**Report Midterm Data 2110531 Data Science and Data Engineering Tools**

**Chapter 1: Introduction**

ปัจจุบันในยุคของเทคโนโลยีข้อมูลส่วนใหญ่มักเก็บอยู่ในรูปแบบออนไลน์เพื่อง่ายต่อการเข้าถึงและจัดเก็บ อย่างไรก็ตามเมื่อข้อมูลมีมากเกินกว่าดึงมาใช้งานได้อย่างสะดวก การจัดกลุ่มข้อมูลให้เป็นระเบียบก็เป็นสิ่งสำคัญ โดยในโจทย์ปัญหานี้ที่เป็นการจัดกลุ่มงานวิจัยในสาขาของวิศวกรรมศาสตร์รวมทั้งหมด 18 สาขาประกอบด้วยวิศวกรรมโยธา วิศวกรรมสิ่งแวดล้อม วิศวกรรมชีวการแพทย์ วิศวกรรมน้ำมัน วิศวกรรมโลหะวิศวกรรมเครื่องกล วิศวกรรมไฟฟ้า วิศวกรรมคอมพิวเตอร์ วิศวกรรมออพติก วิศวกรรมนาโน วิศวกรรมเคมี วิศวกรรมวัสดุ วิศวกรรมเกษตร วิศวกรรมการศึกษา วิศวกรรมอุตสาหกรรม วิศวกรรมความปลอดภัย"คณิตศาสตร์และสถิติ" ในประเด็นของวิศวกรรม และวิทยาศาสตร์วัสดุ 454 งานวิจัยสำหรับการให้เรียนรู้แบบจำลอง ซึ่งแต่ละงานวิจัยจะประกอบด้วยหัวข้อเรื่องและบทคัดย่อที่ถูกจัดกลุ่มอยู่ในสาขาวิศวกรรมอย่างน้อย 1 สาขาจึงทำให้โจทย์ปัญหานี้เป็นโจทย์ปัญหาแบบ multi-label และ multi-class text classification โดยการจัดกลุ่มงานวิจัยให้ถูกต้องต้องมีการเตรียมข้อมูลที่เหมาะแก่ลักษณะของข้อมูลและนำไปทำนายด้วยแบบจำลองที่เหมาะสมควบคู่ไปกับการปรับพารามิเตอร์ที่เหมาะสมจากการแบ่งส่วนให้การเรียนรู้เพื่อการประเมินก่อนไปตรวจสอบกับข้อมูลส่วนทดสอบ

A screenshot of a phone

Description automatically generated

รูปที่ 1 แสดงการสารบัญของการทำการทดลอง

**Chapter 2: Data preparation**

1. เนื่องจากข้อมูลที่มีในส่วนของข้อมูลขาเข้าหรือประโยคที่จะนำเข้าแบบจำลองเพื่อการเรียนรู้ใช้ได้ทั้งสองColumn จึงรวม Column Title และ Abstract เป็น Column ชื่อ Text และทิ้งแถวที่ซ้ำ

A screenshot of a computer

Description automatically generated

รูปที่ 2 แสดงตัวอย่างข้อมูล Text จากการรวม Title และ Abstract

1. นำข้อมูล Text ที่ได้มากำจัดคำ Stop word ด้วยการเรียกแหล่งเก็บ Stop word ภาษาอังกฤษของ nltk มาใช้พร้อมทั้งเพิ่มเติม Stop word บางคำที่มักเจอในงานวิจัย เช่น คำระบุอย่างจำนวน one, two, three คำเชื่อมอื่นๆอย่าง among, beside เป็นต้น

A screenshot of a computer

Description automatically generated

รูปที่ 3 แสดงตัวอย่างโค้ดการตัด Stop word

1. สร้างฟังก์ชันเพื่อนำไปใช้กับข้อมูล Text โดยมีการใช้คำสั่ง compile เพื่อกำหนดการจับคู่เพื่อการตัด stop word และใช้คำสั่ง sub เพื่อเรียกรูปแบบการจับคู่ที่ compile ไว้มาใช้ตัดคำ stop word จากนั้นก็นำฟังก์ชันมา map กับข้อมูล Text
2. สร้างฟังก์ชัน text\_preprocessing เพื่อจัดการกับข้อมูลให้เรียบร้อยมากขึ้นด้วยการใช้คำสั่ง sub เป็นหลักในการแปลคำย่อ เช่น I’m เป็น I am หรือลบอักขระพิเศษ เพื่อให้ข้อมูลสะอาดและแบบจำลองสามารถวิเคราะห์ได้ดีขึ้น
3. เนื่องจาก labels ยังคงอยู่ในรูปของ list จึงต้องนำข้อมูลมาแปลงเป็นอยู่ในรูป Multi label ด้วยการใช้คำสั่ง fit\_transfrom ด้วย MultiLabelBinarizer() และนำผลที่ได้ไปรวมกับ dataframe เก่าโดยตั้งชื่อและละ label ตามคลาสต่างๆ

A screenshot of a computer

Description automatically generated

รูปที่ 4 แสดงตัวอย่างผลการแสดงการแปลงข้อมูลเป็น one hot label

1. นำข้อมูลชื่อคลาสต่างๆมาแปลงเป็น dict แบบตัวเลขเป็นชื่อคลาสและคลาสเป็นชื่อตัวเลขเพื่อเหมาะแก่การนำไปแปลผลหลังจากได้ผลลัพธ์จากแบบจำลอง

A screenshot of a computer

Description automatically generated

รูปที่ 5 แสดงตัวอย่างข้อมูลสุดท้ายหลังการทำ Preprocessing

**Chapter 3: Model**

1. เริ่มต้นจากการแบ่งข้อมูลเพื่อการเรียนรู้ (train) และปรับพารามิเตอร์ (validation) โดยการแบ่ง train : validation เป็น 8 : 2
2. จากนั้นนำข้อมุลที่ได้มารวมเป็น dict ด้วยคำสั่ง dataset เพื่อง่ายต่อการนำมาปรับเปลี่ยน หรือการ map เข้ากับฟังก์ชันแปลงเป็น token ตามที่รูปแบบที่ต้องการก่อนนำไปเข้าแบบจำลอง

A screenshot of a computer

Description automatically generatedรูปที่ 6 แสดงผลการรวมเป็น dict

1. สร้างฟังก์ชัน preprocess\_data ซึ่งเป็นกระบวนการแปลงจากคำเป็น token โดยคำสั่ง tokenizer และกำหนด labels ของแต่ละ text แบบรวมเป็น column เดียวเพื่อการนำไปใช้
2. นำฟังก์ชัน preprocess\_data มา map เข้ากับข้อมูล text เก็บในตัวแปล encoded\_dataset จากนั้นก็ทิ้ง colume อื่นๆ ที่ไม่ได้ใช้งานเหลือเพียงจากการแปลงเป็น token (input\_ids, token\_type\_ids, attention\_mask) และ labels

A white background with black text

Description automatically generated

รูปที่ 7 แสดงตัวอย่างผลการ tokenize

1. แปลงให้ข้อมูลเป็นรูปแบบ torch เพื่อการนำไปเข้าแบบจำลองด้วยฟังก์ชัน set\_format
2. เรียกใช้แบบจำลองที่มีค่าน้ำหนักการให้การเรียนรู้มาแล้วด้วยคำสั่ง AutoModelForSequenceClassification.from\_pretrain และตั้งค่า parameter ให้ถูกต้องตามเหมาะสม

A screenshot of a computer

Description automatically generated

รูปที่ 8 แสดงตัวอย่างผลเรียกใช้แบบจำลองและการตั้งค่าพารามิเตอร์

1. เนื่องจากจะใช้คำสั่ง trainer จึงต้องมีการตั้งค่า Argument ต่างๆ ตามต้องการด้วยคำสั่ง TrainningAgruments เช่น learning rate epoch การแสดงผลบน wanb

A screenshot of a computer code

Description automatically generated

รูปที่ 9 แสดงตัวอย่างการตั้งค่า Argument

1. จากนั้นมีการกำหนดการแสดงวิธีการวัดผลของแบบจำลองด้วยฟังก์ชัน multi\_label\_metrics ที่เป็นการให้สามารถกำหนดข้อมูลจริงและข้อมูลทำนายเพื่อการนำไปวิเคราะห์ค่า f1-score, roc-auc, accuracyและ compute\_matrics ปรับให้เหมาะแก่การนำไปใช้ใน trainer โดยมีการเรียกใช้ multi\_label\_metrics ในการคำนวณผล

A screen shot of a computer code

Description automatically generated

รูปที่ 10 แสดงฟังก์ชันที่จะใช้ในการประเมินค่าแบบจำลอง

1. ตั้งค่าแบบจำลองโดยคำสั่ง trainer เช่น แบบจำลองที่ใช้ ชุดข้อมูลทดลองและทดสอบ

A computer code with black text

Description automatically generated

รูปที่ 11 แสดงตัวอย่างการตั้งค่า Trainer()

1. ให้การเรียนรู้แบบจำลองด้วย .train()

A screenshot of a computer

Description automatically generated

รูปที่ 12 แสดงตัวอย่างผลการให้การเรียนรู้เมื่อใช้คำสั่ง .train()

1. และวัดผลด้วย .evaluate()

A screenshot of a computer

Description automatically generated

รูปที่ 13 แสดงตัวอย่างผลการให้การเรียนรู้เมื่อใช้คำสั่ง .evaluate()

1. บันทึกแบบจำลองด้วย save\_model และเรียกมาใช้ด้วยคำสั่ง from\_pretrain เช่นเดิมเพื่อการนำมาทำนาย

A screenshot of a computer

Description automatically generated

รูปที่ 14 แสดงตัวอย่างการบันทึกแบบจำลอง

1. จากนั้นวน text ละแถวในข้อมูลชุดทดสอบมาทำขั้นตอนเดียวกันการเตรียมข้อมูลก่อนเข้าแบบจำลองโดยย่อและนำไปสร้างเป็นตารางโดยมีการกำหนดลำดับของชื่อของ column ให้ตรงกับรูปแบบที่ต้องส่งตรวจ

A screen shot of a computer code

Description automatically generated

รูปที่ 15 แสดงตัวอย่างการทำนายผลชุดทดสอบ

1. บันทึกผลลัพธ์ที่ได้ด้วยคำสั่ง to\_csv

A screenshot of a computer screen

Description automatically generated

รูปที่ 16 แสดงตัวอย่างผลสำเร็จที่ได้จากการทำนาย

**Chapter 4: Results**

เลือกตัวอย่างผลการให้การเรียนรู้จากการปรับพารามิเตอร์ทั้งหมด 6 ผลการทดสอบโดยมีพารามิเตอร์ ดังนี้

ตารางที่ 1 แสดงผลการทดลองปรับพารามิเตอร์

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Parameter | | | | Eval Loss | Final Macro f1-score |
| Batch size | Learning rate | Epochs | Weight decay |
| 1 | 8 | 1e-5 | 10 | 1e-3 | 0.3867 | 0.0764 |
| 2 | 8 | 1e-5 | 40 | 1e-3 | 0.3239 | 0.3739 |
| 3 | 8 | 1e-4 | 40 | 1e-3 | 0.3119 | 0.5867 |
| 4 | 8 | 1e-4 | 35 | 1e-4 | 0.3206 | 0.6039 |
| 5 | 8 | 1e-3 | 35 | 1e-4 | 0.4268 | 0.0321 |
| 6 | 16 | 1e-4 | 35 | 1e-4 | Out of memory | |

จากตารางพบว่าพารามิเตอร์ในแบบจำลองที่ 4 ได้ผล Macro f1-score สูงสุด จึงเลือกเป็นแบบจำลองที่นำไปทำนายผล

กราฟเปรียบเทียบผลแต่ละแบบจำลอง

A graph of a graph

Description automatically generated with medium confidence

รูปที่ 17 กราฟแสดงผลเปรียบเทียบค่า Macro f1-score แต่ละแบบจำลอง

A graph of a graph

Description automatically generated with medium confidence

รูปที่ 18 กราฟแสดงผลเปรียบเทียบค่า roc-auc แต่ละแบบจำลอง

A graph of a graph

Description automatically generated with medium confidence

รูปที่ 19 กราฟแสดงผลเปรียบเทียบค่า Loss แต่ละแบบจำลอง

A screenshot of a computer

Description automatically generated

รูปที่ 20 แสดง Leaderboard ใน Kaggle competition

จากภาพผลแสดงอันดับใน Kaggle นิสิตได้คะแนน Macro f1-score เท่ากับ 0.57912 จัดเป็นอันดับที่ 11 จาก 30

**Chapter 5: Discussion**

จากการทดสอบพบว่าเมื่อมีการปรับจำนวนรอบการให้เรียนรู้ (Epoch) Loss มีค่าลดลงเล็กน้อยและ Macro f1-score มีค่าสูงขึ้นถึงประมาณ 4 เท่าต่อมาเมื่อต้องการลดความ overfit จึงมีการทดสอบลดค่า learning rate ในแบบจำลองที่ 3 พบว่าค่า loss ลดลงเล็กน้อยแต่ค่า Macro f1-score มีค่าสูงขึ้นอีกสองเท่าเท่ากับ 0.5867 จากนั้นมีการทดสอบลดจำนวนรอบให้การเรียนรู้เนื่องจากเมื่อสังเกตที่ค่า loss และ Macro f1-score ในส่วนท้ายค่อนข้างคงที่ การลดจำนวนรอบให้การเรียนรู้จะช่วยลดความ overfit และการใช้หน่วยความจำโดยไม่จำเป็น และมีการลดค่า weight decay ผลที่ได้พบว่าค่า loss เพิ่มขึ้นเล็กน้อยแต่ค่า Macro f1-score เพิ่มขึ้นร้อยละ 3.4 เป็นค่าเท่ากับ 0.6039 ต่อมาจากแนวโน้มการแปรผกผันของค่า Macro f1-score และ learning rate จึงทดสอบเพิ่มค่า learning rate ซึ่งผลที่ได้พบว่าค่า loss สูงขึ้นมากตรงข้ามกับค่า Macro f1-score ที่ลดลงอย่างมากแสดงถึงการลด learning rate เป็น 0.001 มีผลลบต่อการทำนายของแบบจำลอง นั่นคือการปรับ learning rate ให้เหมาะสมมีผลต่อการทำนายของแบบจำลองอย่างมาก สุดท้ายเมื่อนำแบบจำลองที่ 4ไปทำนายกับข้อมูลชุดทดสอบได้ผล Macro f1-score เท่ากับ 0.5791 ซึ่งมีค่าใกล้เคียงกับผลที่ได้จากการทดสอบข้อมูลชุด Validation

**Chapter 6: Conclusion**

จากการทดสอบแบบจำลองที่ได้ค่า Macro f1-score ที่สูงสุดมีการตั้งค่า Batch sizeเท่ากับ 8, Learning rate เท่ากับ 1e-4, Epochs เท่ากับ 35 และ Weight decay เท่ากับ 1e-4 โดยได้ Macro f1-score เท่ากับ 0.6039 และเมื่อนำไปทำนายกับข้อมูลชุดทดสอบได้ผล Macro f1-score เท่ากับ 0.5791 ซึ่งสอดคล้องกันและจากการปรับพารามิเตอร์พบว่าจำนวนรอบการให้เรียนรู้และ learning rate มีผลอย่างมากต่อการทำนายแบบจำลองแต่ต้องหากปรับเพิ่มรอบมากไปแบบจำลองอาจ overfit และใช้หน่วยความจำอย่างไม่จำเป็น และการปรับ learning rate มากหรือน้อยเกินไปจะทำให้แบบจำลองเรียนรู้ได้อย่างไม่มีประสิทธิภาพ

อย่างไรก็ตามการปรับพารามิเตอร์ในการทดสอบครั้งนี้ค่อนข้างไม่ละเอียดและขาดการพิจารณาในส่วนของความไม่สมดุลของข้อมูลเนื่องจากขาดการจัดการเวลาในเวลาที่จำกัดและการให้การเรียนรู้แบบจำลองแต่ละครั้งค่อนข้างใช้เวลา

**Bonus**

ตารางที่ 2 แสดงผลการทดลองปรับพารามิเตอร์ที่ดีที่สุดใน dataset เดิมเทียบกับปรับพารามิเตอร์ใน dataset bonus

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Parameter | | | | Eval Loss | Macro f1-score |
| Batch size | Learning rate | Epochs | Weight decay |
| 4 | 8 | 1e-4 | 35 | 1e-4 | 0.3206 | 0.6039 |
| Bonus/1 | 8 | 1e-4 | 15 | 1e-4 | 0.3867 | 0.6460 |
| Bonus/2 | 16 | 1e-4 | 15 | 1e-4 | 0.3239 | 0.6837 |
| Bonus/3 | 16 | 1e-4 | 20 | 1e-4 | 0.3119 | 0.69 |

A graph with lines and text

Description automatically generated

รูปที่ 21 แสดงผลเปรียบเทียบค่า Macro f1-score ผลจาก dataset เดิมในแบบจำลองที่ได้ค่า Macro f1-score สูงสุดเทียบกับ dataset bonus

จากการทดสอบพบว่าเนื่องจาก dataset bonus มีจำนวน labels น้อยกว่า dataset เดิมนั่นคือใช้หน่วยความจำในการประมวลที่น้อยกว่าจึงสามารถปรับขนาด batch size เพิ่มขึ้นได้และสามารถใช้รอบการให้การเรียนรู้ที่ไม่สูงเท่าเดิมก็ได้ผลการประเมินที่สูงกว่าเพราะแบบจำลองเรียนรู้รูปแบบข้อมูลของแค่ labels 7 labels แต่ถ้าเพิ่มจนมีรอบการเรียนรู้ที่เท่ากันก็จำให้มีความแม่นยำมากขึ้นแต่ต้องพิจารณาถึงความ overfit