLU ที่มีอยู่หลายชุดข้อมูล โดยบางชุดข้อมูลมีจำนวนข้อมูลสำหรับการฝึกอบรมมีจำกัดหรือก็คือชุดข้อมูลขนาดเล็ก ดังตารางที่ 1

*ตารางที่ 1 แสดงรายละเอียดโดยรวมของชุดข้อมูลภายใน GLUE Benchmark*

| Corpus | Task | Metrics | Train | Dev | Test |
|---|---|---|---|---|---|
| CoLA | Acceptability | Matthew corr. | 8.5k | 1k | 1k |
| SST | Sentiment | Accuracy | 67k | 872 | 1.8k |
| MRPC | NLI | Accuracy/F1 | 3.7k | 408 | 1.7k |
| QQP | Paraphrase | Accuracy/F1 | 364k | 40k | 391k |
| STS-B | Similarity | Pearson/Spearman corr. | 7k | 1.5k | 1.4k |
| MNLI | NLI | Accuracy | 393k | 20k | 20k |
| QNLI | QA/NLI | Accuracy | 108k | 5.7k | 5.7k |
| RTE | NLI | Accuracy | 2.5k | 276 | 3k |
| WNLI | NLI | Accuracy | 636 | 72 | 147 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (GLUE) เน้นงานการเข้าประโยคภาษาอังกฤษโดยมีทั้งหมด 9 ชุดข้อมูล โดยทั้งหมดครอบคลุมทุกโดเมน รวมทั้งปริมาณข้อมูลที่มีความแตกต่างกันและความยากที่แตกต่างกันในแต่ละชุดข้อมูล โดยสามารถจัดกลุ่มงานได้ 3 กลุ่ม ได้แก่ 1.งานด้านภาษาแบบประโยคเดียว (Single-sentence Tasks) 2.งานความคล้ายคลึงกันและการถอดความ (Similarity and Paraphrase Tasks) 3.งานอนุมานด้านภาษา (Inference Tasks)

**3.1.1 งานด้านภาษาแบบประโยคเดียว (Single-sentence Tasks)**

1) The Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019) เป็นชุดข้อมูลที่ประกอบด้วยข้อมูลที่นำมาจากหนังสือและบทความในวารสารภาษอังกฤษที่เกี่ยวข้องกับทฤษฎีภาษาศาสตร์ โดยแต่ละตัวอย่างจะมีป้ายกำกับว่าประโยคดังกล่าวเป็นไปตามหลักไวยากรณ์หรือไม่ ดังตัวอย่างในตารางที่ *2* ชุดข้อมูล CoLA ใช้ค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิว (Mathew's Correlation Coefficient) เป็นตัวชี้วัดในการประเมินผล ซึ่งมีความสามารถในการประเมินประสิทธิภาพในชุดข้อมูลแบบไม่สมดุล (Unbalance Binary Classification)

*ตารางที่ 2 แสดงตัวอย่างจากชุดข้อมูล CoLA*

| Index | ประโยค | ป้ายกำกับ |
|---|---|---|
| 1 | Our friends won't buy this analysis, let alone the next one we propose. | 1 (acceptable) |
| 2 | Bill pushed Harry off the sofa for hours. | 0 (unacceptable) |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2) The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) เป็นชุดข้อมูลที่ประกอบด้วยบทวิจารณ์และคำอธิบายเกี่ยวกับความรู้สึกจากภาพยนตร์ ภารกิจของชุดข้อมูลนี้คือการทำนายความรู้สึกของประโยคที่กำหนดไว้ โดยแบ่งเป็น *2* กลุ่มคือ ความรู้สึกเชิงบวกและความรู้สึกเชิงลบ ดังตารางที่ 3

*ตารางที่ 3 แสดงตัวอย่างจากชุดข้อมูล SST-2*

| Index | ประโยค | ป้ายกำกับ |
|---|---|---|
| 1 | equals the original and in some ways even betters it | 1 (positive) |
| 2 | cold movie | 0 (negative) |

**3.1.2 งานความคล้ายคลึงกันและการถอดความ (Similarity and Paraphrase Tasks)**

1) The Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005) เป็นคลังข้อมูลของคู่ประโยคที่ดึงมากจากแหล่งข่าวออนไลน์โดยอัตโนมัติ โดยมีป้ายกำกับว่าประโยคทั้งคู่มีความหมายเหมือนกันหรือไม่ ดังตารางที่ 4 และเนื่องจากข้อมูลชุดนี้ไม่สมดุล (Imbalance Dataset) จึงใช้ค่าความถูกต้องและค่า F1-score ในการวัดผล

*ตารางที่ 4 แสดงตัวอย่างจากชุดข้อมูล MRPC*

| Index | ประโยค *1* | ประโยค *2* | ป้ายกำกับ |
|---|---|---|---|
| 1 | They had published an advertisement on the Internet on June 10, offering the cargo for sale, he added. | On June 10, the ship 's owners had published an advertisement on the Internet, offering the explosives for sale. | 1 (equivalent) |
| 2 | Yucaipa owned Dominick 's before selling the chain to Safeway in 1998 for $ 2.5 billion. | Yucaipa bought Dominick 's in 1995 for $ 693 million and sold it to Safeway for $ 1.8 billion in 1998. | 0 (Not equivalent) |

2) The Quora Question Pairs (QQP) คือ คู่ของชุดคำถามจากเว็บไซต์ตอบคำถามอย่าง Quora โดยภารกิจหลักของชุดข้อมูลนี้คือ พิจารณาว่าคำถามคู่หนึ่งมีความหมายเหมือนกันหรือไม่ ดังตารางที่ 5 และเช่นเดียวกับ MRPC จำนวนของป้ายกำกับมีความไม่สมดุล ดังนั้นจึงต้องอาศัยค่าความถูกต้องและ F1-score

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

*ตารางที่ 5 แสดงตัวอย่างของชุดข้อมูล QQP*

| Index | *คำถาม* 1 | *คำถาม 2* | *ป้ายกำกับ* |
|---|---|---|---|
| 1 | How do I control my horny emotions? | How do you control your horniness? | 1 (duplicate) |
| 2 | What causes stool color to change to yellow? | What can cause stool to come out as little balls? | 0 (Not duplicate) |

3) The Semantic Textual Similarity Benchmark (STS-B) (Cer, Diab, Agirre, Lopez-Gazpio, & Specia, 2017) เป็นชุดของประโยคคู่ที่นำมาจากหัวข้อข่าว หัวข้อวิดีโอและคำบรรยายภาพและข้อมูลอนุมานภาษาธรรมชาติ โดยประโยคแต่ละคู่ได้รับคะแนนความคล้ายคลึงกันตั้งแต่ 1 ถึง 5 ดังตารางที่ 6 ภารกิจหลักของข้อมูลชุดนี้คือ การทำนายคะแนนความคล้ายคลึงกัน การประเมินผลจะใช้ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและสเปียร์แมน (Pearson and Spearman Correlation Coefficients)

*ตารางที่ 6 แสดงตัวอย่างของชุดข้อมูล STS-B*

| Index | ประโยค 1 | ประโยค 2 | ป้ายกำกับ |
|---|---|---|---|
| 1 | A man is smoking. | A man is skating. | 0.5 |
| 2 | Three men are playing chess. | Two men are playing chess. | 2.6 |
| 3 | A man is playing a large flute. | A man is playing a flute. | 3.8 |
| 4 | A plane is taking off. | An airplane is taking off. | 5 |

**3.1.3 งานอนุมานด้านภาษา (Inference Tasks)**

1) The Multi-Genre Natural Language Inference Corpus (MNLI) (Williams, Nangia, & Bowman, 2018) เป็นชุดข้อมูลที่รวบรวมจากแหล่งข้อมูล 10 แหล่ง เช่น คำพูดในการปราศรัย นิยายและรายงานของรัฐบาล โดยมีลักษณะเป็นคู่ประโยค (ประโยคหลักฐานและประโยคสมมติฐาน) เพื่อทำนายว่าประโยคหลักฐานและประโยคสมมติฐานนั้นมีความเกี่ยวข้องหรือขัดแย้งกันหรือเป็นกลาง ดังตัวอย่างในตารางที่ 7 การประเมินจะแบ่งออกเป็น 2 กลุ่มคือ (1) Matched (In-domain) และ (2) Mismatched (Cross-domain)

*ตารางที่ 7 แสดงตัวอย่างของชุดข้อมูล MNLI*

| Index | ประโยคหลักฐาน | ประโยคสมมติฐาน | ป้ายกำกับ |
|---|---|---|---|
| 1 | How do you know? All this is their information again. | This information belongs to them. | 0 (entailment) |
| 2 | Were they in there? | Were they supposed to be in there? | 1 (neutral) |
| 3 | Vrenna and I both fought him, and he nearly took us. | Neither Vrenna nor myself have ever fought him. | 2 (contradiction) |

2) The Stanford Question Answering (QNLI) (Rajpurkar, Zhang, Lopyrev, & Liang, 2016) เป็นชุดข้อมูลที่ประกอบด้วยคู่ประโยคคำถาม ซึ่งนำมาจาก Wikipedia โดยประโยคหนึ่งเป็นประโยคคำถามและอีกประโยคเป็นประโยคคำตอบที่เกี่ยวข้อง โดยภารกิจหลักคือการทำนายว่าประโยคคำตอบนี้ใช่คำตอบของประโยคคำถามหรือไม่ ดังตัวอย่างในตารางที่ 8

*ตารางที่ 8 แสดงตัวอย่างของชุดข้อมูล QNLI*

| Index | ประโยคคำถาม | ประโยค | ป้ายกำกับ |
|---|---|---|---|
| 1 | In what century was the church established at the location? | Construction of the present church began in *1245*, on the orders of King Henry III. | 1 (Not entailment) |
| 2 | How many paper cups are used by Americans each year? | Americans also use on the order of 16 billion paper cups per year. | 0 (entailment) |

3) The Recognizing Textual Entailment (RTE) (Dagan et al., 2005) เป็นชุดข้อมูลที่สร้างขึ้นจากข้อมูลข่าวและข้อความ Wikipedia โดยมีป้ายกำกับ 2 ป้ายคือ ไม่มีความเชื่อมโยงและมีความเชื่อมโยงกัน ดังตัวอย่างในตารางที่ 9

*ตารางที่ 9 แสดงตัวอย่างของชุดข้อมูล RTE*

| Index | ประโยค 1 | ประโยค 2 | ป้ายกำกับ |
|---|---|---|---|
| 1 | No Weapons of Mass Destruction Found in Iraq Yet. | Weapons of Mass Destruction Found in Iraq. | 1 (Not entailment) |
| 2 | Lin Piao, after all, was the creator of Mao's "Little Red Book" of quotations. | Lin Piao wrote the "Little Red Book". | 0 (entailment) |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4) The Winograd Schema Challenge (WNLI) เป็นชุดข้อมูลที่ใช้การทำความเข้าใจสรรพนามภายในประโยค ภารกิจหลักของชุดข้อมูลนี้คือ ทำนายว่าสรรพนามในประโยคหลังเกี่ยวข้องกับประโยคแรกหรือไม่ ดังตัวอย่างในตารางที่ 10

*ตารางที่ 10 แสดงตัวอย่างของชุดข้อมูล WNLI*

| Index | ประโยค 1 | ประโยค 2 | ป้ายกำกับ |
|---|---|---|---|
| 1 | I stuck a pin through a carrot. When I pulled the pin out, it had a hole. | The carrot had a hole. | 1 (entailment) |
| 2 | George got free tickets to the play, but he gave them to Eric, because he was particularly eager to see it. | George was particularly eager to see it. | 0 (Not entailment) |

**ตัววัดสำคัญในการประเมินสำหรับ GLUE benchmark**

**ค่าความถูกต้อง (Accuracy)**

เป็นสัดส่วนของการทำนายที่ถูกต้อง นั้นคือผลบวกจริง (True Positive) และผลลบจริง (True Negative) โดยสามารถเขียนเป็นสมการได้ดังนี้

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

โดยที่

TP คือ ผลบวกจริง

FP คือ ผลบวกลวง

TN คือ ผลลบจริง

FN คือ ผลลบลวง

**ค่าความระลึก (Recall)**

ค่าความระลึก คือ ค่าวัดประสิทธิภาพของตัวแบบจำลองที่สามารถระบุผลบวกจริงได้อย่างถูกต้อง เช่น สามารถระบุได้อย่างถูกต้องว่ามีคนเป็นผู้ป่วยโควิดกี่คน

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**ค่าความแม่นยำ (Precision)**

ค่าความแม่นยำคือ อัตราส่วนระหว่างผลบวกที่แท้จริงและผลบวกทั้งหมด เช่น สามารถระบุได้ว่า ผู้ป่วยเป็นโควิดจากผู้ป่วยโควิดทั้งหมด

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

**ค่า F1-score**

ค่า F1-score คือ การรวมความแม่นยำและค่าความระลึกของตัวแบบจำลองโดยทำให้เป็นค่าเฉลี่ยฮาร์มอนิก (Harmonic Mean)

$$F1 - score = \frac{precision \cdot recall}{precision + recall}$$

**ค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิว (Mathew's correlation coefficient)**

ค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิว (Mathew's Correlation Coefficient) คือ เครื่องมือสถิติสำหรับการวัดความแตกต่างระหว่างค่าที่คาดการณ์ไว้กับค่าจริง และเทียบเท่ากับสถิติ Chi-square สำหรับตารางไขว้ $2 \times 2$ โดยมีสมการดังนี้

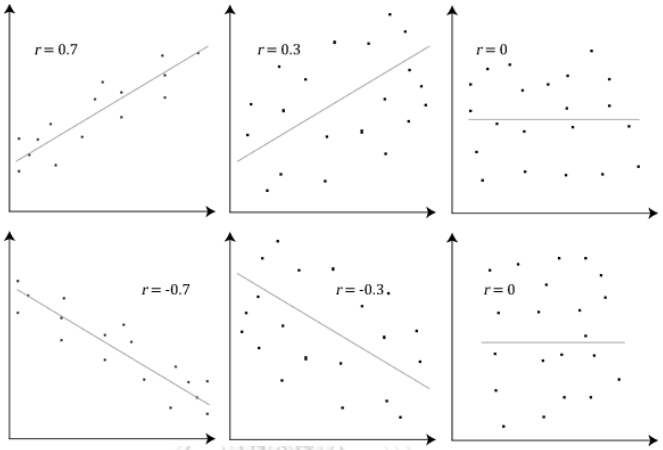
$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

โดยค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิวจะมีค่าระหว่าง +1 และ -1 ถ้ามีค่า +1 คือค่าที่ดีที่สุดระหว่างค่าที่คาดการณ์และค่าจริง แต่ถ้ามีค่า 0 แสดงว่าค่าที่ทำนายเป็นแบบสุ่ม

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

**ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและสเปียร์แมน (Pearson and Spearman Correlation Coefficients)**

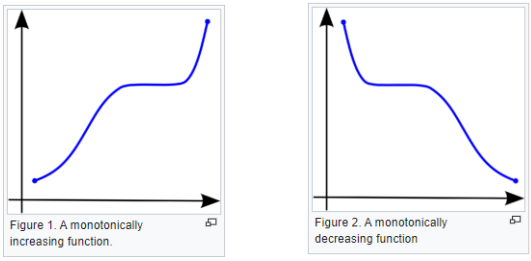
ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson correlation coefficients) คือ ตัวสถิติที่ใช้ในการวัดความสัมพันธ์เชิงเส้นระหว่างตัวแปร $X$ และ $Y$ โดยมีค่าระหว่าง +1 และ -1 ถ้าเป็นค่า +1 คือความสัมพันธ์เชิงเส้นเชิงบวก ถ้า 0 แสดงว่าไม่มีความสัมพันธ์เชิงเส้น และ -1 คือความสัมพันธ์เชิงลบทั้งหมดดังรูปที่ 12




**ภาพที่ 12 แสดงค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน**

**(ที่มา: https://en.wikipedia.org/wiki/Correlation)**

ค่าสัมประสิทธิ์สหสัมพันธ์แบบสเปียร์แมน (Spearman correlation coefficients) คือการวัดที่ไม่อิงพารามิเตอร์ของสัมพันธ์ของอันดับ โดยประเมินว่าสองตัวแปรสามารถอธิบายโดยใช้ฟังก์ชันโมโนโทนิกได้ดีเพียงใด โดยเมื่อค่าของตัวแปรหนึ่งเพิ่มขึ้น ค่าของอีกตัวแปรก็จะเพิ่มขึ้นหรือเมื่อค่าของตัวแปรหนึ่งเพิ่มขึ้น ค่าอีกตัวแปรจะลดลง ดังรูปที่ 13

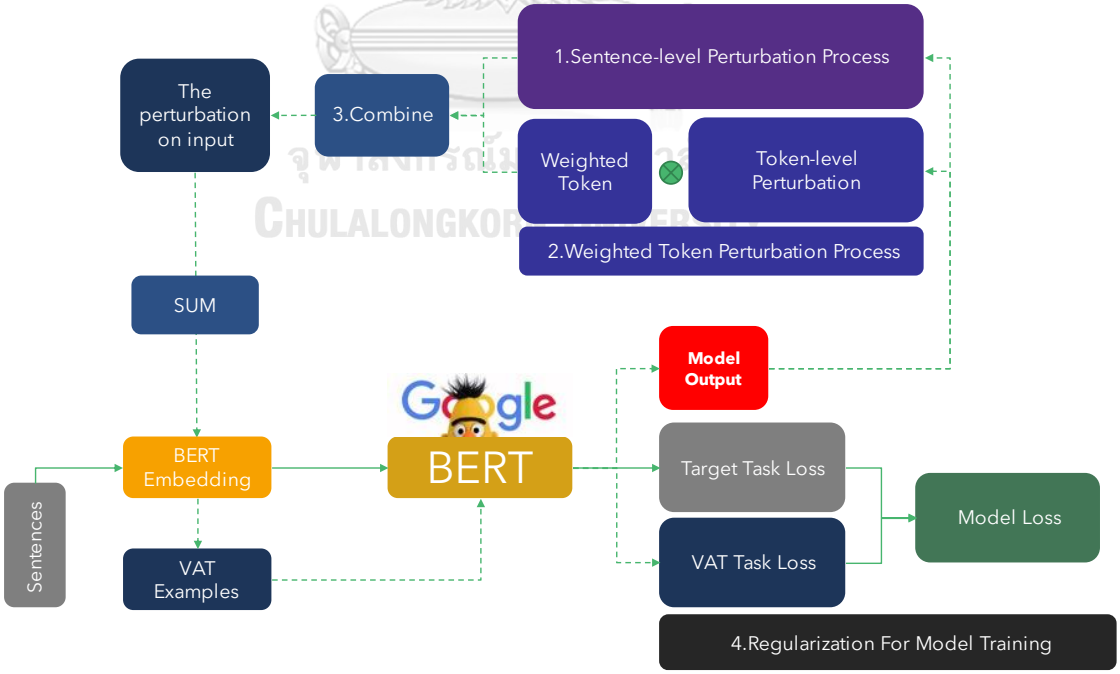



**ภาพที่ 13 แสดงค่าโมโนโทนิกฟังก์ชัน**

**(ที่มา: https://en.wikipedia.org/wiki/Monotonic_function)**

### 3.2 วิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation)

ในงานการประมวลผลภาษาธรรมชาติ แต่ละโทเค็นในประโยคมีระดับความสำคัญแตกต่างกันสำหรับการทำนายผลลัพธ์ในงานการจัดประเภทข้อความ เมื่อพิจารณาถึงข้อจำกัดของการฝึกปรปักษ์เสมือนแบบดั้งเดิมซึ่งใช้ Normalization Ball ระดับประโยคเพื่อสร้างตัวรบกวนนั้นไม่ละเอียดเพียงพอสำหรับระดับโทเค็น งานวิจัยนี้จึงเสนอ วิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) โดยขั้นตอนการดำเนินการคือ ประโยคดั้งเดิมจะถูกแปลงโดย BERT Embedding เพื่อนำเข้าสู่ตัวแบบจำลอง BERT จากนั้นเอาต์พุตของตัวแบบจำลองจะถูกนำไปเข้าสู่กระบวนการสร้างตัวรบกวนระดับประโยคและกระบวนการสร้างตัวรบกวนแบบถ่วงน้ำหนักโทเค็น ซึ่งกระบวนการสร้างตัวรบกวนแบบถ่วงน้ำหนักโทเค็นอาศัย Normalization Ball ระดับโทเค็นในการสร้างตัวรบกวนและสร้างน้ำหนักของแต่ละโทเค็น จากนั้นตัวรบกวนระดับประโยคและตัวรบกวนระดับโทเค็นจะถูกรวมเข้ากับ BERT Embedding เพื่อสร้างตัวอย่างปรปักษ์เสมือนและนำเข้าสู่ตัวแบบจำลอง BERT เพื่อใช้ในการคำนวณค่าความสูญเสียของการฝึกปรปักษ์เสมือน สุดท้ายค่าความสูญเสียของการฝึกปรปักษ์เสมือนจะถูกรวมกับค่าความสูญเสียของตัวแบบจำลอง BERT ดังรูปที่ 14




**ภาพที่ 14 แสดงภาพสถาปัตยกรรมวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น**

**3.2.1 กระบวนการรบกวนระดับประโยค (Sentence-level Perturbation Process)**

ในกระบวนการสร้างตัวรบกวนระดับประโยค เราอาศัยวิธีการแบบดั้งเดิมในการสร้างตัวรบกวน (Miyato et al., 2017) โดยตัวรบกวนระดับประโยคหาได้จากตำแหน่งของอินพุตที่ทำให้ค่าของ KL-Divergence มีค่าสูงที่สุด ซึ่งสามารถคำนวณตัวรบกวนระดับประโยคได้จากสมการดังนี้

$$r_s = \underset{\delta,\|\delta\|_\infty \leq \varepsilon}{\operatorname{argmax}} KL(f(x;\hat{\theta}), f(x+\delta;\hat{\theta}))$$

โดยที่

$x$ คือ การฝังคำของทั้งประโยค (Word Embedding)

$f(x;\hat{\theta})$ คือ ตัวแบบจำลองที่มี $\hat{\theta}$ เป็นพารามิเตอร์

$\delta$ คือ ตัวรบกวนเริ่มต้นในระดับประโยคและระดับโทเค็น

$r_s$ คือ ตัวรบกวนที่เลวร้ายสำหรับระดับประโยค

$\|\cdot\|_p$ คือ Normalization Ball ที่มี $p = \infty$

สมการดังกล่าวมีความยากในการคำนวณ ดังนั้นสำหรับโครงข่ายประสาทเทียม (Miyato et al., 2017) ใช้วิธีการแพร่ย้อนกลับ (Backpropagation) เพื่อประมาณตัวรบกวน การประมาณค่าโดยวิธีการไล่ระดับ (Gradient) $\nabla_x$ โดยเทียบกับอินพุตเพื่อให้ได้ปริมาณในการรบกวนอินพุตที่ส่งผลกับค่าความสูญเสีย สมการสำหรับประมาณค่าตัวรบกวนระดับประโยค คือ

$$r_s = g_s / \|g_s\|_\infty$$

เมื่อ $g_s = \nabla_x KL(f(x;\theta), f(x+\delta;\theta))$

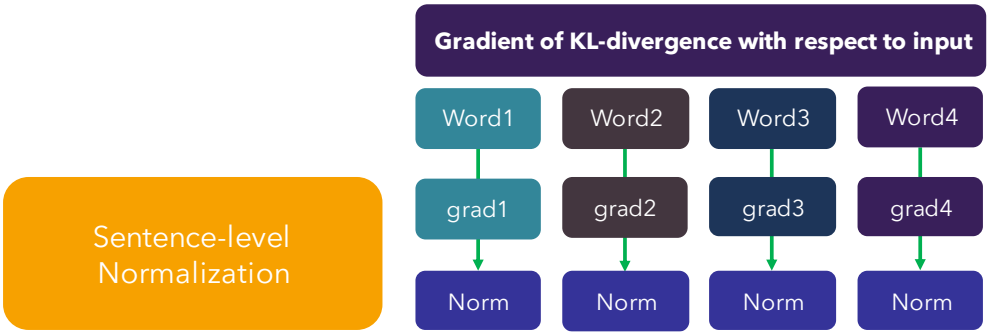
โดยที่

$g_s$ คือ Gradient KL-Divergence โดยเทียบกับอินพุต

$\|g_s\|_\infty$ คือ Normalization Ball ระดับประโยค

ในกระบวนการสร้างตัวรบกวนระดับประโยคจะใช้ Normalization Ball ระดับประโยค โดยที่จะคำนวณค่าทั้งประโยค กล่าวคือทุกโทเค็นในประโยคจะมีค่า Normalization Ball เดียวกันทุกโทเค็น ดังรูปที่ 15

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



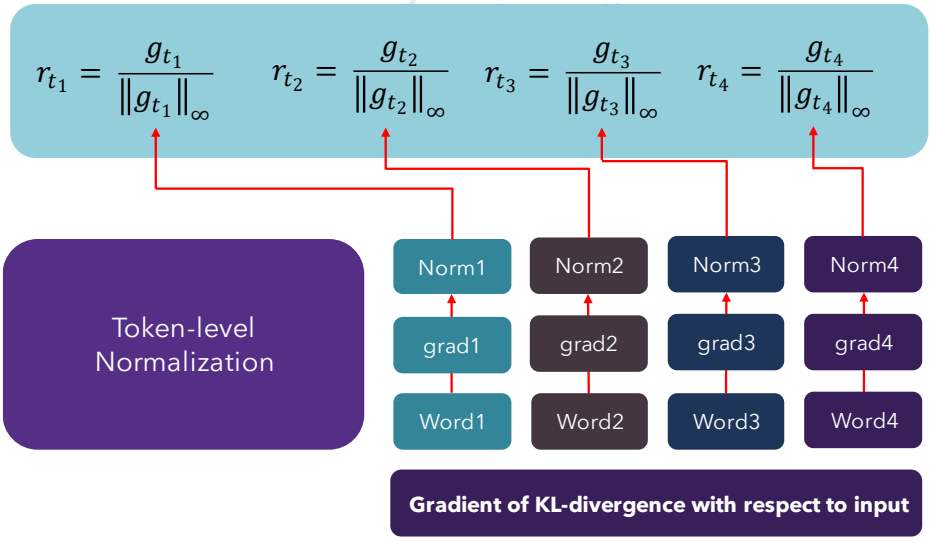

**ภาพที่ 15 อธิบายเกี่ยวกับ Normalization Ball ระดับประโยค**

**3.2.2 กระบวนการสร้างการรบกวนแบบถ่วงน้ำหนักโทเค็น (Weighted Token Perturbation)**

กระบวนการสร้างการรบกวนแบบถ่วงน้ำหนักโทเค็นประกอบไปด้วยสองส่วนสำคัญได้แก่ 1. การสร้างการรบกวนระดับโทเค็น ซึ่งจะสร้างการรบกวนของแต่ละโทเค็น 2. การถ่วงน้ำหนักโทเค็น ซึ่งสร้างน้ำหนักตามความสำคัญในประโยค จากนั้นทั้งสองส่วนจะถูกรวมเพื่อสร้างการรบกวนแบบถ่วงน้ำหนักโทเค็น

**1. การสร้างการรบกวนระดับโทเค็น (Token-level Perturbation)**

ในการสร้างตัวรบกวนระดับโทเค็น คล้ายกับการสร้างตัวรบกวนระดับประโยค แต่อาศัย Normalization Ball ระดับโทเค็น ซึ่งจะคำนวณค่า Normalization Ball ของแต่ละโทเค็นทำให้ Normalization Ball มีค่าแตกต่างกัน ดังรูปที่ 16




**ภาพที่ 16 อธิบายเกี่ยวกับ Normalization Ball ระดับโทเค็น**

สำหรับการประมาณค่าตัวรบกวนของระดับโทเค็นจะใช้วิธีการของ (Miyato et al., 2017) โดยสามารถคำนวณตามสมการ ดังนี้

$$r_{t_i} = g_{t_i} / \|g_{t_i}\|_\infty$$

เมื่อ $g_{t_i} = \nabla_x KL(f(x;\theta), f(x+\delta;\theta))$

โดยที่

$g_{t_i}$ คือ Gradient KL-Divergence เทียบกับอินพุต

$\|g_{t_i}\|_\infty$ คือ Normalization Ball ระดับโทเค็น

**2. การถ่วงน้ำหนักโทเค็น (Weighted Token)**

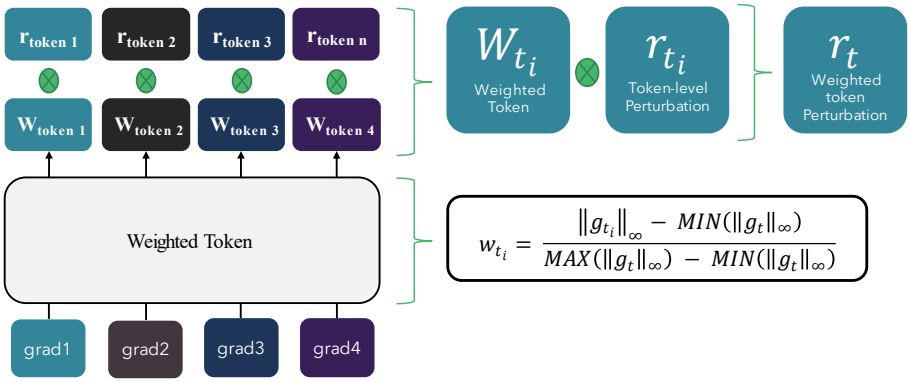
การรบกวนโทเค็นแบบถ่วงน้ำหนักมีความสำคัญในระดับโทเค็นเนื่องจากสำหรับงานด้านภาษาศาสตร์ โทเค็นแต่ละตัวมีความสำคัญต่อการทำนายที่แตกต่างกัน โทเค็นแบบถ่วงน้ำหนักจะทำหน้าที่ควบคุมผลจาก Normalization Ball ระดับโทเค็น โดยคำที่มีความสำคัญในประโยคจะมีขนาดของ Normalization Ball ใหญ่กว่าคำอื่น ๆ (Yao Qiu, 2021) โดยขั้นตอนแรกเราจะคำนวนค่า Normalization Ball ของแต่ละโทเค็นจากการไล่ระดับสี (Gradient) ของ KL-Divergence เทียบกับอินพุต จากนั้นจะนำค่าที่ได้เข้าไปคำนวณด้วยการถ่วงน้ำหนักโทเค็น ดังสมการนี้

$$w_{t_i} = \frac{\|g_{t_i}\|_\infty - MIN(\|g_t\|_\infty)}{MAX(\|g_t\|_\infty) - MIN(\|g_t\|_\infty)}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

โทเค็นที่มีการไล่ระดับสีที่ใหญ่นั้นมีความละเอียดอ่อนมากในการทำนาย (โทเค็นสำคัญ) ดังนั้นน้ำหนักของโทเค็นนี้จึงเข้าใกล้ 1 ในทางตรงกันข้ามโทเค็นที่มีการไล่ระดับที่น้อยกว่าจะมีผลกระทบเล็กน้อยต่อการทำนาย น้ำหนักของโทเค็นนั้นจะเข้าใกล้ 0 สุดท้ายน้ำหนักของโทเค็น $w_{t_i}$ จะถูกคูณด้วยการรบกวนระดับโทเค็น $r_{t_i}$ ดังรูปที่ 17 และสมการนี้

$$r_t = w_{t_i} \cdot r_{t_i}$$




**ภาพที่ 17 อธิบายกระบวนการคำนวณการถ่วงน้ำหนักโทเค็น**

### 3.2.3 กระบวนการสร้างตัวอย่างปรปักษ์เสมือน (Virtual Adversarial Examples)

ในกระบวนการนี้ จะรวมตัวรบกวนระดับประโยคและตัวรบกวนโทเค็นแบบถ่วงน้ำหนักเข้ากับ BERT Embedding ดังสมการนี้

$$\tilde{x} = \mathrm{x} + r_s + r_t$$

โดยที่

$\mathrm{x}$ คือ อินพุตของ BERT Embedding

$\tilde{x}$ คือ ตัวอย่างปรปักษ์แบบเสมือน

$r_s$ คือ ตัวรบกวนระดับประโยค

$r_t$ คือ ตัวรบกวนแบบถ่วงน้ำหนักโทเค็น

สุดท้ายตัวอย่างปรปักษ์เสมือนที่ถูกสร้างขึ้นจะถูกนำเอาตัวแบบจำลอง BERT เพื่อฝึกและคำนวณค่าความสูญเสียของ Virtual Adversarial Training โดยสามารถคำนวณจากสมการนี้

$$\mathcal{L}_{vat}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{\|\tilde{x}-\mathrm{x}\|_\infty \le \epsilon} \ell_{vat}\big(f(x;\theta), f(\tilde{x};\,\theta)\big)$$

โดยที่

$\ell_{vat}$ คือ symmetrized KL-divergence

$x$ คือ Word Embedding ของอินพุตดั้งเดิม

$\tilde{x}$ คือ Word Embedding ของอินพุตที่รวมกับตัวรบกวน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**3.3 Regularization สำหรับกระบวนการฝึกอบรมของตัวแบบจำลอง**

กระบวนการสุดท้ายคือรวมการสูญเสียของการฝึกปรปักษ์เสมือน (Virtual Adversarial Training) กับการสูญเสียเป้าหมาย (Target Loss) เพื่อให้ได้ค่าสูญเสียรวมน้อยที่สุด สมการความสูญเสียรวมคือ

$$\mathcal{L}_{total} = \mathcal{L}_{target}(\theta) + \alpha\mathcal{L}_{vat}(\theta)$$

เมื่อ $\mathcal{L}_{target}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_{target}(f(x;\theta), Y)$

โดยที่

$\mathcal{L}_{target}$ คือ ค่าความสูญเสีย Cross-Entropy

$\mathcal{L}_{vat}$ คือ ค่าความสูญเสียของวิธีการฝึกปรปักษ์เสมือน

$\alpha$ คือ ไฮเปอร์พารามิเตอร์ของกระบวนการ Regularization

สามารถดูรายละเอียดของอัลกอริทึมของวิธีการ Virtual Adversarial Training กับการรบกวนระดับโทเค็นแบบถ่วงน้ำหนักจากอัลกอริทึม 1 โดยวิธีการที่นำเสนอจะถูกสร้างขึ้นจาก ALUM ซึ่งเป็นวิธีการหาค่าการรบกวนที่ทำให้เกิดค่าความสูญเสียของวิธีการฝึกปรปักษ์สูงสุด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**อัลกอริทึม 1** Virtual Adversarial Training กับการรบกวนระดับโทเค็นแบบถ่วงน้ำหนัก

**อินพุต**: $\mathrm{T}$: the total number of iterations, $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ : The training dataset, $f(x; \theta)$ is the machine learning model parametrized by $\theta$, $K$: number of iterations for perturbation estimation, $\delta_s$: the initialization of sentence-level perturbation, $\delta_t$: the initialization of token-level perturbation $\sigma^2$: the variance of the random initialization of perturbation, $\epsilon$: perturbation bond, $\eta$: the step size of updating perturbation, $\tau$: global learning rate, $\alpha$: hyper-parameter regularization, $\prod$: the projection operation.

1: **for** $t = 1, .., T$ **do**
2: **for** $(x, y) \in S$ **do**
3: $\delta_s \sim \mathcal{N}(0, \sigma^2), \delta_t \sim \mathcal{N}(0, \sigma^2)$
4: **for** $m = 1, .., K$ **do**
5: $g_{t_i} \leftarrow \nabla_{\delta_{t_i}} KL\big(f(x; \theta), f(\mathrm{x} + \delta_s + \delta_t;\ \theta)\big)$
6: $w_{t_i} \leftarrow \frac{\|g_{t_i}\|_\infty - MIN(\|g_t\|_\infty)}{MAX(\|g_t\|_\infty) - MIN(\|g_t\|_\infty)}$
7: $r_t \leftarrow w_t \cdot \prod_{\|g_t\|_\infty < \epsilon}(\delta_t + \eta g_t)$
8: $g_s \leftarrow \nabla_{\delta_s} KL\big(f(x; \theta), f(\mathrm{x} + \delta_s + \delta_t;\ \theta)\big)$
9: $r_s \leftarrow \prod_{\|g_s\|_\infty < \epsilon}(\delta_s + \eta g_s)$
10: **end for**
11: $g_\theta \leftarrow \nabla_\theta \mathcal{L}_{target} + \alpha \nabla_\theta \mathcal{L}_{vat}\big(f(x; \theta), f(\mathrm{x} + r_s + r_t;\ \theta)\big)$
12: $\theta \leftarrow \theta - \tau g_\theta$
13: **end for**
14: **end for**

**Output:** $\theta$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 4
# การทดลองและผลการทดลอง

บทนี้จะอธิบายเกี่ยวกับการทดลองและผลการทดลองโดยมีเนื้อหาดังนี้ ระบบและเฟรมเวิร์คที่ใช้ในการสร้างวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น ชุดข้อมูลที่ใช้ในการทดลอง ตัวแบบจำลองที่ใช้ในการเปรียบเทียบประสิทธิภาพ และสุดท้าย ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของแต่ละชุดข้อมูล ซึ่งผลการทดลองจากแต่ละชุดข้อมูลจะเป็นตัวชี้วัดด้านประสิทธิภาพของวิธีการที่นำเสนอ

## 4.1 ระบบและเฟรมเวิร์คที่ใช้ในการทดลอง

สถาปัตยกรรมของวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) ถูกพัฒนาบนคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางคือ Intel Xeon CPU @2.30 GHz มีหน่วยความจำขนาด 32 GB มีหน่วยเก็บข้อมูลความจุ 124 GB หน่วยประมวลผลกราฟิกคือ NVIDIA Tesla P100 มีหน่วยความจำกราฟิก 12 GB

สถาปัตยกรรมของวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) จะใช้ภาษา Python โดยอาศัยเฟรมเวิร์ค PyTorch, Transformer – Hugging Face (Wolf et al., 2019) และ MT-DNN SMART (X. Liu, Wang, et al., 2020) เพื่อพัฒนาตัวแบบจำลองและดำเนินการทดลอง ในส่วนของตัวแบบจำลองที่พัฒนาจากต้นฉบับนั้น ใช้วิธีการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming: OOP) และสำหรับการทดลองถูกพัฒนาบนเครื่องมือ Jupyter Notebook ทั้งหมด

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

## 4.2 ชุดข้อมูลที่สำหรับการทดลอง

จากเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE) (Wang et al., 2018) มีชุดข้อมูลทั้งหมด 9 ชุด ดังที่แสดงไปในบทที่ 3 วิธีการดำเนินการงานวิจัย แต่การทดลองในงานวิจัยนี้จะไม่มีการดำเนินการทดลองในชุดข้อมูล The Winograd Schema Challenge (WNLI) เนื่องจากชุดข้อมูลดังกล่าวมีปัญหาด้านป้ายกำกับที่เป็นปรปักษ์ โดยตัวอย่างที่มีประโยคเดียวกันแต่มีป้ายกำกับที่ไม่เหมือนกัน ดังนั้นงานวิจัยนี้จะใช้ชุดการทดลอง 8 ชุด ดังตารางที่ 11

*ตารางที่ 11 แสดงรายละเอียดโดยรวมของ 8 ชุดข้อมูลที่ใช้ในการทดลอง*

| Corpus | Metrics | Train | Dev | Test |
|---|---|---|---|---|
| CoLA | Matthew corr. | 8,500 | 1,043 | 1,063 |
| SST | Accuracy | 67,349 | 872 | 1,821 |
| MRPC | Accuracy/F1 | 3,668 | 408 | 1,725 |
| QQP | Accuracy/F1 | 363,871 | 40,432 | 390,965 |
| STS-B | Pearson/Spearman corr. | 5,749 | 1,500 | 1,379 |
| MNLI | Accuracy | 392,702 | 19,647 | 19,643 |
| QNLI | Accuracy | 104,743 | 5,463 | 5,463 |
| RTE | Accuracy | 2,490 | 277 | 3,000 |

**4.3 ตัวแบบจำลองที่ใช้ในการเปรียบเทียบประสิทธิภาพ**

ในการศึกษาประสิทธิของวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) ทางผู้วิจัยจึงนำตัวแบบจำลองที่เกี่ยวข้องได้แก่ ตัวแบบจำลอง $BERT_{base}$ และตัวแบบจำลอง $SMART_{BERT}$ ที่เป็นตัวแบบจำลองล้ำสมัย เพื่อเปรียบเทียบว่าวิธีการที่นำเสนอสามารถพัฒนาตัวแบบจำลอง $BERT_{base}$ ให้มีประสิทธิภาพสูงขึ้นได้และเปรียบเทียบกับวิธีการฝึกปรปักษ์แบบดั้งเดิมอย่าง $SMART_{BERT}$ เพื่อนำเสนอว่าวิธีการที่นำเสนอสามารถปรับปรุงข้อจำกัดของการฝึกปรปักษ์แบบดั้งเดิม

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สำหรับตัวแบบจำลอง BERT โดยใช้ขนาดพื้นฐาน (Base) โดยมีสถาปัตยกรรมเลเยอร์ 12 ชั้น Hidden size ทั้งหมด 768 มี Self-Attention 12 ตัวและมีจำนวนของพารามิเตอร์ 110M ในงานวิจัยของ (Devlin, Chang, Lee, & Toutanova, 2019) นั้นรายงานผลของชุดพัฒนา (Development Set) เพียงบางชุดข้อมูลในเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป ดังนั้นทางผู้จัดทำวิจัยจึงต้องพัฒนาตัวแบบจำลอง $BERT_{base}$ ขึ้นมาใหม่ เพื่อใช้เป็นตัวแบบจำลองพื้นฐาน (Baseline Model) สำหรับผลของชุดพัฒนา ในส่วนของชุดทดสอบ (Test set) ทางผู้จัดทำวิจัยใช้ผลของ (Devlin, Chang, Lee, & Toutanova, 2019) เพื่อเปรียบเทียบประสิทธิภาพ ซึ่งมีความน่าเชื่อถือและเป็นที่ยอมรับในวงการ

สำหรับตัวแบบจำลอง $SMART_{BERT}$ เป็นแบบจำลอง $BERT_{base}$ ที่ใช้วิธีการฝึกปรปักษ์เสมือน SMART เพื่อปรับปรุงประสิทธิภาพและแก้ไขปัญหา Overfitting ซึ่งในผู้จัดทำการวิจัยใช้เป็นตัวแทน

ของวิธีการฝึกปรปักษ์เสมือนแบบดั้งเดิม โดยตัวแบบจำลอง SMART ถูกใช้เปรียบเทียบในชุดข้อมูลพัฒนา (Development Set) ซึ่งผลถูกนำมาจากงานวิจัยของ SMART

**4.4 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของแต่ละชุดข้อมูล**

ในการวัดประสิทธิภาพวิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) จะมีค่าไฮเปอร์พารามิเตอร์ที่แตกต่างกันในแต่ละชุดข้อมูล โดยวิธีการปรับค่าไฮเปอร์พารามิเตอร์นี้ เราอาศัยวิธีการสุ่มแต่ละค่าและปรับค่าแต่ละตัวจนได้ผลลัพธ์ที่ดี

**4.4.1 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Corpus of Linguistic Acceptability (CoLA)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล CoLA

*ตารางที่ 12 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล CoLA*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00001 |
| ค่า Epochs | 6 |
| ค่า $\alpha$ Regularization term | 0.2 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล CoLA

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล CoLA มีข้อมูลสำหรับฝึกทั้งหมด 8,500 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 1,043 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 1,063 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิว (Mathew's Correlation Coefficient)

*ตารางที่ 13 แสดงผลการทดลองบนชุดข้อมูล CoLA*

| Model | CoLa Mathew's Correlation Coefficient |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 54.7 |
| $SMART_{BERT}$ | 59.1 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **60.9** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 52.1 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **56.0** |

จากตารางที่ 13 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 6.2 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิวที่ร้อยละ 60.9 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 1.8

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 3.9 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์ของแมทธิวที่ร้อยละ 56

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**4.4.2 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Stanford Sentiment Treebank (SST-2)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล SST-2

*ตารางที่ 14 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล SST-2*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 5 |
| ค่า $\alpha$ Regularization term | 0.1 |

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล SST-2

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล SST-2 มีข้อมูลสำหรับฝึกทั้งหมด 67,349 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 872 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 1,821 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy)

*ตารางที่ 15 แสดงผลการทดลองบนชุดข้อมูล SST-2*

| Model | **SST-2 Accuracy** |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 92.9 |
| $SMART_{BERT}$ | 93 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **93.6** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 93.5 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **94.4** |

จากตารางที่ 15 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.7 โดยได้รับค่าความถูกต้องที่ร้อยละ 93.6 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.6

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.9 โดยได้รับค่าความถูกต้องที่ร้อยละ 94.4

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

### 4.4.3 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Microsoft Research Paraphrase Corpus (MRPC)

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล MRPC

*ตารางที่ 16 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล MRPC*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 5 |
| ค่า $\alpha$ Regularization term | 0.1 |

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล MRPC

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล MRPC มีข้อมูลสำหรับฝึกทั้งหมด 3,668 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 408 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 1,725 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy) และค่า F1-score

จากตารางที่ 17 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 4.1 โดยได้รับค่าความถูกต้องที่ร้อยละ 88.2 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.5 ในส่วนของ F1-score วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 2.8 โดยได้รับค่า F1-score ที่ร้อยละ 91.8 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.5

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.7 โดยได้รับค่าความถูกต้องที่ร้อยละ 85.5 สำหรับในส่วนของ F1-score วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.3 โดยได้รับค่า F1-score ที่ร้อยละ 89.2

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

*ตารางที่ 17 แสดงผลการทดลองบนชุดข้อมูล MRPC*

| Model | MRPC Accuracy/F1 |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 84.1/89 |
| $SMART_{BERT}$ | 87.7/91.3 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **88.2/91.8** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 84.8/88.9 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **85.5/89.2** |

**4.4.4 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Quora Question Pairs (QQP)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล QQP

*ตารางที่ 18 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล QQP*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 3 |
| ค่า $\alpha$ Regularization term | 0.2 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล QQP

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล QQP มีข้อมูลสำหรับฝึกทั้งหมด 363,871 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 40,432 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 390,965 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy) และค่า F1-score

*ตารางที่ 19 แสดงผลการทดลองบนชุดข้อมูล QQP*

| Model | QQP Accuracy/F1 |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 90.9/88.3 |
| $SMART_{BERT}$ | 91.5/88.5 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **91.6/88.6** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 89.2/71.2 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **89.8/72.9** |

จากตารางที่ 19 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 1.3 โดยได้รับค่าความถูกต้องที่ร้อยละ 91.6 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.1 ในส่วนของ F1-score วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.3 โดยได้รับค่า F1-score ที่ร้อยละ 88.6 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.1

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.6 โดยได้รับค่าความถูกต้องที่ร้อยละ 89.8 สำหรับในส่วนของ F1-score วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.7 โดยได้รับค่า F1-score ที่ร้อยละ 72.9

Chulalongkorn University

**4.4.5 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Semantic Textual Similarity Benchmark (STS-B)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล STS-B

*ตารางที่ 20 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล STS-B*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 5 |
| ค่า $\alpha$ Regularization term | 0.2 |

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล STS-B

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล STS-B มีข้อมูลสำหรับฝึกทั้งหมด 5,749 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 1,500 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 1,379 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันและสเปียร์แมน (Pearson and Spearman Correlation Coefficients)

*ตารางที่ 21 แสดงผลการทดลองบนชุดข้อมูล STS-B*

| **Model** | **STS-B P/S Corr.** |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 89.2/88.8 |
| $SMART_{BERT}$ | 90.0/89.4 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **90.2/89.8** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 87.1/85.8 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **87.3/86.2** |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 21 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 1.0 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันที่ร้อยละ 90.2 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.2 ในส่วนของ ค่าสัมประสิทธิ์สหสัมพันธ์แบบเปียร์แมน วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 1.0 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์แบบเปียร์แมนที่ร้อยละ 89.8 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.4

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.2 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์แบบเพียร์สันที่ร้อยละ 87.3 สำหรับในส่วนของค่าสัมประสิทธิ์สหสัมพันธ์แบบเปียร์แมน วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.4 โดยได้รับค่าสัมประสิทธิ์สหสัมพันธ์แบบเปียร์แมนที่ร้อยละ 86.2

**4.4.6 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Multi-Genre Natural Language Inference Corpus (MNLI)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล MNLI

*ตารางที่ 22 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล MNLI*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 3 |
| ค่า $\alpha$ Regularization term | 0.1 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล MNLI

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล MNLI มีข้อมูลสำหรับฝึกทั้งหมด 392,702 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 2 ชุด ได้แก่ MNLI-Matched 9,815 และตัวอย่าง MNLI-Mismatched 9,832 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 2 ชุด ได้แก่ MNLI-Matched 9,796 และตัวอย่าง MNLI-Mismatched 9,847 ตัวอย่างเครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy)

*ตารางที่ 23 แสดงผลการทดลองบนชุดข้อมูล MNLI*

| Model | MNLI-m Accuracy | MNLI-mm Accuracy |
|---|---|---|
| **ชุดทวนสอบ (Development Set)** | | |
| $BERT_{base}$ | 84.5 | 84.4 |
| $SMART_{BERT}$ | **85.6** | **86.0** |
| Virtual Adversarial Training with the Weighted Token Perturbation | 85.2 | 85.7 |
| **ชุดทดสอบ (Test Set)** | | |
| $BERT_{base}$ | 84.6 | 83.4 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **85.5** | **84.8** |

จากตารางที่ 23 ในชุดข้อมูลสำหรับทวนสอบแบบ MNLI-m (Matched) วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.7 โดยได้รับค่าความถูกต้องที่ร้อยละ 85.2 และเมื่อเปรียบเทียบกับวิธีการของ SMART พบว่าวิธีการ SMART มีประสิทธิภาพที่ดีกว่าวิธีการที่นำเสนอร้อยละ 0.4

ในชุดข้อมูลสำหรับทวนสอบแบบ MNLI-mm (Mismatched) วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 1.6 โดยได้รับค่าความถูกต้องที่ร้อยละ 85.7 และเมื่อเปรียบเทียบกับวิธีการของ SMART พบว่าวิธีการ SMART มีประสิทธิภาพที่ดีกว่าวิธีการที่นำเสนอร้อยละ 0.3

ในส่วนของชุดทดสอบแบบ MNLI-m (Matched) วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.9 โดยได้รับค่าความถูกต้องที่ร้อยละ 85.5 เมื่อทดลองด้วยชุดข้อมูลทดสอบแบบ MNLI-mm (Mismatched) วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 1.4 โดยได้รับค่าความถูกต้องที่ร้อยละ 84.8

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

### 4.4.7 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Stanford Question Answering (QNLI)

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล QNLI

*ตารางที่ 24 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล QNLI*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00003 |
| ค่า Epochs | 3 |
| ค่า $\alpha$ Regularization term | 0.2 |

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล QNLI

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล QNLI มีข้อมูลสำหรับฝึกทั้งหมด 104,743 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 5,463 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 5,463 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy)

*ตารางที่ 25 แสดงผลการทดลองบนชุดข้อมูล QNLI*

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

| **Model** | **QNLI Accuracy** |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 91.1 |
| $SMART_{BERT}$ | 91.7 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **92.0** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 90.5 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **91.4** |

จากตารางที่ 25 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.9 โดยได้รับค่าความถูกต้องที่ร้อยละ 92.0 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 0.3

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 0.9 โดยได้รับค่าความถูกต้องที่ร้อยละ 91.4

**4.4.8 ค่าไฮเปอร์พารามิเตอร์และผลการทดลองของชุดข้อมูล The Recognizing Textual Entailment (RTE)**

1) ค่าไฮเปอร์พารามิเตอร์ของตัวแบบจำลองสำหรับชุดข้อมูล RTE

*ตารางที่ 26 แสดงค่าไฮเปอร์พารามิเตอร์สำหรับชุดข้อมูล RTE*

| ค่าไฮเปอร์พารามิเตอร์ | ค่า |
|---|---|
| ค่า $\eta$ the step size of updating perturbation | 0.001 |
| ค่า Dropout | 0.1 |
| ค่า Learning Rate | 0.00005 |
| ค่า Epochs | 6 |
| ค่า $\alpha$ Regularization term | 0.1 |

2) ผลการทดลองของตัวแบบจำลองบนชุดข้อมูล RTE

การทดลองนี้ได้ทำการเปรียบเทียบตัวแบบจำลองทั้งหมด 3 ตัวแบบจำลอง โดยชุดข้อมูล RTE มีข้อมูลสำหรับฝึกทั้งหมด 2,490 ตัวอย่าง มีข้อมูลสำหรับทวนสอบ 277 ตัวอย่าง และมีข้อมูลสำหรับทดสอบ 3,000 ตัวอย่าง เครื่องมือวัดประสิทธิภาพของข้อมูลชุดนี้คือ ค่าความถูกต้อง (Accuracy)

*ตารางที่ 27 แสดงผลการทดลองบนชุดข้อมูล RTE*

| Model | RTE Accuracy |
|---|---|
| **ชุดทวนสอบ (Development Set)** | |
| $BERT_{base}$ | 63.5 |
| $SMART_{BERT}$ | 71.2 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **72.6** |
| **ชุดทดสอบ (Test Set)** | |
| $BERT_{base}$ | 66.4 |
| Virtual Adversarial Training with the Weighted Token Perturbation | **70.4** |

จากตารางที่ 27 ในชุดข้อมูลสำหรับทวนสอบ วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าตัวแบบจำลอง BERT ที่ร้อยละ 9.1 โดยได้รับค่าความถูกต้องที่ร้อยละ 72.6 และเมื่อเปรียบเทียบกับวิธีการของ SMART วิธีการที่นำเสนอมีประสิทธิภาพที่ดีกว่าร้อยละ 1.4

ในส่วนของชุดทดสอบ วิธีการที่นำเสนอให้ประสิทธิภาพที่สูงกว่าตัวแบบจำลอง BERT ที่ร้อยละ 4.0 โดยได้รับค่าความถูกต้องที่ร้อยละ 70.4

### 4.5 วิเคราะห์ความสำคัญขององค์ประกอบของการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นประกอบไปด้วย 2 องค์ประกอบหลักที่สำคัญ ได้แก่ 1. การฝึกปรปักษ์เสมือนด้วยกันรบกวนสองระดับ (ระดับประโยคและระดับโทเค็น) และ 2. การถ่วงน้ำหนักโทเค็น (Weighted Token) เพื่อทดลองว่าองค์ประกอบทั้งสองส่งผลกับประสิทธิภาพ ผู้จัดทำวิจัยได้นำส่วนประกอบการถ่วงน้ำหนักโทเค็นออกจากวิธีการรบกวนสองระดับ (Two-level Perturbation) หรือก็คือการรวมการรบกวนระดับประโยคและระดับโทเค็นเข้าด้วยกัน และใช้ SMART เพื่อเป็นตัวแทนของการรบกวนหนึ่งระดับ (Single-level Perturbation) หรือก็คือ การรบกวนระดับประโยคเพียงอย่างเดียว (Only Sentence-level Perturbation) โดยทำการทดลองบนเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป ทั้งหมด 5 ชุดข้อมูล

*ตารางที่ 28 ผลการทดลององค์ประกอบสำคัญบน 5 ชุดข้อมูล*

| Model | RTE Acc | QNLI Acc | MRPC Acc/F1 | CoLA Mcc | SST Acc |
|---|---|---|---|---|---|
| **การรบกวนหนึ่งระดับ (Single-level perturbation)** | | | | | |
| การรบกวนระดับประโยคเพียงอย่างเดียว | 71.2 | 91.7 | 87.7/91.3 | 59.1 | 93.0 |
| **การรบกวนสองระดับ (Two-level perturbations)** | | | | | |
| *ไม่มี*การถ่วงน้ำหนักโทเค็น | 72.2 | 91.9 | **88.5**/91.6 | 59.9 | 93.5 |
| *มี*การถ่วงน้ำหนักโทเค็น (Weighed token) | **72.6** | **92.0** | 88.2/**91.8** | **60.9** | **93.6** |

จากตารางที่ 28 แสดงผลการทดลองการนำองค์ประกอบบางส่วนออกจากวิธีการที่นำเสนอ โดยทำการทดสอบบน 5 ชุดข้อมูลโดยใช้ชุดทวนสอบ (Development Set) จากการทดลองพบว่าในชุดข้อมูล QNLI วิธีการรบกวนสองระดับที่มีการถ่วงน้ำหนักโทเค็นมีประสิทธิภาพสูงกว่าวิธีการวิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นร้อยละ 0.1 และมีประสิทธิภาพสูงกว่าวิธีการรบกวนหนึ่งระดับร้อยละ 0.3 สำหรับชุดข้อมูล SST วิธีการรบกวนสองระดับที่*มี*การถ่วงน้ำหนักโทเค็นมีประสิทธิภาพสูงกว่าวิธีการวิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นร้อยละ 0.1 ในชุดข้อมูล CoLA วิธีการรบกวนสองระดับที่มีการถ่วงน้ำหนักโทเค็นมีประสิทธิภาพสูงกว่าวิธีการวิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นร้อยละ 1.0 ในชุดข้อมูล RTE วิธีการรบกวนสองระดับที่มีการถ่วงน้ำหนักโทเค็นมีประสิทธิภาพสูงกว่าวิธีการวิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นร้อยละ 0.4 สำหรับชุดข้อมูล MRPC วิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นมีค่าความถูกต้องสูงกว่าวิธีการวิธีการรบกวนสองระดับที่มีการถ่วงน้ำหนักโทเค็นร้อยละ 0.3 แต่ในส่วนของ F1-score วิธีการรบกวนสองระดับที่มีการถ่วงน้ำหนักโทเค็นมีค่าความถูกต้องสูงกว่าวิธีการวิธีการรบกวนสองระดับที่ไม่มีการถ่วงน้ำหนักโทเค็นร้อยละ 0.2

# บทที่ 5
# สรุปผลการวิจัยและข้อเสนอแนะ

## 5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้เสนอการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) สำหรับงานการจัดประเภทข้อความ โดยได้ดำเนินการทดลองบนเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE) โดยมีทั้งหมด 8 ชุดข้อมูลที่มีภารกิจและตัววัดผลแตกต่างกัน เพื่อศึกษาและเปรียบเทียบวิธีการที่นำเสนอกับตัวแบบจำลองที่ล้ำสมัยได้แก่ ตัวแบบจำลอง $BERT_{base}$ และตัวแบบจำลอง $SMART_{BERT}$

ในการศึกษาและเปรียบเทียบประสิทธิภาพ ผู้วิจัยได้แบ่งการทดลองเป็น 2 ส่วน ได้แก่ 1. การทดลองเพื่อเปรียบเทียบประสิทธิภาพโดยการทดลองนี้จะเปรียบเทียบผลลัพธ์จากตัวแบบจำลอง $BERT_{base}$, ตัวแบบจำลอง $SMART_{BERT}$ และวิธีการที่นำเสนอ 2. การทดลองเพื่อศึกษาความสำคัญขององค์ประกอบของการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นโดยผู้วิจัยได้นำบางส่วนประกอบนำเสนอออกจากวิธีการที่นำเสนอเพื่อวัดประสิทธิภาพเมื่อองค์ประกอบถูกนำออกไป

จากผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพด้วยเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป (General Language Understanding Evaluation: GLUE) ทั้งหมด 8 ชุดข้อมูล พบว่า จากชุดข้อมูลสำหรับทวนสอบ (Development Set) พบว่า วิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็น (Virtual Adversarial Training with the Weighted Token Perturbation) มีประสิทธิภาพในการทำนายดีกว่าตัวแบบจำลอง $BERT_{base}$ ใน 8 ชุดข้อมูลของเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไป และเมื่อเปรียบเทียบกับตัวแบบจำลอง $SMART_{BERT}$ วิธีการฝึกปรปักษ์เสมือนกับการรบกวนโทเค็นแบบถ่วงน้ำหนักมีประสิทธิภาพเหนือกว่าตัวแบบจำลอง $SMART_{BERT}$ 7 ชุดข้อมูลจากทั้งหมด 8 ชุดข้อมูล จากชุดข้อมูลสำหรับทดสอบ (Test Set) วิธีการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นมีประสิทธิภาพในการทำนายดีกว่าตัวแบบจำลอง $BERT_{base}$ ในทุก ชุดข้อมูล โดยสามารถเพิ่มคะแนนความถูกต้องเฉลี่ยในระบบของเกณฑ์มาตรฐานการประเมินความเข้าใจภาษาทั่วไปจากร้อยละ 78.3 เป็น 79.5 และสำหรับชุดข้อมูลขนาดเล็กวิธีการที่นำเสนอสามารถแก้ไขปัญหา Overfitting และเพิ่มประสิทธิภาพของตัวแบบจำลอง BERT ได้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จากผลการทดลองเพื่อศึกษาความสำคัญขององค์ประกอบของการฝึกปรปักษ์เสมือนด้วยการรบกวนแบบถ่วงน้ำหนักโทเค็นแสดงให้เห็นว่าการรบกวนสองระดับและโทเค็นที่ถ่วงน้ำหนักมีความสำคัญต่อกันและกัน ถ้ามีการนำส่วนประกอบบางส่วนออกจะทำให้ประสิทธิภาพของวิธีการที่นำเสนอลดลง ที่สำคัญกว่านั้น แนวทางทั้งหมดของเราสามารถปรับปรุงประสิทธิภาพและดีกว่าการรบกวนระดับเดียว (การรบกวนระดับประโยคเท่านั้น) ในชุดข้อมูล 5 ชุดบน GLUE

**5.2 ข้อเสนอแนะ**

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพและแก้ไขปัญหา Overfitting สำหรับตัวแบบจำลอง BERT สำหรับงานการจัดประเภทข้อความ และการทดลองแสดงให้เห็นแล้วว่าวิธีการดังกล่าวสามารถช่วยปรับปรุงประสิทธิภาพและแก้ไขปัญหา Overfitting ได้ แต่วิธีการดังกล่าวสามารถกับใช้ได้เฉพาะกับตัวแบบจำลอง BERT เท่านั้นและยังมีตัวแบบจำลองจำนวนมากที่ยังพบกับปัญหา Overfitting ถ้าสามารถปรับแต่งให้วิธีการดังกล่าวสามารถปรับใช้กับตัวแบบจำลองอื่นได้จะสามารถทำให้เกิดประโยชน์ในวงการด้านภาษาศาสตร์

ในส่วนของทรัพยากรที่ใช้ในการคำนวณ วิทยานิพนธ์ฉบับนี้ไม่ได้วัดผลการดำเนินการด้านเวลาที่ใช้ในคำนวณระหว่างวิธีที่นำเสนอกับวิธีการแบบดั้งเดิม ดังนั้นอาจจะเป็นประเด็นสำหรับการศึกษาและทดลองในอนาคต แต่ทางผู้วิจัยคาดว่า เวลาที่ใช้ในการคำนวณระหว่างวิธีการดั้งเดิมอาจจะไม่แตกต่างจากวิธีการที่นำเสนออย่างมีนัยสำคัญ

วิธีการที่นำเสนอถูกออกแบบมาเพื่องานจัดประเภทข้อความเท่านั้น แต่มีหลายโดเมนของงานด้านภาษาศาสตร์ เช่น การรู้จำเอนทิตีที่มีชื่อ (Named Entity Recognition) และการแปลด้วยเครื่อง (Machine Translation) ซึ่งยังพบปัญหา Overfitting ในอนาคต ผู้วิจัยจะปรับใช้วิธีการฝึกปรปักษ์เสมือนกับการรบกวนโทเค็นที่ถ่วงน้ำหนักเพื่อคาดการณ์คำหรือประโยคที่ถูกต้องแม่นยำในโดเมนเหล่านี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

# บรรณานุกรม


Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). *Lump at SemEval-2017 Task 1: Towards an Interlingua Semantic Similarity.* Paper presented at the Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).

Dagan, I., Glickman, O., & Magnini, B. (2005). *The PASCAL Recognising Textual Entailment Challenge.* Paper presented at the MLCW.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Paper presented at the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).

Dolan, W. B., & Brockett, C. (2005). *Automatically Constructing a Corpus of Sentential Paraphrases.* Paper presented at the IJCNLP.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *CoRR, abs/1412.6572*.

Gunel, B., Du, J., Conneau, A., & Stoyanov, V. (2021). Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. *International Conference on Learning Representations, abs/2011.01403*.

Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Zhao, T. (2020). *SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization.* Paper presented at the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Joyce, J. (2011). Kullback-Leibler Divergence. In *International Encyclopedia of Statistical Science* (pp. 720-722).

Karimi, A., Rossi, L., Prati, A., & Full, K. (2021). Adversarial Training for Aspect-Based Sentiment Analysis with BERT. *2020 25th International Conference on Pattern Recognition (ICPR)*, 8797-8803.

Kobayashi, S. (2018). *Contextual Augmentation: Data Augmentation by Words with*


จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

*Paradigmatic Relations.* Paper presented at the Proceedings of NAACL-HLT 2018.

Kou, X., Yang, Y., Wang, Y., Zhang, C., Chen, Y., Tong, Y., . . . Bai, J. (2021). Improving BERT With Self-Supervised Attention. *IEEE Access, 9*, 144129-144139.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *International Conference on Learning Representations*.

Li, L., & Qiu, X. (2021). *Token-Aware Virtual Adversarial Training in Natural Language Understanding.* Paper presented at the AAAI.

Liu, X., Cheng, H., He, P., Chen, W., Wang, Y., Poon, H., & Gao, J. (2020). Adversarial Training for Large Neural Language Models. *ArXiv, abs/2004.08994*.

Liu, X., Wang, Y., Ji, J., Cheng, H., Zhu, X., Awa, E., . . . Gao, J. (2020). *The Microsoft Toolkit of Multi-Task Deep Neural Networks for Natural Language Understanding.* Paper presented at the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., . . . Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *International Conference on Learning Representations*.

Miyato, T., Dai, A. M., & Goodfellow, I. J. (2017). Adversarial Training Methods for Semi-Supervised Text Classification. *international conference on learning representations, (2017)*.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Pereira, L., Liu, X., Cheng, F., Asahara, M., & Kobayashi, I. (2020). *Adversarial Training for Commonsense Inference.* Paper presented at the Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020).

Pereira, L., Liu, X., Cheng, H., Poon, H., Gao, J., & Kobayashi, I. (2021). *Targeted Adversarial Training for Natural Language Understanding.* Paper presented at the NAACL.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text.* Paper presented at the EMNLP.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *EMC^2: 5th Edition Co-located with*

*NeurIPS'19*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.* Paper presented at the EMNLP.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.* Paper presented at the Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP.

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics, 7*, 625-641. doi:10.1162/tacl_a_00290

Wei, J., & Zou, K. (2019). *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks.* Paper presented at the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.

Williams, A., Nangia, N., & Bowman, S. R. (2018). *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference.* Paper presented at the NAACL.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., . . . Brew, J. (2019). *Transformers: State-of-the-art Natural Language Processing.* Paper presented at the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Wu, X., Lv, S., Zang, L., Han, J., & Hu, S. (2019). *Conditional BERT Contextual Augmentation.* Paper presented at the ICCS.

Yao Qiu, J. Z., and Jie Zhou. (2021). *Improving Gradient-based Adversarial Training for Text Classification by Contrastive Learning and Auto-Encoder.* Paper presented at the In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.

Zhou, M., Li, Z., & Xie, P. (2021). Self-supervised Regularization for Text Classification. *Transactions of the Association for Computational Linguistics, 9*, 641-656. doi:10.1162/tacl_a_00389

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., & Liu, J. (2020). FreeLB: Enhanced

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Adversarial Training for Natural Language Understanding. *International Conference on Learning Representations*.



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# ประวัติผู้เขียน

| | |
|---|---|
| **ชื่อ-สกุล** | ธีรพงศ์ แซ่ลิ้ม |
| **วัน เดือน ปี เกิด** | 29 มิถุนายน 2539 |
| **สถานที่เกิด** | นครศรีธรรมราช |
| **วุฒิการศึกษา** | วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมการจัดการและโลจิสติกส์ มหาวิทยาลัยศิลปากร |
| **ที่อยู่ปัจจุบัน** | 131 ถนนกะโรม ตำบลโพธิ์เสด็จ อำเภอเมือง จังหวัดนครศรีธรรมราช 80000 |

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University