การจำแนกปัญหาของเทคโนโลยีฐานข้อมูลในชุมชนถามตอบออนไลน์

นายณัฐนัย สุวรรณชูชิต

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

Classification of Database Technology Problems in Online Question and Answer Community

Mr. Nuttanai Suwonchoochit

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

หัวข้อวิทยานิพนธ์	การจำแนกปัญหาของเทคโนโลยีฐานข้อมูลในชุมชนถามตอบออนไลน์
โดย	นายณัฐนัย สุวรรณชูชิต
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ
(ดร.ดวงดาว วิชาดากุล)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.เบญจพร ลิ้มธรรมาภรณ์)

ณัฐนัย สุวรรณชูชิต : การจำแนกปัญหาของเทคโนโลยีฐานข้อมูลในชุมชนถามตอบออนไลน์. (Classification of Database Technology Problems in Online Question and Answer Community) อ.ที่ปรึกษาหลัก : รองศาสตราจารย์. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

วิทยานิพนธ์นี้นำเสนอแนวทางการสร้างเครื่องมือการทำงานอัตโนมัติเพื่อจำแนกคำถามบนเว็บไซต์สแต็กโอเวอร์โฟลว์ โดยเฉพาะที่เกี่ยวกับชนิดของผลิตภัณฑ์ฐานข้อมูล ซึ่งถือเป็นข้อมูลที่มีค่าสำหรับเจ้าของผลิตภัณฑ์ฐานข้อมูลในการนำไปปรับปรุงผลิตภัณฑ์ หมวดหมู่ของคำถามกำหนดไว้เป็นสองระดับได้แก่ ระดับปัญหา และ ปัญหาย่อย โดยที่ระดับปัญหาประกอบด้วย การพัฒนา การติดตั้ง และ การปรับปรุงประสิทธิภาพ ในขณะที่ ปัญหาย่อย ประกอบด้วย การออกแบบ ข้อจำกัด และการอภิปรายปัญหา ด้วยการรวมทั้งสองระดับเข้าด้วยกัน คำถามจะถูกจำแนกออกเป็นเก้าหมวดของปัญหา-ปัญหาย่อย การประมวลผลภาษาธรรมชาติและการจำแนกข้อความถูกนำมาใช้ โดยใช้อัลกอริทึมการเรียนรู้ของเครื่องที่หลากหลาย โมเดลการจำแนกประเภทที่มีประสิทธิภาพดีที่สุดจะถูกนำมาใช้ในเว็บแอปพลิเคชัน เพื่อจำแนกแต่ละคำถามโดยใช้แท็กปัญหา-ปัญหาย่อย นอกจากนี้คำถามที่ถูกจำแนกออกตามหมวดแล้ว สามารถนำมาวิเคราะห์เพิ่มเติมโดยใช้อัลกอริทึมการสร้างแบบจำลองหัวข้อ เพื่อให้ทราบว่าคำถามในแต่ละหมวดนั้นกล่าวถึงหัวข้อใดบ้าง ซึ่งจะเป็นข้อมูลเพิ่มเติมให้กับเจ้าของผลิตภัณฑ์ฐานข้อมูลในการทำความเข้าใจถึงปัญหาของผลิตภัณฑ์เพื่อจะได้ทำการปรับปรุงต่อไป

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070490021 : MAJOR SOFTWARE ENGINEERING

KEYWORD: text classification, natural language processing, machine learning, Stack Overflow, software maintenance

Nuttanai Suwonchoochit : Classification of Database Technology Problems in Online Question and Answer Community . Advisor: Assc.Prof. Dr. TWITTIE SENIVONGSE

This thesis proposes an automated approach to classifying questions that are posted on Stack Overflow website with regard to a certain kind of database products in particular. Such information is valuable to database product owners for improving their products. The categories of questions are defined at two levels, i.e. problem and subproblem. The problem level includes development, installation, and performance tuning, while the subproblem level consists of design, limitation, and discussion. By cross-combining the two levels, questions can be classified into nine problem-subproblem classes. Natural language processing and text classification are used with several machine learning algorithms. The best classifier for all classes is used in a web application that can classify each question by a problem-subproblem tag. In addition, all classified questions are further analyzed by using a topic modeling algorithm to identify the topics that are addressed in those questions. This will be additional information for a database product owner to understand the issues of the database product for further improvement.

Field of Study: Software Engineering Student's Signature .....

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

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ทำให้สำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ของข้าพเจ้า รศ.ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา ผู้ที่ช่วยชี้แนะ แนะนำแนวทางและช่วยสอน ให้ความช่วยเหลือในการแก้ไขปัญหาต่าง ๆ ที่ตัวข้าพเจ้าประสบพบเจอในการทำวิทยานิพนธ์นี้ ร่วมถึงกำลังใจและแนวทางการทำงานที่ได้เรียนรู้จากอาจารย์ ขอบพระคุณเป็นอย่างสูง

ขอขอบคุณอาจารย์ รศ.ดร.วิวัฒน์ วัฒนาวุฒิ ผู้ซึ่งเป็นประธานกรรมการการสอบวิทยานิพนธ์ ดร. ดวงดาว วิชาดากุล และ รศ.ดร.เบญจพร ลิ้มธรรมาภรณ์ กรรมการสอบวิทยานิพนธ์ ที่ได้เสียสละเวลาเป็นอย่างมากในการตรวจสอบและสอบถาม พร้อมทั้งให้คำแนะนำในเรื่องดังกล่าวเพื่อทำให้วิทยานิพนธ์นี้สมบูรณ์มากขึ้น

ขอขอบคุณเพื่อน ๆ พี่ๆ น้อง ๆ ทั้งในภาควิชา และที่ทำงานที่ได้ช่วยแนะนำ ทั้งช่วยทดสอบประเมินผลต่าง ๆ ช่วยเรื่องของคำแนะนำการใช้งานโปรแกรมต่าง ๆ และช่วยติดป้ายข้อมูลที่มีมากถึง 13,000 ชุด ขอบคุณทางบริษัทที่ข้าพเจ้าร่วมงานได้ช่วยอำนวยความสะดวกอย่างดีเยี่ยมในการออกมาศึกษาต่อได้ ขอบคุณทุก ๆ ความช่วยเหลือที่มีมา

ณัฐนัย สุวรรณชูชิต

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง 1

สารบัญรูปภาพ 1

บทที่ 1 1

บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตงานวิจัย 2

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ประโยชน์ของงานวิจัย 3

1.6 บทความวิจัยที่ได้รับการตีพิมพ์ 3

บทที่ 2 4

ทฤษฎีที่เกี่ยวข้อง 4

2.1 Natural Language Processing (NLP) 4

2.2 Word2Vec 4

2.3 TF-IDF 4

2.4 Bag-of-Words (BOW) 5

2.5 Latent Dirichlet Allocation (LDA) 5

2.6 Support Vector Machine (SVM) 6
2.7 Naive Bayes 6
2.8 Decision tree 7
2.9 Random Forest 8
2.10 SOTorrent 8
2.11 Stack Overflow 8
2.12 Ensemble Method 10
2.13 XGBoost 12
บทที่ 3 13
งานวิจัยที่เกี่ยวข้อง 13
3.1 Toward Empirically Investigating Non-Functional Requirements of iOS Developers on Stack Overflow 13
3.2 SOTagRec: A Combined Tag Recommendation Approach for Stack Overflow 14
3.3 We Need to Talk about Microservices: an Analysis from the Discussions on Stack Overflow 15
3.4 SOTorrent : Reconstructing and Analyzing the Evolution of Stack Overflow Posts 16
3.5 Scalable Tag Recommendation for Software Information Sites 18
บทที่ 4 20
ภาพรวมงานวิจัย 20
บทที่ 5 23
การจำแนกปัญหาของระบบฐานข้อมูล 23
5.1 การรวบรวมสำรวจปัญหาและแยกกลุ่มเทคโนโลยี 23
5.2 การจัดเตรียมข้อมูลสำหรับการแยกกลุ่มปัญหาด้วยผู้เชี่ยวชาญ 25
บทที่ 6 29

การสร้างโมเดลการจำแนกประเภทปัญหาของระบบฐานข้อมูล 29
6.1 การประมวลผลข้อความเบื้องต้น 29
6.2 การทำ Feature Extraction และ Feature Transformations 30
6.3 การพัฒนาโมเดลการเรียนรู้ของเครื่องจากข้อมูล 31
6.4 เทคนิคที่ใช้แก้ปัญหาที่พบในการพัฒนาโมเดล 32
6.5 การประเมินผลประสิทธิภาพโมเดลการจำแนกปัญหา 34
6.6 การสรุปผลการประเมินผลประสิทธิภาพโมเดลการจำแนกปัญหาและแนวทางการปรับปรุงในอนาคต 45
6.7 การสร้างโมเดล Topic Modeling โดยใช้เทคนิค Latent Dirichlet Allocation (LDA) 48
6.8 การประเมินประสิทธิภาพ Topic Modeling ซึ่งใช้เทคนิค Latent Dirichlet Allocation (LDA) [29] 48
บทที่ 7 64
การพัฒนาเครื่องมือช่วยเหลือเจ้าของผลิตภัณฑ์และเครื่องมือช่วยค้นหาตัวอย่างกลุ่มปัญหาที่พบ 64
บทที่ 8 71
สรุปผลงานวิจัยและข้อเสนอแนะ 71
บรรณานุกรม 75
ประวัติผู้เขียน 79

สารบัญตาราง

หน้า

ตารางที่ 5-1 ลักษณะของกลุ่มปัญหา 23

ตารางที่ 5-1 ลักษณะของกลุ่มปัญหา (ต่อ) 24

ตารางที่ 5-2 ลักษณะของปัญหาย่อย 24

ตารางที่ 5-3 ตารางสรุปจำนวนข้อมูลแยกตามปัญหาและตามชื่อผลิตภัณฑ์ฐานข้อมูล 28

ตารางที่ 6-1 ตารางสรุปจำนวนก่อนและหลังทำข้อมูลแบบ SMOTE 33

ตารางที่ 6-2 สรุปเทคนิคที่ใช้งาน 35

ตารางที่ 6-3 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Installation ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ 36

ตารางที่ 6-4 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Development ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ 37

ตารางที่ 6-5 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Performance Tuning ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ 38

ตารางที่ 6-6 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Installation ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 13,000 รายการ 39

ตารางที่ 6-7 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Development ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 13,000 รายการ 40

ตารางที่ 6-8 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Performance Tuning ในแต่ละปัญหาย่อยระดับ ข้อมูลที่ 13,000 รายการ 41

ตารางที่ 6-9 ตารางแสดงประสิทธิภาพโดยรวมของโมเดล Multiclass Classification โดยใช้ข้อมูล 5,000 โพสต์ 44

ตารางที่ 6-10 ตารางสรุปผลค่า Coherence ของ Topics และ ตัวแปรที่ใช้ 55

ตารางที่ 6-11 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Design .. 56

ตารางที่ 6-12 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Discussion 56

ตารางที่ 6-13 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Limitation 57

ตารางที่ 6-14 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Design 57

ตารางที่ 6-15 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Discussion 58

ตารางที่ 6-16 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Limitation 58

ตารางที่ 6-17 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Design 59

ตารางที่ 6-18 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Discussion 60

ตารางที่ 6-19 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Limitation 60

ตารางที่ 6-20 ตารางสรุปจำนวน topic และ post ที่ใช้ทดสอบโมเดล 61

ตารางที่ 6-21 ตารางแสดงตัวอย่างข้อมูลที่ส่งให้ผู้เชี่ยวชาญอ่านและประเมินผลโดยเป็นข้อมูลจากกลุ่มปัญหา Development-Design (แสดงเฉพาะส่วน Post Header) 62

ตารางที่ 6-22 ตารางสรุปผลการประเมินเปรียบเทียบผลจากโมเดลและมุมมองของผู้เชี่ยวชาญ 62

ตารางที่ 6-23 ตารางค่าความแม่นยำของโมเดลจากการเฉลี่ยผลการประเมิน 62

ตารางที่ 7-1 สรุปรายการ API ที่เป็นส่วนประกอบสำคัญของระบบ 70

สารบัญรูปภาพ

หน้า

รูปที่ 2-1 รูปแบบการทำงานของ LDA [2] 5

รูปที่ 2-2 ตัวอย่าง Support Vector Machine [2] 6

รูปที่ 2-3 ตัวอย่าง Decision Tree [6] 7

รูปที่ 2-4 กระบวนการ Random Forest [7] 8

รูปที่ 2-5 ตัวอย่างหน้าเว็บไซต์ สแต็กโอเวอร์โฟลว์ในหน้ากระดานคำถาม 9

รูปที่ 2-6 หน้าเว็บไซต์ สแต็กโอเวอร์โฟลว์ในส่วนของหน้าเนื้อหาแต่ละคำถาม 9

รูปที่ 2-7 ตัวอย่างชื่อแท็กที่เกี่ยวข้องกับเนื้อหา 10

รูปที่ 2-8 ภาพโครงสร้าง Ensemble Method 10

รูปที่ 2-9 ภาพตัวอย่างการทำงานของ Ensemble Method 11

รูปที่ 3-1 ภาพรวมงานวิจัย [13] 13

รูปที่ 3-2 ภาพรวมของงานวิจัย SOTagRec [14] 14

รูปที่ 3-3 ภาพรวมของงานวิจัย [15] 15

รูปที่ 3-4 กลุ่มต่าง ๆ หลังจากการตีความหัวข้อที่ได้จาก topic modeling [15] 16

รูปที่ 3-5 ตัวอย่างโครงสร้างเนื้อหาในสแต็กโอเวอร์โฟลว์ [16] 17

รูปที่ 3-6 ตัวอย่างการเปรียบเทียบเพื่อดูวิวัฒนาการของเนื้อหา [16] 18

รูปที่ 3-7 ภาพรวมของ TagMulRec [17] 19

รูปที่ 4-1 ภาพรวมงานวิจัย 21

รูปที่ 4-2 ตัวอย่างหน้าจอเว็บไซต์ และส่วนของข้อมูลที่นำมาใช้งาน 22

รูปที่ 5-2 ข้อมูลบนเว็บไซต์สแต็กโอเวอร์โฟลว์ 25

รูปที่ 5-3 ตัวอย่างข้อมูลที่ได้จาก SOTorrent 25

รูปที่ 5-4 ตัวอย่างข้อมูลที่ได้จาก SOTorrent 26

รูปที่ 5-5 Classification Group 27

รูปที่ 5-6 ตัวอย่างการติดป้ายกับข้อมูล 27
รูปที่ 6-1 ตัวอย่างการทำ SMOTE [26],[27] 32
รูปที่ 6-2 สัดส่วนของข้อมูลที่มีคำศัพท์ที่สามารถแยกกลุ่มได้ทันทีกับคำทั่วไป 42
รูปที่ 6-3 รูปกราฟแสดงตัวอย่างแนวโน้มของผลการทดลองต่อจำนวนข้อมูลที่ใช้ทดลองด้วย TF-IDF 43
รูปที่ 6-4 รูปกราฟแสดงตัวอย่างแนวโน้มของผลการทดลองต่อจำนวนข้อมูลที่ใช้ทดลองด้วย Word2Vec 44
รูปที่ 6-5 รูปตัวอย่างโพสต์ที่สามารถตีความได้ 2 กลุ่มปัญหา 46
รูปที่ 6-6 การหาค่า Coherence จาก Wn-Wn โดยที่ W คือ Word [31] 49
รูปที่ 6-7 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Design 49
รูปที่ 6-8 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Discussion .. 51
รูปที่ 6-9 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Limitation .. 51
รูปที่ 6-10 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Design 52
รูปที่ 6-11 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Discussion 52
รูปที่ 6-12 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Limitation 53
รูปที่ 6-13 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Design 53
รูปที่ 6-14 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Discussion 54
รูปที่ 6-15 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Limitation 54
รูปที่ 6-16 flow การประเมินผลโมเดล LDA กับผู้ประเมิน 61
รูปที่ 7-1 ภาพโครงสร้างระบบและการทำงานเบื้องต้น 64
รูปที่ 7-2 ภาพตัวอย่างหน้าจอระบบในหน้าแรก 65
รูปที่ 7-3 ภาพตัวอย่างหน้ารายละเอียดของโพสต์ 66

รูปที่ 7-4 ตัวอย่างกราฟแท่ง (Bar) แสดงความถี่ของแต่ละปัญหาของฐานข้อมูล MySQL ในช่วง-Q4 ปี 2019 67

รูปที่ 7-5 ตัวอย่างการแสดงผลที่แตกต่างกันระหว่างเนื้อหากับภาษาโปรแกรม 67

รูปที่ 7-6 ตัวอย่างการแสดงผล และทดสอบการเชื่อมโยง link ของข้อมูล 68

รูปที่ 7-7 ตัวอย่างการใช้เลขรหัสเจ้าของโพสต์ในระบบของงานวิจัย 69

รูปที่ 7-8 ตัวอย่างการใช้เลขรหัสเจ้าของโพสต์ในเว็บไซต์ต้นฉบับ 69

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในโลกของเทคโนโลยีจากอดีตสู่ปัจจุบันได้มีการปรับปรุงและเปลี่ยนแปลงไปในหลาย ๆ ด้าน ทั้งในแง่ของการใช้งาน เอกสารคู่มือ การแก้ไขปัญหา และในทุก ๆ ครั้งที่มีการปรับปรุงการเปลี่ยนแปลงในหลาย ๆ ครั้ง เทคโนโลยีที่เปลี่ยนไป รูปแบบการทำงานการใช้งานที่เปลี่ยนไปได้สร้างปัญหาและวิธีการเรียนรู้ใหม่ ๆ อยู่เสมอ ซึ่งในจุดนี้ผู้ให้บริการเทคโนโลยีดังกล่าวจำเป็นต้องมีการสร้างคู่มือการใช้งาน หรือการเก็บข้อมูลการใช้งานว่าสิ่งที่นำเสนอออกไปนั้น สามารถแก้ไขปัญหาดังกล่าวได้จริงถูกต้องและไม่สร้างปัญหาใหม่ การทำงานในหลาย ๆ ครั้งสำหรับแวดวงไอทีอาจจะต้องมีการเลือกใช้เทคโนโลยี ซึ่งการเลือกใช้เทคโนโลยีมักจะมีคำถามเหล่านี้เสมอเช่น “การใช้งานยากไหม” ”ติดตั้งง่ายไหม” เป็นต้น คำถามที่ยกมาข้างต้น ไม่เพียงจะขึ้นกับผู้ใช้งานทั่วไปเท่านั้น ในแง่เจ้าของผลงาน เจ้าของเทคโนโลยี หรือทีมนักพัฒนาที่ได้รับหน้าที่ในการดูแลเองก็ประสบพบปัญหาดังกล่าว ในปัจจุบันปัญหาการใช้งานต่าง ๆ ทั้งในแง่การติดตั้งเทคโนโลยี การนำไปใช้เอกสารคู่มือรวมทั้งแนวทางการแก้ปัญหาได้มีการรวบรวมไว้ในชุมชนถามตอบออนไลน์ ซึ่งชุมชนที่ได้รับความนิยมแห่งหนึ่งได้แก่ สแต็กโอเวอร์โฟลว์ (Stack Overflow) โดยในชุมชนดังกล่าวได้มีการรวบรวมและให้บริการพื้นที่สำหรับรายงานปัญหาที่พบและการตอบคำถาม ใช้ในการระดมความคิดในการแก้ไขปัญหาและสร้างสรรค์องค์ความรู้ใหม่ ๆ ในวงการเทคโนโลยี แต่ข้อมูลในสื่อดังกล่าวที่มีจำนวนมากนั้น ไม่ได้มีการแยกหมวดหมู่ในระดับรูปแบบการใช้งานอย่างชัดเจน เช่น ปัญหาการติดตั้ง ปัญหาการนำไปใช้งาน ปัญหาการปรับปรุงขีดความสามารถการทำงาน ซึ่งหมวดหมู่ทั้ง 3 หัวข้อหลักก็เป็นเรื่องที่น่าสนใจในมุมมองการนำไปพัฒนาต่อยอดหรือปรับปรุงให้มีประสิทธิภาพมากยิ่งขึ้น สำหรับกลุ่มผู้ใช้ทั่วไปจะได้ประโยชน์ในการสืบค้นและได้เครื่องมือช่วยในการตัดสินใจนำไปใช้งาน เพราะสามารถรับรู้ปัญหาและวิธีแก้ไขก่อนนำไปใช้งานได้ และสำหรับกลุ่มเจ้าของเทคโนโลยีจะได้ประโยชน์ในการเก็บข้อมูลการใช้งานจริง ปัญหาที่พบหลังจากการนำผลงานออกสู่ตลาด เพื่อให้สามารถแก้ไขและปรับปรุงตามปัญหาที่พบจริง

ดังนั้น ในงานวิจัยนี้จะนำเสนอการนำข้อมูลคำถาม ที่เกี่ยวข้องกับเทคโนโลยีจากสแต็กโอเวอร์โฟลว์มาใช้งานและสร้างเครื่องมือพร้อมโมเดลการเรียนรู้ของเครื่อง ในการจำแนกข้อความ (Text Classification) เพื่อคัดแยกหมวดหมู่ของปัญหาและทำการแนะนำ Tag ตามหมวดหมู่ดังกล่าวขึ้นมาซึ่งประกอบไปด้วย ปัญหาด้านการติดตั้ง (Installation) ปัญหาด้านการนำไปพัฒนา (Development) และปัญหาด้านการปรับปรุงประสิทธิภาพ (Performance Tuning) โดยในกลุ่มทั้งสามยังมีการจัดหมวดหมู่ย่อยเพิ่มเติมในแง่ของการอภิปรายปัญหา (Discussion) การออกแบบ

(Design) และข้อจำกัดของเทคโนโลยี (Limitation) ทั้งสามหัวข้อย่อยจะช่วยให้ทำการทวนสอบย้อนกลับและนำไปใช้ปรับปรุงเทคโนโลยีได้ นอกจากนี้คำถามที่ถูกจำแนก แยกตามหมวดแล้วสามารถนำมาวิเคราะห์เพิ่มเติมโดยใช้อัลกอริทึมการสร้างแบบจำลองหัวข้อ (Topic Modeling) เพื่อให้ทราบว่าคำถามในแต่ละหมวดนั้นกล่าวถึงหัวข้อใดบ้าง ซึ่งจะเป็นข้อมูลเพิ่มเติมให้กับเจ้าของผลิตภัณฑ์ฐานข้อมูลในการทำความเข้าใจถึงปัญหาของผลิตภัณฑ์เพื่อจะได้ทำการปรับปรุงต่อไป

1.2 วัตถุประสงค์

เพื่อสร้างโมเดลในการจำแนกประเภทปัญหาของเทคโนโลยีสำหรับชุมชนถามตอบออนไลน์และสร้างเครื่องมือสำหรับใช้งานโมเดลจำแนกและออกรายงานแสดงผล

1.3 ขอบเขตงานวิจัย

1.3.1 สร้างโมเดลสำหรับการจำแนกโพสต์ในสแต็กโอเวอร์โฟลว์

1.32 สร้างฐานข้อมูลสำหรับรวบรวมปัญหาที่เกี่ยวกับเทคโนโลยีจากเว็บไซต์สแต็กโอเวอร์โฟลว์โดยใช้ข้อมูลตั้งต้นจากแหล่งข้อมูล SOTorrent และทดลองกับข้อมูลผลิตภัณฑ์ฐานข้อมูล

1.3.3 พัฒนาวิธีการจำแนกโพสต์ว่าอยู่ในกลุ่มปัญหาประเภทใดโดย ส่วนแรกจะเป็นกลุ่มของปัญหาได้แก่ Installation, Development, Performance Tuning ส่วนที่สองจะเป็นกลุ่มของปัญหาย่อยได้แก่ Limitation, Discussion และ Design โดยการพัฒนาจะใช้ภาษาไพทอน

1.3.4 พัฒนาวิธีการวิเคราะห์หัวข้อต่าง ๆ ที่ ถูกกล่าวถึงในคำถามของแต่ละหมวดหมู่โดยการพัฒนาจะใช้ภาษาไพทอน

1.3.5 สร้างเครื่องมือสนับสนุนผู้ใช้ผลิตภัณฑ์และเจ้าของผลิตภัณฑ์ โดยสามารถจำแนกโพสต์ในสแต็กโอเวอร์โฟลว์ ค้นหาโพสต์ และ แสดงผลประเภทปัญหาตามกลุ่มผลิตภัณฑ์ฐานข้อมูล ผ่านเทคโนโลยีเว็บแอปพลิเคชัน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาข้อมูลเอกสารและงานวิจัยที่เกี่ยวข้องกับรูปแบบข้อมูลที่ต้องใช้งาน

1.4.2 ศึกษาข้อมูลรูปแบบวิธีการในการตรวจสอบเทคโนโลยีบนเว็บไซต์สแต็กโอเวอร์โฟลว์

1.4.3 ศึกษาข้อมูลและสร้างรูปแบบข้อมูลที่ควรจะเป็น

1.4.4 พัฒนาแบบจำลองการจำแนกเพื่อใช้ในการแยกแต่ละหมวดหมู่ปัญหาของคำถามและแบบจำลองวิเคราะห์หัวข้อต่าง ๆ ที่ปรากฏในคำถาม

1.4.5 พัฒนาเครื่องมือสนับสนุนและใช้งานแบบจำลอง

1.4.6 ทดสอบแบบจำลองและ ปรับปรุง

1.4.7 สรุปผลการวิจัยและเรียบเรียงและจัดทำบทความวิชาการ

1.4.8 เรียบเรียงและจัดทำวิทยานิพนธ์

1.5 ประโยชน์ของงานวิจัย

1.5.1 เข้าใจภาพรวมปัญหาของกลุ่มเทคโนโลยีและช่วยค้นหาสาเหตุของปัญหาได้

1.5.2 ได้โมเดลหรือแบบจำลองวิธีการคัดแยกกลุ่มปัญหาในสแต็กโอเวอร์โฟลว์

1.5.3 ผู้ใช้ทั่วไปได้เครื่องมือที่แสดงผลปัญหาประเภทต่าง ๆ เพื่อช่วยในการตัดสินใจเลือกเทคโนโลยีเพื่อนำไปใช้งาน

1.5.4 เจ้าของเทคโนโลยีมีเครื่องมือรวบรวมและจำแนกประเภทปัญหาและให้เข้าใจปัญหาและสามารถนำไปใช้ปรับปรุงหลังจากปล่อยเทคโนโลยีที่นำเสนอออกสู่ตลาด

1.6 บทความวิจัยที่ได้รับการตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์ได้รับการตีพิมพ์เป็นบทความวิชาการเรื่อง “Classification of Database Technology Problems on Stack Overflow” ในงานประชุมวิชาการที่งาน 2021 IEEE/ACIS 19th International Conference On Software Engineering Research, Management Applications (SERA) ในช่วงระหว่างวันที่ 20-22 มิถุนายน พ.ศ. 2564 ณ เมืองคานาซาว่า ประเทศญี่ปุ่น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Natural Language Processing (NLP)

NLP หรือ Natural Language Processing [1] เป็นหนึ่งในสาขาย่อยของปัญญาประดิษฐ์และหลักภาษาศาสตร์ที่ศึกษาปัญหาในการประมวลผลและใช้งานภาษาธรรมชาติ รวมทั้งการทำความเข้าใจภาษาธรรมชาติ ทั้งนี้เพื่อให้คอมพิวเตอร์สามารถเข้าใจภาษามนุษย์ได้โดยในหลักการดังกล่าวสามารถนำไปปรับใช้งานได้หลากหลายรูปแบบเพื่อทำหน้าที่เป็นกระบวนการในการเรียนรู้เข้าใจระหว่างภาษาทั้งสองเข้าด้วยกัน โดยในหลักการ แนวคิดสามารถนำมาใช้งานในการแยกคำ บริบทของภาษาโดยใช้คอมพิวเตอร์ให้ช่วยในการทำงานได้อย่างถูกต้องทั้งในแง่การสะกด การแยกคำ เป็นต้น

2.2 Word2Vec

Word2Vec [2] เป็นอีกหนึ่งในโมเดลด้าน Feature Extraction สำหรับงาน NLP หรือ Natural Language Processing ที่มีไว้สำหรับเพื่อการทำ word embedding เพื่อทำให้การแบ่งหรือจัดกลุ่มคำอยู่ในรูปแบบ vector เพื่อที่จะไว้ใช้งานหรือทำกระบวนการอื่นต่อ ๆ ไปได้ ในโมเดลดังกล่าวมีเป้าหมายสำคัญ ๆ คือ การทำ Word similarity เพื่อใช้ในการคัดแยกความเหมือนและจัดกลุ่มต่าง ๆ นอกจากนี้ ยังนำผลที่ได้จากโมเดลไปปรับปรุงและช่วยในการทำงานร่วมกับโมเดลประเภท Deep Learning อีกด้วย แนวทางการทำ Word2Vec แบ่งได้สองประเภทคือ **Skip-Gram และ CBOW (Continuous Bag-of-Words)**

2.3 TF-IDF

TF-IDF (Term Frequency Inverse Document Frequency) [2] เป็นวิธีการในการคำนวณค่าน้ำหนักของคำจากความถี่ของคำที่พบในเอกสาร เป็นหนึ่งในกระบวนการ Feature Extraction โดยในการคำนวณค่า TF-IDF จะคำนวณน้ำหนักของคำที่พบในเอกสารหลังจากตัด stopword ออกไปโดยสูตรเป็นดังสมการที่ (1)

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D) \quad (1)$$

ในส่วนของ TF-IDF จะถูกแยกเป็น 2 ส่วน นั้นคือ TF กับ IDF โดย TF คือ Term Frequency กับ IDF คือ Inverse Document Frequency โดย TF จะมีแบ่งเป็นวิธีย่อย ๆ เช่น Raw Count, log normalization, term frequency เป็นต้น ส่วน IDF จะมีแบ่งเป็นวิธีย่อย ๆ เช่น IDF Normal,IDF Smooth ,IDF Max และ Probabilistic IDF โดยในงานวิจัยฉบับนี้จะใช้สูตร tf ดังสมการที่ (2) และ idf ดังสมการที่ (3)

$$1 + \log f_{t,d} \tag{2}$$

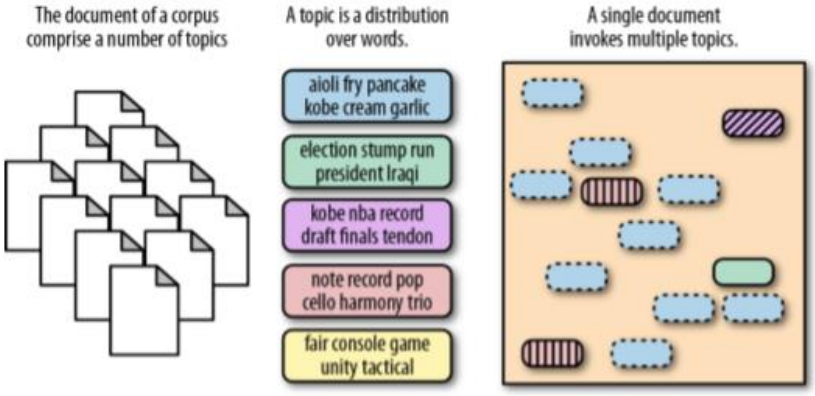
$$\log\left(1 + \frac{N}{n_t}\right) \tag{3}$$

2.4 Bag-of-Words (BOW)

Bag-of-Words (BOW) [2] เป็นหนึ่งในโมเดลที่ทำการรวบรวมคำในเอกสารและช่วยในการจัดหมวดหมู่กลุ่มคำจัดแบ่งประเภท เป็นเสมือนกระเป๋าเก็บคำที่ได้จากการเรียนรู้ เพื่อสร้างเป็น Vector คลังข้อมูลคำโดยไม่ได้คำนึงถึงหลักไวยากรณ์และลำดับของคำในเอกสาร

2.5 Latent Dirichlet Allocation (LDA)

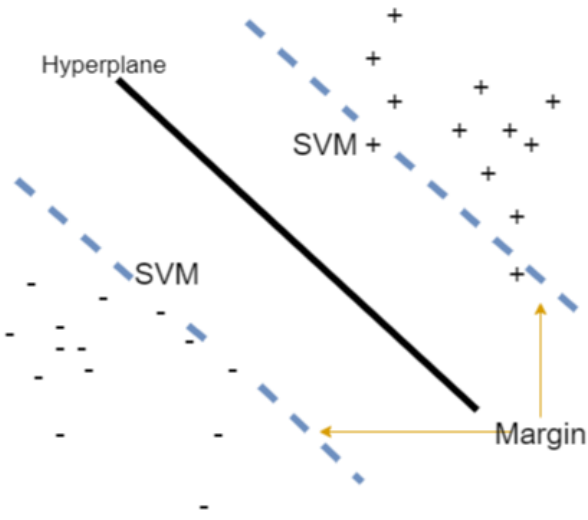
Latent Dirichlet Allocation (LDA) [2],[3] เป็นเทคนิคสำหรับการค้นหาหัวข้อหรือกลุ่มหรือประเด็นที่อยู่ในเอกสาร LDA ใช้เทคนิคการคำนวณความน่าจะเป็นร่วมกับความถี่ เพื่อให้สืบทราบและค้นหาหัวข้อที่น่าจะเป็นในเอกสาร LDA มีคุณลักษณะเฉพาะของโมเดลคือชุดคำศัพท์ของหัวข้อไม่จำเป็นต้องแตกต่างกันและอาจมีคำชุดคำศัพท์หัวข้อที่เมื่อแสดงผลออกมา อาจจะไม่สามารถสื่อความหมายได้ชัดเจนว่าหมายถึงกลุ่มอะไรตามรูปที่ 2-1 ข้อเสียที่เด่นชัดของ LDA คือต้องมีการใช้มนุษย์ในการตีความชุดคำศัพท์ว่าสื่อความหมายดีหรือไม่ หรือคำศัพท์นี้หมายถึงกลุ่มหัวข้ออะไร



รูปที่ 2-1 รูปแบบการทำงานของ LDA [2]

2.6 Support Vector Machine (SVM)

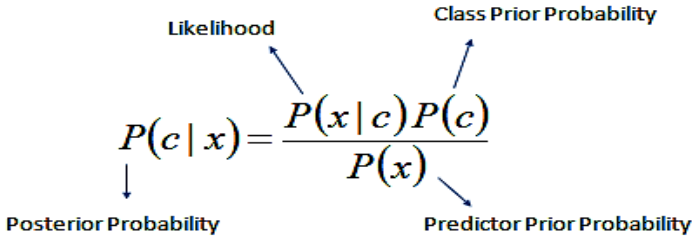
SVM [4] คือขั้นตอนวิธีการเพื่อช่วยแยกหรือจำแนกประเภทกลุ่มข้อมูล โดยมีแนวคิดคือนำข้อมูลมาแทนค่าเป็นเวคเตอร์แล้วนำข้อมูลมาจัดเรียงพร้อมหาเส้นที่ใช้แบ่งข้อมูลทั้งสองออกจากกัน โดยจะสร้างเส้นแบ่ง (Hyperplane) เพื่อให้ทราบว่าเส้นตรงที่แบ่งสองกลุ่มออกจากกันนั้น เส้นตรงใดเป็นเส้นที่ดีที่สุด เส้นตรงเส้นใดที่ดีที่สุดจะถูกนิยาม Margin ซึ่งเป็นผลรวมระยะห่างของเส้นตรงที่เป็นเส้นแบ่งตามรูปที่ 2-2



รูปที่ 2-2 ตัวอย่าง Support Vector Machine [2]

2.7 Naive Bayes

Naive Bayes [5] เป็นหนึ่งในวิธีการและแนวคิดในเรื่องของการเรียนรู้ของเครื่องโดยมีแนวคิดคือการคำนวณความน่าจะเป็นมาใช้หรือการสุ่ม สามารถคำนวณได้ตามสมการที่ (4)



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c) \qquad (4)$$

จากสมการของ Naive Bayes จะมี 4 ส่วนคือ Posterior probability หรือ P(C|X) คือ ค่าความน่าจะเป็นที่ข้อมูลที่มีแอตทริบิวต์เป็น X จะมีคลาส C

Likelihood หรือ P(X|C) คือ ค่าความน่าจะเป็นที่ข้อมูลสำหรับให้โมเดลเรียนรู้ที่มีคลาส C และมีแอตทริบิวต์ X

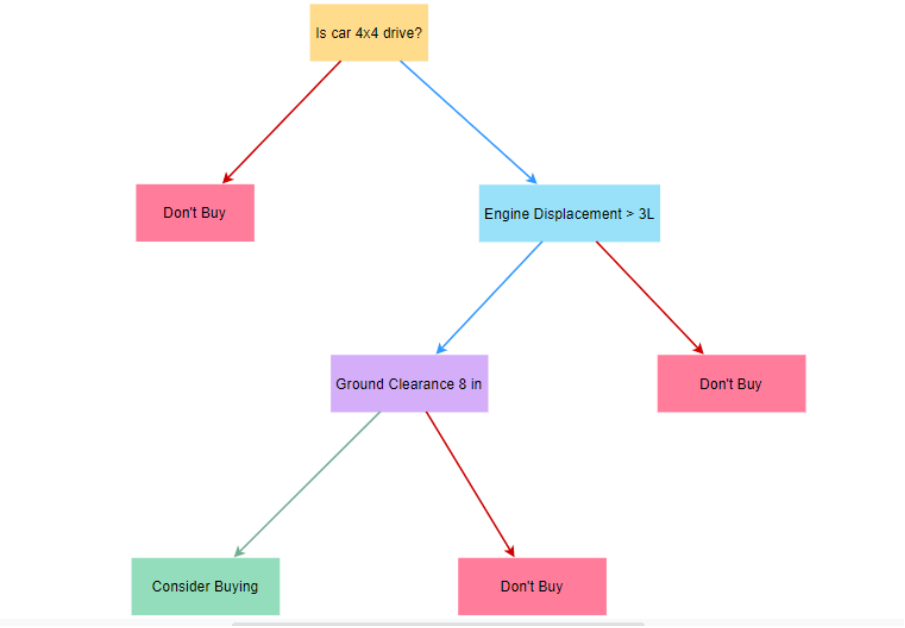
Class Prior probability หรือ P(C) คือ ค่าความน่าจะเป็นของคลาส C

Predictor Prior probability หรือ P(X) คือ ค่าความน่าจะเป็นของการตัวทำนายผล X

แนวทางของ Naïve Bayes สามารถใช้ช่วยแก้ไขปัญหาความไม่แน่นอน ช่วยให้ตัดสินใจความน่าเชื่อถือซึ่งในงานวิจัยนี้ได้เลือกวิธีการดังกล่าวเป็นหนึ่งในวิธีการจัดกลุ่ม

2.8 Decision tree

Decision tree [4] หรือ การเรียนรู้แบบต้นไม้ตัดสินใจ เป็นหนึ่งในวิธีการและแนวคิดในเรื่องของการเรียนรู้ของเครื่องโดยมีแนวคิดคือการที่เครื่องเรียนรู้ข้อมูลแล้วนำข้อมูลที่ได้เรียนรู้มาช่วยแบ่งกลุ่มในลักษณะการตัดสินใจแบบใช่หรือไม่ใช่และการตัดสินใจของแนวคิดการเรียนรู้ดังกล่าวจะเป็นแบบ “ถ้า-แล้ว” แบ่งเป็นเส้นทางและทำวนต่อเนื่องจนไม่มีกระบวนการการตัดสินใจเกิดขึ้นอีก ซึ่งแสดงให้เห็นตัวอย่างตามรูปที่ 2-3

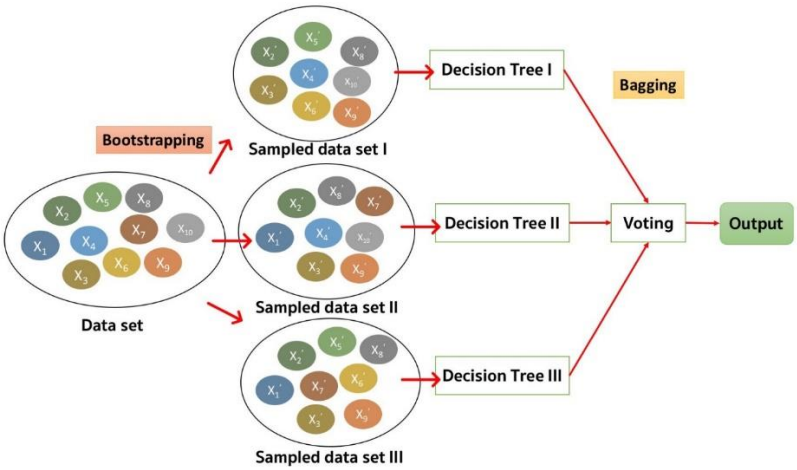


รูปที่ 2-3 ตัวอย่าง Decision Tree [6]

แนวทางของ Decision tree สามารถใช้ช่วยแก้ไขปัญหา ช่วยในเรื่องของการทำนาย การตัดสินใจและการจัดกลุ่ม

2.9 Random Forest

เป็นวิธีการที่ต่อยอดมาจาก Decision tree ในส่วนของ Random Forest [4],[7] จะมีวิธีการที่มองเป็นภาพใหญ่ของการคำนวณความน่าจะเป็นรวมกับวิธี Decision tree โดยจะเป็นการสร้าง Decision tree เป็นจำนวนมากและใช้การคำนวณความน่าจะเป็นจากกลุ่มของ Decision tree หรือการทำ Voting ซึ่งข้อมูลที่ใช้สร้างRandom Forest จะถูกสุ่มกระจายไปตาม Decision tree ชุดต่าง ๆ แล้วถึงจะนำผลที่ได้มาเทียบเคียงกันตามรูปที่ 2-4



รูปที่ 2-4 กระบวนการ Random Forest [7]

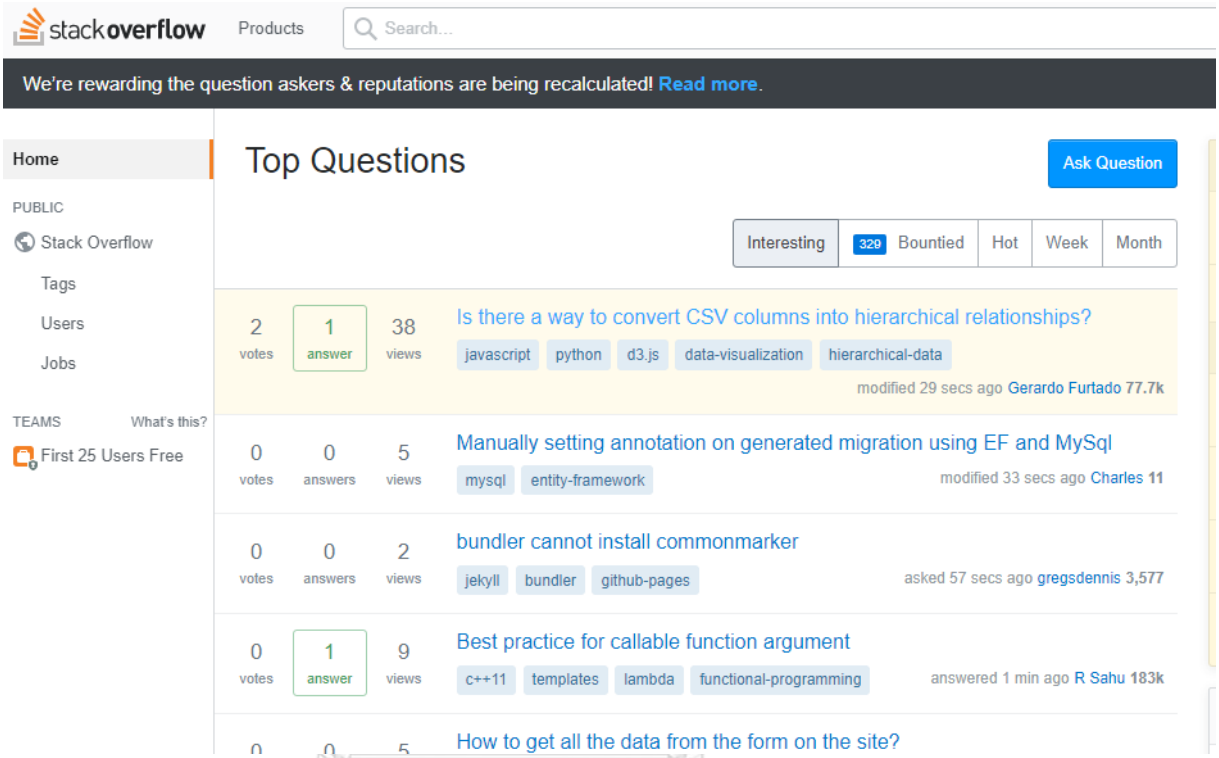
2.10 SOTorrent

SOTorrent [8] เป็นชุดข้อมูลหรือ Data set ที่มีการเก็บรวบรวมข้อมูลจากเว็บไซต์ สแต็กโอเวอร์โฟลว์ (Stack Overflow) ซึ่งเป็นเว็บไซต์ที่ทำหน้าทีเป็นกระดานคำถาม คำตอบสำหรับเรื่องเทคโนโลยีและการพูดคุย ซึ่ง SOTorrent เก็บรวบรวมโดยทีม Empirical-Software Engineering ได้ทำการรวบรวมทั้งหัวข้อและงานวิจัย พร้อมข้อมูลในช่วงต่างของสแต็กโอเวอร์โฟลว์อีกด้วย ในชุดข้อมูลดังกล่าวมีข้อมูลจำนวนมากที่เปิดให้ใช้งานได้โดยลักษณะของชุดข้อมูล SOTorrent มีการเก็บเป็นไฟล์นามต่าง ๆ โดยมีชนิดไฟล์ นามสกุลไฟล์ เช่น XML และ CSV เป็นต้น

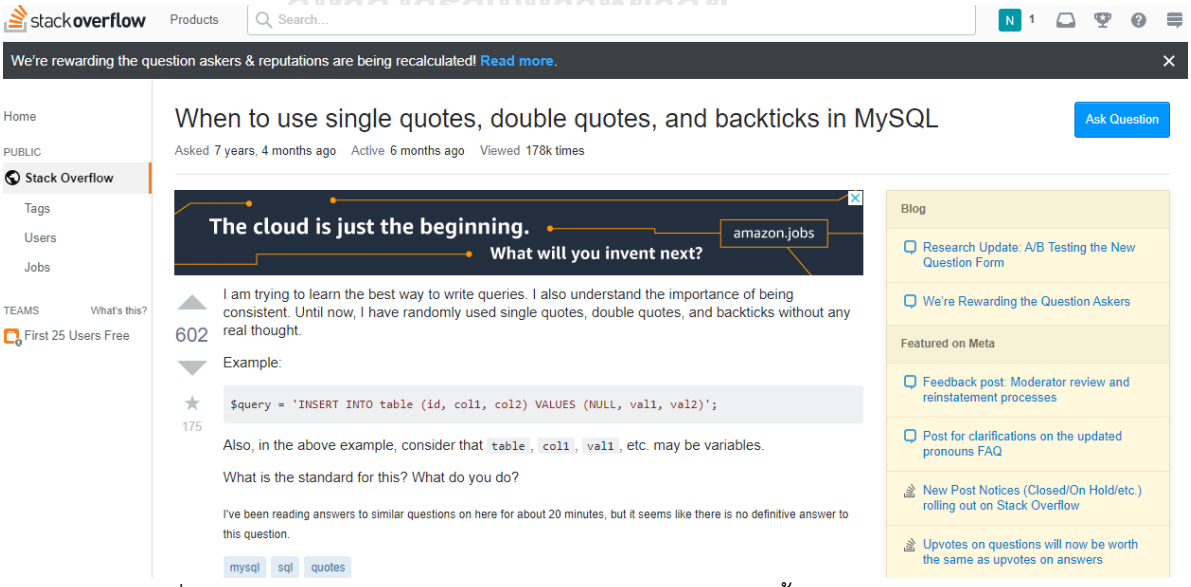
2.11 Stack Overflow

เว็บไซต์ สแต็กโอเวอร์โฟลว์ (Stack Overflow) [9] เป็นเว็บไซต์ที่ทำหน้าที่เป็นกระดานคำถาม คำตอบสำหรับเรื่องเทคโนโลยีและการพูดคุย หรือแสดงความคิดเห็นในแง่มุมต่าง ๆ สำหรับเทคโนโลยีนั้น ๆ ซึ่งเป็นชุมชนออนไลน์ขนาดใหญ่ที่มีข้อมูลจำนวนมากและมีการพูดถึง หรืออ้างอิง

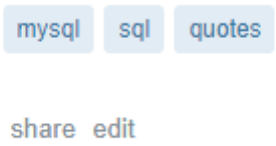
บ่อยครั้ง โดยในเว็บดังกล่าวนอกจากให้บริการในเรื่องของพื้นที่ตั้งคำถามและตอบ ยังให้บริการเครื่องมืออำนวยความสะดวกแก่สายงานด้านเทคโนโลยี ในเว็บดังกล่าวมีการคัดแยกเรื่อง หัวข้อต่าง ๆตามหมวดหมู่โดยใช้ระบบ Tag ระบบดังกล่าวจะเป็นตัวบอกว่าเนื้อในส่วนนี้เกี่ยวข้องกับเรื่องอะไร เทคโนโลยีอะไรซึ่งมีการแสดงผลตามรูปที่ 2-5, 2-6 และ 2-7



รูปที่ 2-5 ตัวอย่างหน้าเว็บไซต์ สแต็กโอเวอร์โฟลว์ในหน้ากระดานคำถาม



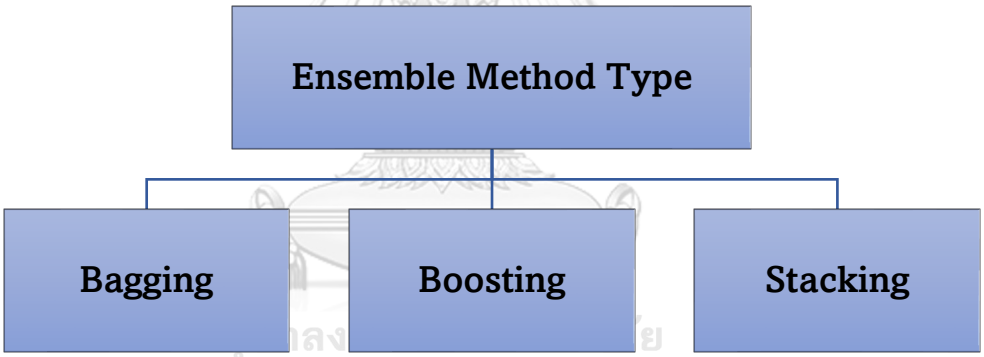
รูปที่ 2-6 หน้าเว็บไซต์ สแต็กโอเวอร์โฟลว์ในส่วนของหน้าเนื้อหาแต่ละคำถาม



รูปที่ 2-7 ตัวอย่างชื่อแท็กที่เกี่ยวข้องกับเนื้อหา

2.12 Ensemble Method

Ensemble Method [10],[11] คือหนึ่งในเทคนิคของ Machine learning ที่ใช้วิธีการนำโมเดลหลาย ๆ โมเดลมารวมกันเพื่อที่จะทำให้ผลลัพธ์ออกมาดีที่สุด สำหรับการแก้ไขปัญหาของโมเดลที่มีข้อดีข้อเสียแตกต่างกัน โดยสามารถช่วยให้การรวมคุณสมบัติที่ดีที่สุดของแต่ละส่วนมารวบรวมเพื่อสร้างอันใหม่ที่ดีที่สุด Ensemble Method มีลักษณะอัลกอริทึม 3 รูปแบบย่อยตามรูปที่ 2-8 เป็นภาพโครงสร้างที่ประกอบด้วย Bagging , Boosting ,Stacking

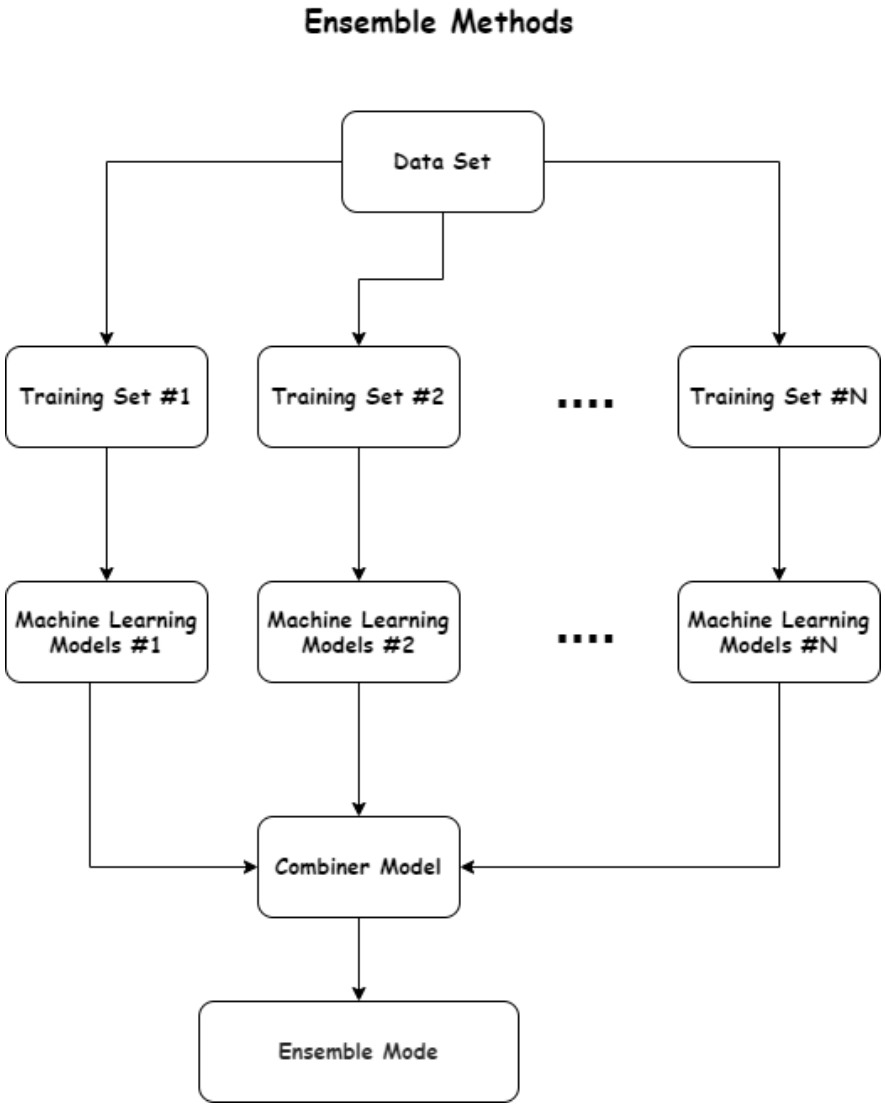


รูปที่ 2-8 ภาพโครงสร้าง Ensemble Method

Bagging เป็นลักษณะของการแยกการทำงาน แยกข้อมูลเป็นส่วน ๆ แล้วทำการเรียนรู้แยกกัน (Parallel method) เพื่อหาผลที่ได้ค่าที่ดีที่สุดในแต่ละตัวโดยยึดหลัก ค่าเฉลี่ย (Averaging) กับ ค่าที่พบมากที่สุด (Voting) แล้วจึงนำมารวมกัน ทำให้โมเดลที่ได้มีคุณสมบัติที่หลากหลายและดีที่สุด ตัวอย่างที่ใช้ในงานวิจัยในรูปแบบนี้คือ Random Forest

Boosting เป็นลักษณะของการรวมข้อเสีย ข้อผิดพลาด (Error) ของการเรียนรู้โมเดลจากการทำงานวนซ้ำไปเป็นรอบ ๆ โดยจะนำข้อเสีย ข้อผิดพลาดในแต่ละรอบมาร่วมปรับปรุงโมเดลด้วยเพื่อให้ได้ผลที่ดีที่สุด ตัวอย่างที่ใช้ในงานวิจัยในรูปแบบนี้คือ XGBoost

Stacking เป็นลักษณะการรวมโมเดลเข้าด้วยกันและการทำให้โมเดลเรียนรู้ต่อเนื่องและปรับปรุงคุณภาพเพิ่มขึ้นไป รูปแบบนี้คือการการเรียนแบบต่อเนื่อง เช่นเมื่อรอบแรกได้โมเดลมาเรียบร้อย รอบที่สองนำโมเดลรอบแรกมาทำงานเรียนรู้ต่อเนื่องทำให้เกิดโมเดลใหม่ ๆ ขึ้นมา



รูปที่ 2-9 ภาพตัวอย่างการทำงานของ Ensemble Method

Ensemble Method จากรูปด้านบนจะเป็นการแสดงให้เห็นการทำงานในวิธีดังกล่าว โดยไม่จำกัดจำนวนโมเดลที่ทำร่วมกัน

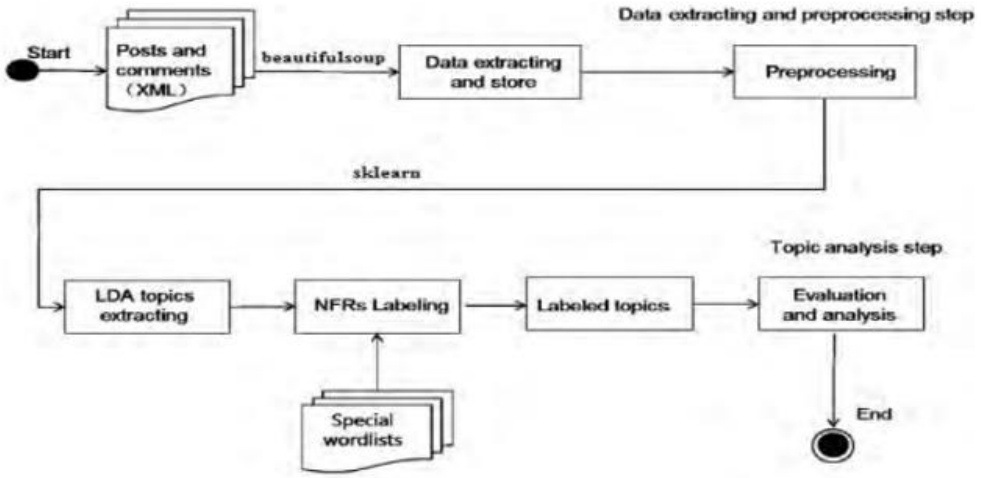
2.13 XGBoost

XGBoot ถือเป็นหนึ่งใน ensemble learning ชนิด Boost ที่พัฒนาโดยทีม The XGBoost Contributors จาก https://XGBoost.ai/ [12] สามารถทำงานร่วมกันได้หลายภาษาและหลาย ๆ เครื่องมือ โดยในงานวิจัยนี้ได้ใช้ร่วมกับภาษาไพธอน ชื่อเต็ม ๆ ของ XGBoost คือ eXtreme Gradient Boosting อัลกอลิทึ่มดังกล่าวเป็นตัวพัฒนาและปรับปรุงความสามารถมาจาก Gradient Boosting ให้สามารถทำงานได้ดีขึ้น เร็วขึ้น ใช้เวลาน้อยลง คุณสมบัติและความสามารถหลาย ๆ อย่างยังคงอ้างอิงของ Gradient Boosting แต่มีการปรับปรุงส่วนสำคัญคือ การปรับค่า Boosting

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

3.1 Toward Empirically Investigating Non-Functional Requirements of iOS Developers on Stack Overflow

งานวิจัยฉบับนี้ ได้มีการนำเสนอการค้นหาข้อมูลที่เกี่ยวกับ IOS ในฐานข้อมูล หรือเว็บไซต์ Stack Overflow โดยมีกระบวนการคือ การรวบรวมข้อมูลที่มี แท็ก(Tag) ที่เกี่ยวข้องกับ IOS โดยใน IOS ก็ได้มีการทำกระบวนการหาเนื้อหาสำคัญ ๆ ที่อยู่ภายใต้ IOS Tag ว่ามีการพูดถึงเรื่องอะไรบ้างโดยใช้วิธีการทำ Text Classification โดยใช้โมเดล LDA topic modelling ผลที่ได้จากการค้นหาจากโมเดลในงานวิจัยฉบับนี้ก็ยังทำการแยกและจัดกลุ่มในหมวดหมู่ของ Non-Functional Requirements ในด้านต่าง ๆ โดยภาพรวมงานเป็นไปตามรูปที่ 3-1



รูปที่ 3-1 ภาพรวมงานวิจัย [13]

ในวิทยานิพนธ์นี้มีการใช้ฐานข้อมูลสแต็กโอเวอร์โฟลว์เดียวกัน และการทำ LDA topic modelling เพื่อหากลุ่มข้อมูลที่มีความน่าสนใจแต่วิทยานิพนธ์นี้จะทำการจำแนกหมวดหมู่ของข้อมูลตามประเภทปัญหาโดยใช้วิธี Text Classification ร่วมด้วย

3.2 SOTagRec: A Combined Tag Recommendation Approach for Stack Overflow

งานวิจัยนี้ นำเสนอวิธีการพัฒนาเครื่องมือแนะนำการใส่ป้ายของข้อมูลในเว็บไซต์สแต็กโอเวอร์โฟลว์ สำหรับโพสต์ที่ไม่มีการใส่ป้ายของข้อมูล โดยในงานวิจัยฉบับนี้จะพูดถึงการนำข้อมูลที่มีป้ายของข้อมูลมาเรียนรู้โดยคอมพิวเตอร์ (Machine learning) โดยใช้กระบวนการ Deep Learning ซึ่งเป็นขั้นตอนวิธีในการแนะนำ(Recommendation) โดยในงานวิจัยนี้ได้ใช้ Deep Learning เป็น Convolutional neural network (CNN) และ Collaborative filtering (CF) โดยใช้ทั้ง 2 โมเดลรวมกันทำ Combined Model ตามรูปที่ 3-2 เพื่อใช้ในการแนะนำป้ายของข้อมูลที่เกี่ยวข้องกับโพสต์ข้อมูลบนเว็บไซต์สแต็กโอเวอร์โฟลว์ งานวิจัยดังกล่าวสามารถนำข้อมูลที่มีป้ายของข้อมูลมาสร้างรูปแบบผ่านโมเดลทั้งสองเพื่อสร้างระบบและโมเดลในการแนะนำป้ายข้อมูลสำหรับโพสต์ที่ไม่มีป้ายของข้อมูลได้แม่นยำถึง 80 %

วิทยานิพนธ์นี้ได้มีการใช้แหล่งข้อมูลที่เหมือนกันกับงานวิจัยที่ยกมาคือสแต็กโอเวอร์โฟลว์ และในวิทยานิพนธ์นี้มีการใช้โมเดลการเรียนรู้ของเครื่องคือ Convolutional neural network (CNN) เช่นเดียวกัน แต่การจัดหมวดหมู่ของโพสต์จะจัดตามประเภทปัญหา ไม่ใช่การจัดการตาม Tag ที่มีอยู่ในสแต็กโอเวอร์โฟลว์

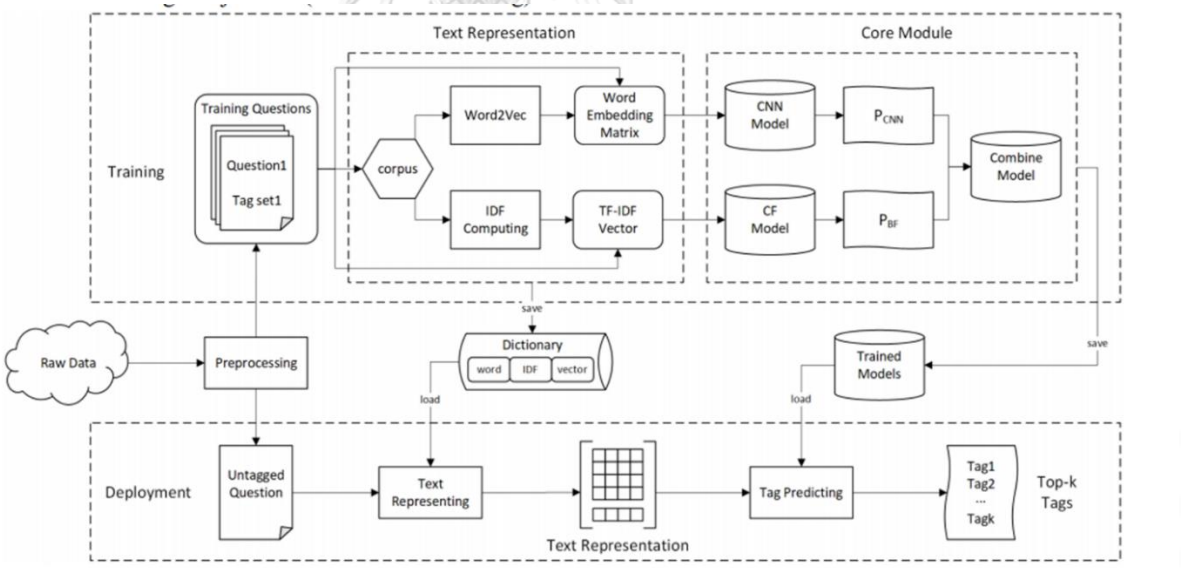
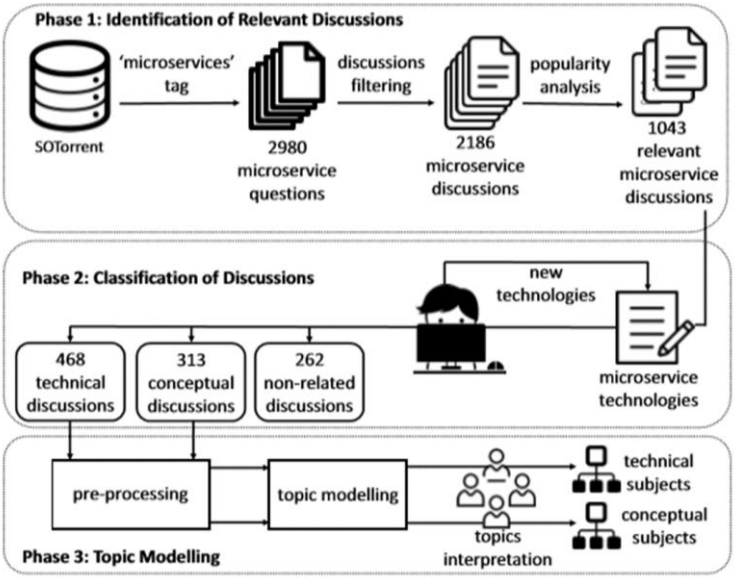


Figure 1. Overall framework of SOTagRec.

รูปที่ 3-2 ภาพรวมของงานวิจัย SOTagRec [14]

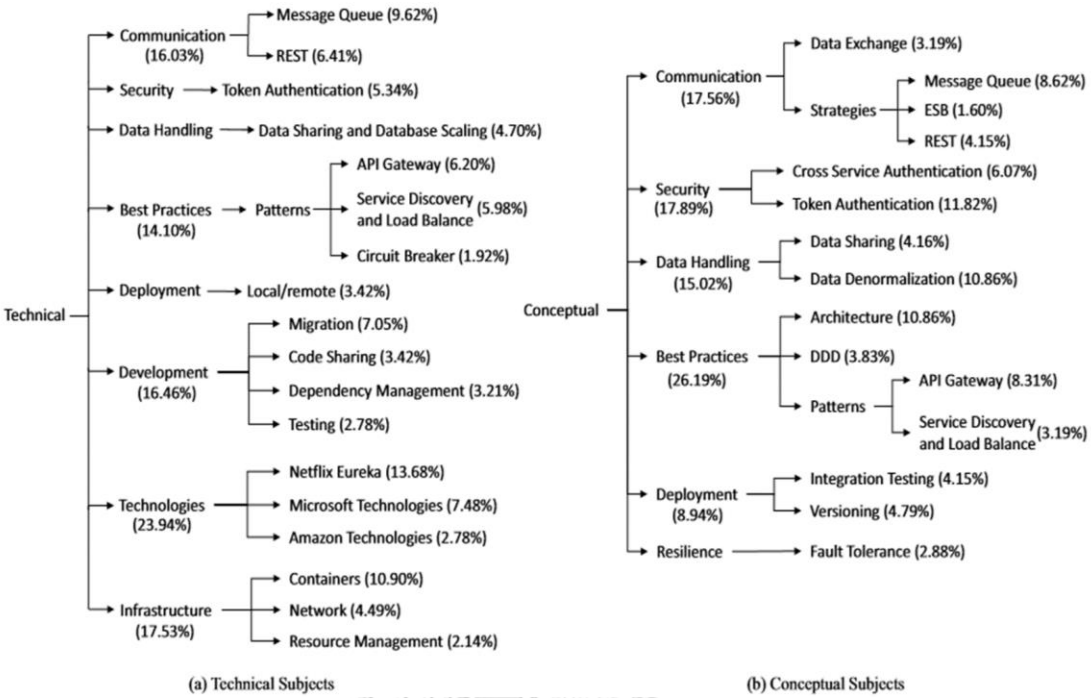
3.3 We Need to Talk about Microservices: an Analysis from the Discussions on Stack Overflow

งานวิจัยนี้ นำเสนอปัญหาที่ต้องการตรวจสอบการพูดถึงเกี่ยวกับแนวคิดเรื่องของ Microservice บนเว็บไซต์สแต็กโอเวอร์โฟลว์พร้อมทั้งวิธีการเก็บข้อมูลจากเว็บดังกล่าว โดยในงานชิ้นนี้ได้มีวิธีการคือเก็บข้อมูลมาในหัวข้อที่พูดถึง Microservice และมีการจัดแบ่งกลุ่มเป็นกลุ่มใหญ่ ๆ 2 กลุ่มคือ technical และ conceptual การทำงานแบ่งเป็น 3 ส่วนตามรูปที่ 3-3



รูปที่ 3-3 ภาพรวมของงานวิจัย [15]

ส่วนที่ 1 เป็นส่วนของการรวบรวมข้อมูลซึ่งในที่นี้ใช้ข้อมูลจากสแต็กโอเวอร์โฟลว์จากนั้นทำการคัดแยกข้อมูลที่ต้องการใช้โดยมีหลักการในการเลือกคือเป็น Tag Microservices ส่วนที่ 2 เป็นการจัดกลุ่ม Classification โดยมนุษย์แยกย่อยเป็น 3 กลุ่มเพื่อเตรียมข้อมูลก่อนนำส่งไปสู่ส่วนที่ 3 ส่วนที่ 3 เป็นการทำ topic modelling พร้อมทั้งมีการตีความหัวข้อที่ได้ด้วยมนุษย์เพื่อจัดกลุ่มเนื้อหาที่มีการพูดถึงในเว็บไซต์ดังกล่าวตามหัวข้อตามรูปที่ 3-4



รูปที่ 3-4 กลุ่มต่าง ๆ หลังจากการตีความหัวข้อที่ได้จาก topic modeling [15]

วิทยานิพนธ์นี้มีแนวคิดในการแบ่งกลุ่มและใช้แหล่งข้อมูลสแต็กโอเวอร์โฟลว์เหมือนกัน พร้อมทั้งวิธีการจัดกลุ่มแบบสองระดับ เช่น Technical ในระดับที่ 1 Infrastructure ในระดับที่ 2 เป็นต้น และหัวข้อ Development ก็ถูกยกมาหัวข้อที่ต้องการจำแนกในวิทยานิพนธ์นี้เช่นกัน

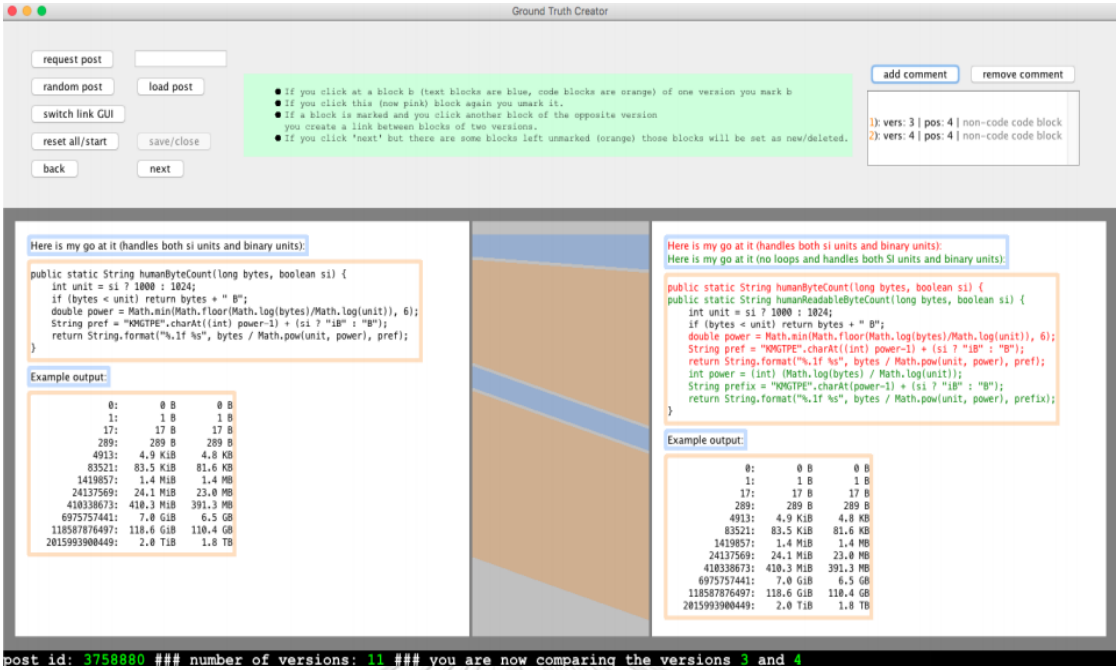
3.4 SOTorrent : Reconstructing and Analyzing the Evolution of Stack Overflow Posts

งานวิจัยนี้ [16] นำเสนอการเพิ่มการปรับปรุงแนวโน้มหรือลักษณะการปรับปรุงของโพสต์ในสแต็กโอเวอร์โฟลว์ ในงานชิ้นนี้ได้ใช้แหล่งข้อมูลจาก SOTorrent และข้อมูลมีขนาดใหญ่พร้อมข้อมูลถูกปรับปรุงตลอดเวลา วิธีคิดของงานชิ้นนี้จะทำการสำรวจเนื้อหาในส่วนต่าง ๆ ของสแต็กโอเวอร์โฟลว์โดยจะแยกส่วนตามรูปที่ 3-5

รูปที่ 3-5 ตัวอย่างโครงสร้างเนื้อหาในสแต็กโอเวอร์โฟลว์ [16]

ส่วนที่เน้นที่สุดจะเป็นส่วนของ Code Block ซึ่งเป็นส่วนที่ก่อให้เกิดความเปลี่ยนแปลงตลอดเวลา เช่น ในวันที่ตอบคำถามจะใช้เทคโนโลยีรุ่นหนึ่ง ผ่านไปอีก 2-3 ปี อาจจะมีการกลับมาแก้ไขบางส่วนเพื่อให้สอดคล้องกับยุคปัจจุบัน ซึ่งหลังจากผ่านกระบวนการนับแล้วได้ผลว่า 6% ของการแก้ไขทั้งหมดจะมีการเปลี่ยนแปลงโค้ด (Code block) แต่จะไม่มีการเปลี่ยนแปลงอัปเดตบล็อกข้อความ (Text block) 78% จะมีการแก้ไขโพสต์ในวันเดียวกับที่ลงโพสต์และ 87% ของการแก้ไขทำโดยผู้เขียนโพสต์โดยมีตัวอย่างการเปรียบเทียบตามรูปที่ 3-6

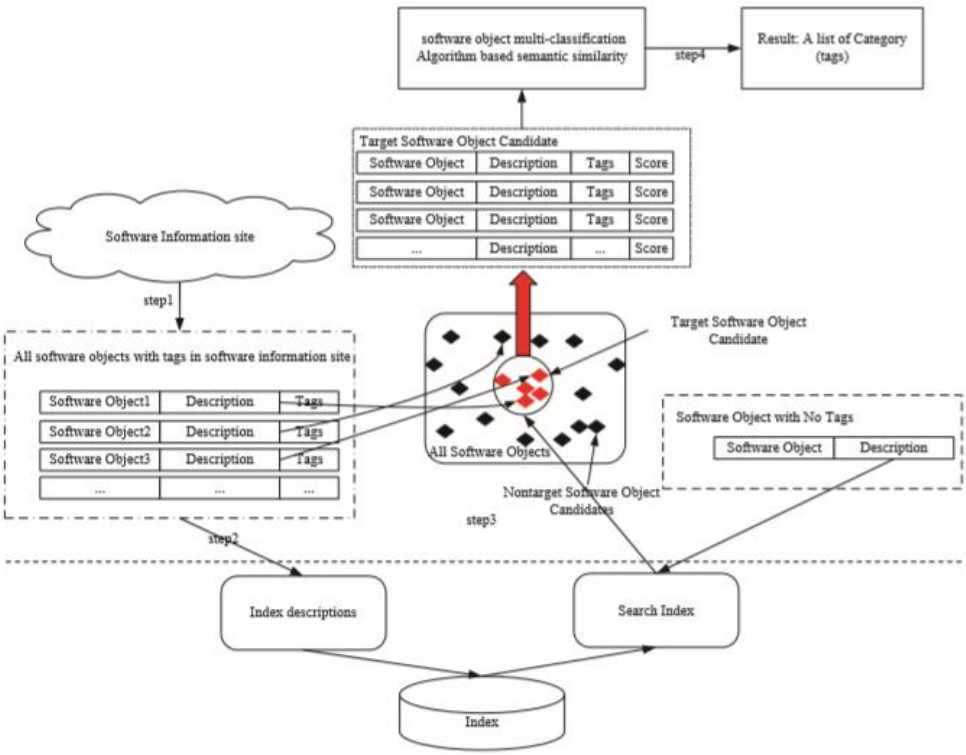
ในงานวิจัยนี้จะอธิบายมุมมองการใช้ข้อมูลจาก SOTorrent แนวคิดหลักในการจัดเก็บข้อมูล รูปแบบข้อมูล และทำให้เห็นปัญหา และแนวทางการเข้าถึงข้อมูลหรือการจัดเก็บรวบรวมข้อมูลเพื่อมาปรับปรุงและเรียนรู้ในงานวิทยานิพนธ์ฉบับนี้



รูปที่ 3-6 ตัวอย่างการเปรียบเทียบเพื่อดูวิวัฒนาการของเนื้อหา [16]

3.5 Scalable Tag Recommendation for Software Information Sites

งานวิจัยนี้ นำเสนอปัญหาการใช้งานแท็กไม่ถูกต้อง หรือไม่เข้าใจแต่ละชื่อแท็กโดยมีการนำเสนอเครื่องมือที่เรียกว่า TagMulRec เพื่อแนะนำแท็กโดยอัตโนมัติแบบหลายจำนวนพร้อม ๆกัน และช่วยจัดประเภทปัญหาข้อมูลซอฟต์แวร์ขนาดใหญ่ TagMulRec ได้มีการทดลองกับ เว็บไซต์ข้อมูลซอฟต์แวร์สี่แห่งคือ Stack Overflow, AskUbuntu, AskDifferent และ Freecode และนำผลที่ได้มาสร้างดัชนีสำหรับการค้นหาข้อมูลเพิ่มเติม ในงานวิจัยนี้ได้ใช้วิธีการคำนวณแบบ Similarity Score Computation สำหรับการช่วยในการแนะนำโดยมีวิธีการคือการวัดความถี่และน้ำหนักของคำ ในภาพรวมของงานจะมีกระบวนการตามรูปที่ 3-7



รูปที่ 3-7 ภาพรวมของ TagMulRec [17]

สำหรับงานวิจัยนี้ได้มีการออกแบบขั้นตอนและกระบวนการสร้างระบบค้นหาข้อมูลที่มีข้อมูลในลักษณะเดียวกับวิทยานิพนธ์ฉบับนี้ โดยแนวคิดดังกล่าวได้นำมาปรับใช้ในการทำระบบสำหรับการค้นหาข้อมูลและสร้างรูปแบบการจัดเก็บข้อมูล เพิ่มเติมสำหรับข้อมูลที่แบ่งกลุ่มตามแท็กเรียบร้อยแล้ว แต่ข้อมูลจะเน้นในส่วนของเทคโนโลยีฐานข้อมูลเป็นหลัก

บทที่ 4
ภาพรวมงานวิจัย

งานวิจัยนี้ได้มีแนวคิดริเริ่มจากการแสดงผลการค้นหาจากเว็บไซต์สแต็กโอเวอร์โฟลว์ที่ต้องการค้นหาปัญหาเป็นกลุ่มต่าง ๆ ในแต่ละเทคโนโลยี เพื่อให้ผู้พัฒนาเจ้าของเทคโนโลยีต่าง ๆ สามารถรับข้อมูล ตรวจสอบ จากผู้ใช้งานจริงว่าเทคโนโลยีดังกล่าวมีข้อผิดพลาดในส่วนใดบ้างหรือต้องปรับปรุงอะไรบ้างและจากงานวิจัยที่เกี่ยวข้องเห็นได้ว่าได้มีการนำเสนอวิธีการช่วยเหลือการจัดหมวดหมู่กลุ่มในรูปแบบต่าง ๆ แต่ไม่มีในรูปแบบที่เป็นแง่ของการรวบรวมข้อมูลเทคโนโลยีเพื่อมาจำแนกปัญหาหรือข้อจำกัด ดังนั้นงานวิจัยชิ้นนี้จึงมีภาพรวมที่จะพัฒนาวิธีการและเครื่องมือเพื่อแก้ปัญหาดังกล่าวโดยภาพรวมงานวิจัยเป็นดังรูปที่ 4-1 โดยจะแยกเป็น 2 ส่วน คือ

1). ส่วนการพัฒนาโมเดลการเรียนรู้ของเครื่อง (Training)

ประกอบด้วยขั้นตอนดังต่อไปนี้

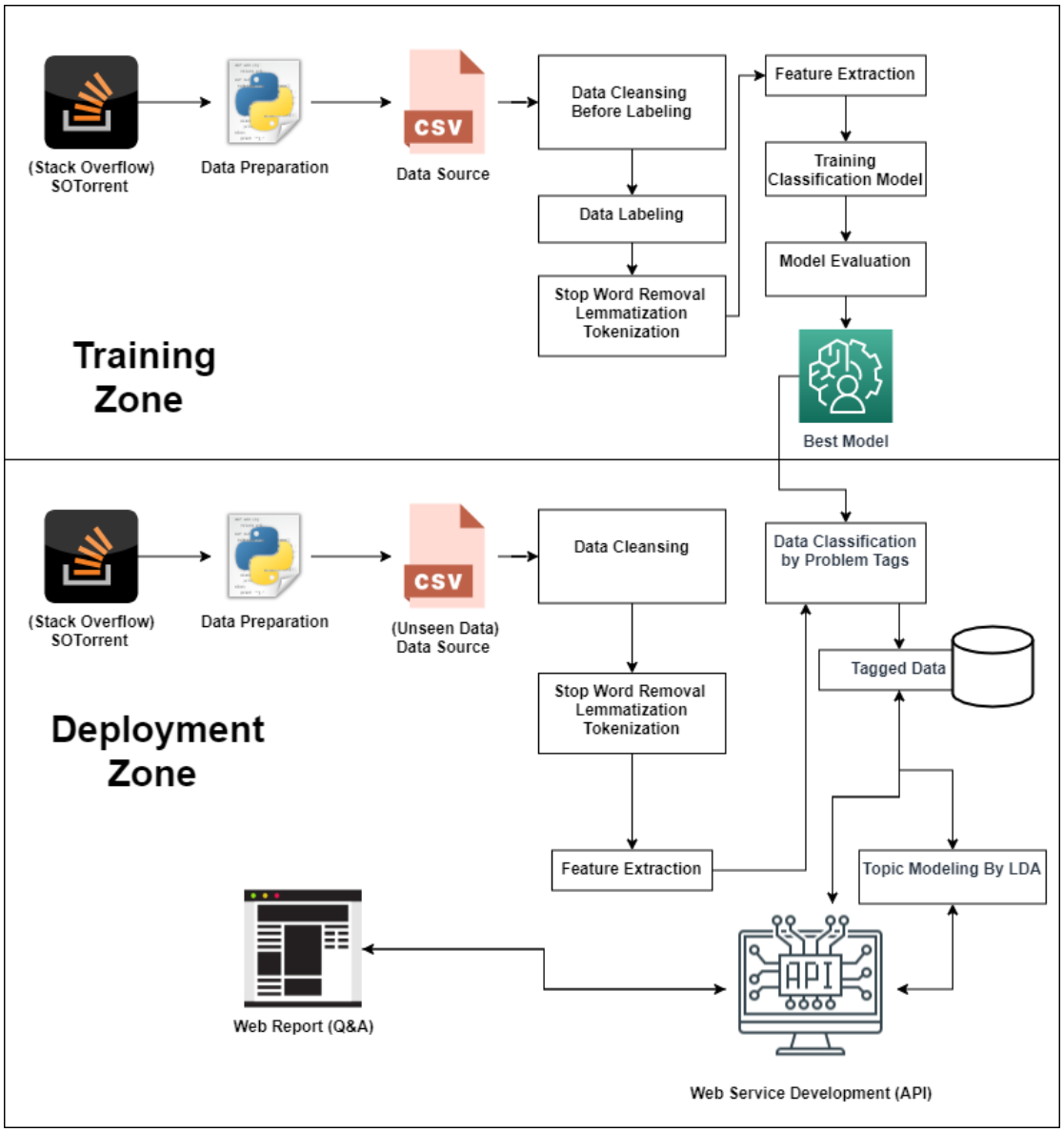
- โหลดข้อมูลที่ต้องการใช้งาน
- แปลงข้อมูลเพื่อสามารถจัดและนำไปใช้งานต่อ
- ทำความสะอาดข้อมูล (เฉพาะส่วนที่จำเป็นในการอ่านง่ายขึ้น)
- ติดป้ายข้อมูล
- ทำความสะอาดข้อมูล (สำหรับเตรียมการสร้างโมเดล)
- ทำกระบวนการ feature extraction
- ทำการสร้างโมเดลด้วยอัลกอริทึมการเรียนรู้ของเครื่อง
- ประเมิน ปรับปรุงโมเดล

2). ส่วนการนำไปใช้งาน (Deployment)

ประกอบด้วยขั้นตอนดังต่อไปนี้

- โหลดข้อมูลที่ต้องการใช้งาน (ไม่ซ้ำกับข้อมูลที่ใช้สร้างโมเดล)
- แปลงข้อมูลเพื่อสามารถจัดและนำไปใช้งานต่อ
- ทำความสะอาดข้อมูล
- ทำกระบวนการ feature extraction
- นำข้อมูลผ่านโมเดลในการแยกปัญหา 9 กลุ่ม
- สร้างโมเดล Topic Modeling ด้วยการทำ LDA

- ทำการเชื่อมต่อเว็บ API และ หน้าจอแสดงผลผ่านระบบเว็บ



รูปที่ 4-1 ภาพรวมงานวิจัย

สำหรับข้อมูลในงานวิจัยนี้ เริ่มจากการค้นหาแหล่งข้อมูลการสอบถามปัญหาการใช้งานเทคโนโลยีฐานข้อมูลว่าในโลกปัจจุบันมีที่ใดที่รวบรวมไว้บ้าง ซึ่งพบว่าเว็บไซต์สแต็กโอเวอร์โฟลว์ที่เป็นกระดานถามตอบปัญหา มีการพูดถึงปัญหาในเรื่องดังกล่าวเป็นจำนวนมาก ทางผู้วิจัยจึงได้ทำการสำรวจและรวบรวมข้อมูลจากเว็บไซต์ดังกล่าวเพื่อทำการวิจัยในลำดับถัดไป

สำหรับเว็บไซต์สแต็กโอเวอร์โฟลว์นั้น เป็นเว็บไซต์ที่ทำงานเป็นหน้ากระดานถามตอบ และสืบค้นสาธารณะ ให้ผู้ใช้งานสามารถสร้างหัวข้อ สร้างเนื้อหา ตั้งคำถาม หรือข้อสงสัย แม้กระทั่งแสดงความเห็นในเรื่องต่าง ๆ ซึ่งในที่นี้งานวิจัยนี้ได้เน้นไปที่ปัญหาที่เกี่ยวข้องกับงานเทคโนโลยีฐานข้อมูล

ในขั้นตอนการรวบรวมได้พบว่ามีแหล่งข้อมูลที่ได้ทำการจัดเก็บไว้ชื่อ SOTorrent ซึ่งเป็นการรวบรวมข้อมูลของเว็บไซต์สแต็กโอเวอร์โฟลว์ จากระบบ Stack Exchange Data Dump ทำให้มีข้อมูลที่ต้องการครบพร้อมใช้งานและมีการจำแนกจัดเก็บไว้หลากหลายรูปแบบ เช่น google cloud big query หรือ แม้กระทั่ง XML ไฟล์ เป็นต้น

ลำดับถัดมาทางผู้วิจัยได้ทำการดึงข้อมูลแยกเพื่อรวบรวมในส่วนที่จะมาวิจัยเท่านั้น โดยใช้เกณฑ์การแยกตามชื่อยี่ห้อของผลิตภัณฑ์เทคโนโลยีฐานข้อมูลจำนวน 5 ชื่อ ประกอบด้วย

1). mysql 2). oracle 3). postgresql 4). mongodb 5). sqlserver ซึ่งเป็นข้อมูล 5 ปีย้อนหลัง จำนวน 13,000 ชุดหรือโพสต์

รูปที่ 4-2 ตัวอย่างหน้าจอเว็บไซต์ และส่วนของข้อมูลที่นำมาใช้งาน

สำหรับส่วนของข้อมูลที่นำมาใช้ในงานวิจัยจากรูปที่ 4-2 ด้านบน จะมีการดึงมาเฉพาะส่วนตามกรอบ เท่านั้น โดยจะมีการนำข้อมูลที่ใช้คือ หัวข้อปัญหาหรือชื่อของโพสต์ และเนื้อหาของโพสต์ ในงานวิจัยนี้ได้วางเป้าหมายแรกคือการจำแนกข้อมูลจากสองส่วนนี้เท่านั้นว่าเป็นปัญหาในมุมใด เรื่องอะไรเท่านั้น ในอนาคตสามารถต่อยอดงานวิจัยในส่วนของคำตอบในโพสต์ได้หรือระดับการมีส่วนร่วมในโพสต์การจัดเก็บข้อมูลในขั้นต้นได้มีการทำการจัดเก็บข้อมูลดังกล่าวไว้ในรูปแบบของไฟล์นามสกุล CSV เพื่อให้ง่ายต่อการนำไปใช้งานและประมวลผลในลำดับถัด

บทที่ 5
การจำแนกปัญหาของระบบฐานข้อมูล

ในบทนี้จะพูดถึงขั้นตอนและวิธีการที่งานวิจัยดำเนินการโดยละเอียด โดยมีขั้นตอนเรียงตามลำดับดังนี้

5.1 การรวบรวมสำรวจปัญหาและแยกกลุ่มเทคโนโลยี

ในเครือข่ายเว็บถามตอบปัญหาเทคโนโลยีสแต็กโอเวอร์โฟลว์มีลักษณะการทำงานที่เหมือนการถามตอบข้อสงสัยทั่วไปแต่อยู่ในรูปแบบของสื่อเว็บไซต์ ในเว็บดังกล่าวมีการแบ่งแยกชนิดของเทคโนโลยีจากฟังก์ชันการทำงานที่มีชื่อว่าป้ายของข้อมูลซึ่งป้ายของข้อมูลจะทำหน้าที่ในการสร้างแท็กหรือคำสั้น ๆ ที่ใช้บ่งบอกว่าเนื้อหาว่าด้วยเรื่องอะไรหรือเกี่ยวข้องอะไรกับกลุ่มคำดังกล่าว โดยกลุ่มคำดังกล่าวเป็นลักษณะของคำที่ใช้ในการค้นหา (keyword) แต่คำค้นหาดังกล่าวหรือหมวดหมู่ดังกล่าวเกิดขึ้นจากผู้ถามคำถามเป็นผู้กำหนดขึ้นมาเพื่อให้ผู้ที่มีความรู้หรือสามารถตอบคำถามในเรื่องราวดังกล่าวค้นหาได้ง่ายและเพียงพอสำหรับการแบ่งกลุ่มชื่อเทคโนโลยีเท่านั้น แต่ไม่สามารถบ่งบอกได้ว่าในการถามคำถามอยู่ในบริบทอะไรบ้าง หรือลักษณะของคำถามเป็นรูปแบบใด

จากลักษณะดังกล่าวในข้างต้น พบว่าปัญหาที่เกิดขึ้นส่วนใหญ่นั้นมักเกี่ยวข้องกับวิธีการใช้งาน การติดตั้ง และการนำไปใช้งานต่อให้มีประสิทธิภาพมากขึ้น และบ่อยครั้งเป็นการถามคำถามที่มีการทำงานที่คล้ายกัน แต่เป็นการนำไปใช้คนละรูปแบบ

ในงานวิจัยฉบับนี้ได้แยกลักษณะของกลุ่มปัญหาไว้ตามตารางที่ 5-1

ตารางที่ 5-1 ลักษณะของกลุ่มปัญหา

ลักษณะของกลุ่มปัญหา	
ชื่อลักษณะ	**คำนิยาม (Descriptions)**
Installation	การติดตั้งและการตั้งค่า หมายถึง“install a system or component, set initial parameters, and prepare the system or component for operational use” [18] โดยมีกลุ่มคำที่จัดอยู่ในกลุ่มนี้เช่น install, setup, config, setting, Installation และชนิดของ Installation เช่น Attended installation, Silent installation, Unattended installation, Headless installation, Scheduled or automated installation, Clean installation, Network installation
Development	การพัฒนา หมายถึง “specification, construction

ลักษณะของกลุ่มปัญหา	
ชื่อลักษณะ	**คำนิยาม (Descriptions)**
(Software development)	testing and delivery of a new application or of a discrete addition to an existing application" [18] โดยมีกลุ่มคำที่จัดอยู่ในกลุ่มนี้เช่น dev, build, coding, implement
Performance Tuning	การปรับปรุงประสิทธิภาพ หมายถึง "The art of increasing performance for a specific application set."[19],[20] โดยมีกลุ่มคำที่จัดอยู่ในกลุ่มนี้ เช่น Performance Tuning, Tuning, speed, slows, load, Optimization, Optimize

ในลักษณะของกลุ่มปัญหาข้างต้นยังแยกเป็นลักษณะของปัญหาย่อย ในตารางที่ 5-2

ตารางที่ 5-2 ลักษณะของปัญหาย่อย

ลักษณะของปัญหาย่อย	
ชื่อลักษณะ	**คำนิยาม (Descriptions)**
Limitation	ข้อจำกัดและขีดจำกัดทั้งใน Software และเอกสาร หมายถึง "the act of controlling and especially reducing something" [20],[21] โดยมีกลุ่มคำที่จัดอยู่ในกลุ่มนี้เช่น Limitation, restriction, boundary, limit, confine, restriction, stinginess, localization, constraint, confinement, limiting
Design	การออกแบบและการวางแผน หมายถึง"Design is defined as both "the process of defining the architecture, components, interfaces, and other characteristics of a system or component" and "the result of that process" [20] โดยมีกลุ่มคำที่จัดอยู่ในกลุ่มนี้เช่น design, style, plan, layout, scheme, diagram, pattern, format, outline, framework, model, architecture, architect [22]
Discussion	การอภิปราย แนะนำ สอบถาม ในเรื่องอื่น ๆ ที่ไม่เกี่ยวข้องกับเรื่อง Limitation และ Design

5.2 การจัดเตรียมข้อมูลสำหรับการแยกกลุ่มปัญหาด้วยผู้เชี่ยวชาญ

ในการรวบรวมข้อมูลจากสแต็กโอเวอร์โฟลว์ในงานวิจัยนี้ได้เลือกข้อมูลจากแหล่งที่มาของเว็บไซต์ SOTorrent ที่มีการเก็บข้อมูลจากสแต็กโอเวอร์โฟลว์ไว้โดยข้อมูลที่ใช้จะใช้ข้อมูลย้อนหลัง 2 ปี คือ 2018 ถึง 2019 ในการเรียนรู้ และในกรณีที่ผลของการทำโมเดลไม่ได้ตามที่คาดหวังไว้ จะทำการเพิ่มข้อมูลเป็นปี 2015 ถึง 2019 แทน

แท็ก (tags) สำหรับใช้ในการเลือกข้อมูลมาประมวลผลมีทั้งหมด 5 Tags ตามชื่อของฐานข้อมูล ได้แก่ Mysql, mongodb, postgresql, oracle และ sqlserver ซึ่งมีจำนวนทั้งสิ้น 13,000 โพสต์ สำหรับ 5 Tags โดยในฐานข้อมูลที่มีการเก็บมาจำเป็นต้องทำการทำความสะอาดข้อมูลและเตรียมข้อมูลก่อนนำไปใช้งานซึ่งมีขั้นตอนดังนี้

- Data load

กระบวนการนี้จะทำการเรียกข้อมูลจากฐานข้อมูล SOTorrent ที่เป็นเวอร์ชันล่าสุดเข้าไปในฐานข้อมูลที่ได้จัดเตรียมไว้ ในส่วนของข้อมูลจาก SOTorrent จะมาจาก Google Bigquery โดยแปลงออกมาเป็นไฟล์นามสกุล CSV เพื่อช่วยในการค้นหาสิ่งที่น่าสนใจในข้อมูลเบื้องต้น

รูปที่ 5-2 ข้อมูลบนเว็บไซต์สแต็กโอเวอร์โฟลว์

จากรูปที่5-2 เป็นตัวอย่างข้อมูลจริงจากหน้าเว็บไซต์สแต็กโอเวอร์โฟลว์และรูปที่ 5-3ด้านล่างมาจาก SOTorrent ซึ่งทั้งสองเป็นข้อมูลเดียวกัน

2019-08-2	How to use regex and match data using find/aggregate?	<javascript><node.js><mongodb><mongodb-c
2019-08-3	Which dependencies would i need to use in a social media app when I want to implement a feature to follow	<node.js><reactjs><mongodb><express>
2019-09-0	how to install mongodb compass from stable version onto windows 10	<mongodb><mongodb-compass>
2019-08-2	How can I stream a map of polylines efficiently?	<javascript><mongodb><streaming><query-op
2019-08-3	How to store data in Firebase (or other nosql document database) taking Max Document Size Into Account?	<database><mongodb><firebase><firebase-r
2019-08-2	How can I get a CSV from my atlas mongodb cluster?	<java><mongodb><atlas>

รูปที่ 5-3 ตัวอย่างข้อมูลที่ได้จาก SOTorrent

ตัวอย่างข้อมูลที่ได้จาก SOTorrent แสดงดังรูปที่ 5-4

2	Difficulties to iterate JS objects attributes with mongo shell interpreter	<mongodb>	0	1
2	Can we store drool rules (XLS) in mongodb database?	<java><mongodb><drools><hazelcast><rule-e	0	1
2	Put a condition in Project statement	<mongodb><mongoose><aggregation-framev	0	1
3	Passing MongoDB collections to modularized Express routes	<javascript><node.js><mongodb><express>	0	1
2	Getting BSON converter issue while saving message from Kafka topic to MongoDB using Kafka Connect	<mongodb><apache-kafka><apache-kafka-co	0	1
2	why mongoos need schema to return model instance which already exist?	<node.js><mongodb><mongoose>	0	1
2	Init scripts not executing on MongoDB docker containers	<mongodb><docker>	0	1
2	How to design mongodb dataset for opening hours datamodel	<mongodb>	0	1
3	Spring MongoDB $in with array and embedded document	<spring><mongodb><mongodb-query><spring	0	1
1	API testing problem with Postman the data is not being posted correctly	<node.js><mongodb><express><mongoose><	0	1
2	What is the best way to save user settings for web app	<javascript><angularjs><database><mongodl	0	1
2	How to use regex and match data using find/aggregate?	<javascript><node.js><mongodb><mongodb-c	0	1
3	Which dependencies would i need to use in a social media app when I want to implement a feature to follow	<node.js><reactjs><mongodb><express>	0	2
0	how to install mongodb compass from stable version onto windows 10	<mongodb><mongodb-compass>	0	2
2	How can I stream a map of polylines efficiently?	<javascript><mongodb><streaming><query-op	0	2
3	How to store data in Firebase (or other nosql document database) taking Max Document Size Into Account?	<database><mongodb><firebase><firebase-r	0	2
2	How can I get a CSV from my atlas mongodb cluster?	<java><mongodb><atlas>	0	2
2	How can I map collection name to WiredTiger URL without db.collection.stats()	<mongodb><wiredtiger>	0	2
3	MongoDB $lookup with <collection to join> coming from the input document	<mongodb><mongodb-query><aggregation-fr	0	2
0	Failed to retrieve all objects on the database	<node.js><mongodb>	0	2
3	How to store downloaded file local url in mongodb	<mongodb><react-native>	0	2
3	Routing issue in React	<reactjs><mongodb><terminal><react-router-	0	2
2	Find a document using expression tree and nested BsonDocument	<c#><mongodb><mongodb-.net-driver>	0	2
2	How to access Mongodb remotely?	<mongodb>	0	2
2	http url for post req from android to mongodb via nodejs	<android><node.js><mongodb><http>	0	2
2	Creating a PHP wrapper for MongoDB and MysqlDB X DevApi	<php><mysql><mongodb><nosql><document	0	2
1	MongoDB: latest of multiple causally consistent sessions	<mongodb><consistency>	0	2

รูปที่ 5-4 ตัวอย่างข้อมูลที่ได้จาก SOTorrent

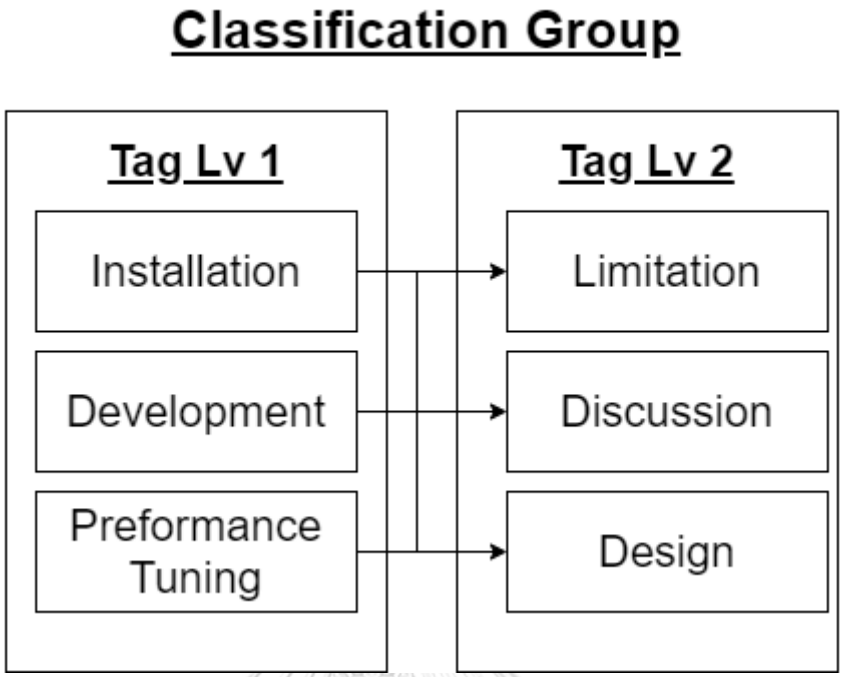
- Data Cleansing สำหรับ Labeling

ในกระบวนการนี้จะทำความสะอาดข้อมูลที่ได้จากข้อมูลดิบ โดยจะทำการลบข้อมูลที่ไม่ได้ใช้งาน และที่เป็นตัวอักขระพิเศษออกไป เช่น & # * ! เป็นต้น ในกระบวนการนี้จะทำการทำความสะอาดขั้นต้นเพื่อให้ผู้เชี่ยวชาญสามารถอ่านข้อมูลข้อความได้ง่ายขึ้นเท่านั้น ไม่สามารถนำไปประมวลผลการสร้างโมเดลต่อได้จำเป็นต้องทำความสะอาดเพิ่มเติม

- Data Labeling

ในกระบวนการติดป้ายข้อมูลจะเป็นกระบวนการลงทะเบียนข้อมูลเพื่อให้ระบบประมวลผลสามารถรับรู้ได้ว่าเป็นข้อมูลชนิดใดและกลุ่มใด โดยกระบวนการดังกล่าวจะใช้อาสาสมัครที่มีประสบการณ์หรือเชี่ยวชาญด้านฐานข้อมูลเป็นเวลา 1-5 ปี จำนวน 5 ท่านในการติดป้ายข้อมูล โดยจะเป็นผู้ที่เคยใช้งาน ผู้ที่เคยค้นหาหรือถามตอบคำถามในเว็บไซต์สแต็กโอเวอร์โฟลว์ ซึ่งทั้ง 5 ท่านจะใช้นิยามเดียวกันจากตารางในหัวข้อที่ 5.1 เพื่อให้มีความหมายและความเข้าใจที่ตรงกัน

การติดป้ายจะจำแนกข้อมูลเป็น 9 กลุ่มโดยแยกเป็น 2 ชั้น ตามรูปที่ 5-5 ชั้นแรกจะจำแนกข้อมูลออกเป็น Installation, Development, Performance Tuning โดยทั้งสามเป็นปัญหาที่เกี่ยวกับกิจกรรมหลักสำหรับการใช้งานเทคโนโลยีและชั้นที่สองจะจำแนกข้อมูลตามปัญหาย่อยที่ประกอบด้วยเรื่อง Limitation, Discussion, Design ตัวอย่างของการติดป้ายแสดงดังรูปที่ 5-6



รูปที่ 5-5 Classification Group

NO	Post	Content	Tech Tag	Label Lv1	Label Lv2
1	"CAST" function with "DISTIN	I have two tables parent and child .	postgresql	Development	Design
2	"create is not valid input at th	I have just installed mysql-workbenc	mysql	Installation	Discussion
3	"createlang: command not fo	I am trying to get musicbrainz datab	postgresql	Development	Discussion
4	"ERROR! MySQL server PID f	I notice that a <code>mysqld</cod	mysql	Installation	Discussion
5	"From" multiple tables and th	I have this query that works beautifi	mysql	Development	Discussion
6	"GRANT SELECT table TO ro	I have very simple db (PostgreSQL)	postgresql	Development	Discussion
7	"insert data out of date range	I'm getting the following error while i	postgresql	Development	Discussion
8	"insufficient privileges" error c	I have installed Oracle 11g on my P	oracle	Installation	Discussion
9	"MongoDB not able to start"	Osmoxis error Log</p>
&#xA	mongodb	Development	Discussion
10	"mount" a PostgreSQL datab	I've been given a project to extract	postgresql	Installation	Discussion
11	"Reverse Join" needed, using	I need to do whast I describe as a "	mysql	Installation	Design
12	"row is too big (...) maximum	I'm facing weird issue with postgres	postgresql	Development	Design
13	"Subquery returned more tha	have two tables from these first tw	sql-server	Development	Design
14	"Subquery returns more than	select tf.id from text_fields as tf WH	mysql	Development	Discussion

รูปที่ 5-6 ตัวอย่างการติดป้ายกับข้อมูล

เนื่องจากกระบวนการดังกล่าว เป็นกระบวนการที่ใช้ผู้เชี่ยวชาญ ดังนั้นจึงมีโอกาสเกิดกรณีที่มีความเห็นไม่สอดคล้องกันในเนื้อหา ทางแก้ไขคือ จะมีการพูดคุยกันระหว่างผู้ติดป้าย เพื่อหามติในการหาทางออก การแบ่งข้อมูลเพื่อการติดป้าย ไม่ได้ใช้วิธีการแบ่งข้อมูลแบบเท่า ๆ กัน แต่จะใช้วิธีการแบ่งตามสัดส่วนที่ผู้ติดสามารถทำได้ ระหว่าง 1,000 ถึง 3,000 ชุด โดยมีการคละชื่อผลิตภัณฑ์ฐานข้อมูล ไม่ได้เจาะจงให้ทำในผลิตภัณฑ์เดียว เพื่อให้ผลที่ได้มีความหลากหลายมากขึ้น

หลังจากผู้ติดป้ายได้ทำการติดป้ายมาเรียบร้อยลำดับถัดไปได้มีการรวบรวมข้อมูลและสรุปจำนวน ตามแต่ละกลุ่มของปัญหาตามตารางที่ 5-3

ตารางที่ 5-3 ตารางสรุปจำนวนข้อมูลแยกตามปัญหาและตามชื่อผลิตภัณฑ์ฐานข้อมูล

Tags	mysql	oracle	mongodb	postgresql	sqlserver
Development-Design	340	2	230	1283	1544
Development-Discussion	1514	18	1267	842	30
Development-Limitation	268	0	193	312	647
Installation-Design	195	40	70	93	219
Installation-Discussion	228	81	188	45	292
Installation-Limitation	223	1	59	56	91
Performance-Tuning-Design	485	1	134	159	164
Performance-Tuning - Discussion	722	4	293	43	51
Performance-Tuning - Limitation	447	0	245	48	80

บทที่ 6
การสร้างโมเดลการจำแนกประเภทปัญหาของระบบฐานข้อมูล

ในกระบวนการสร้างโมเดลการจำแนกประเภทปัญหาต่าง ๆ ของงานวิจัยฉบับนี้มีขั้นตอนแยกย่อย 4 ส่วนสำคัญคือ

1). การทำความสะอาดข้อมูล

2). การทำ Feature Extraction และ Feature Transformations

3). การสร้างโมเดลจากข้อมูล

4). การปรับปรุงประสิทธิภาพของโมเดล

6.1 การประมวลผลข้อความเบื้องต้น

สำหรับข้อมูลถามตอบจากสแต็กโอเวอร์โฟลว์มีข้อได้เปรียบ ในบางครั้งทางเว็บสแต็กโอเวอร์โฟลว์ได้มีทีมช่วยในการตรวจสอบไวยากรณ์และมีการแก้ไขคำผิดเบื้องต้น แต่ก็ยังมีข้อมูลบางส่วนที่ไม่สามารถใช้ได้ทั้งคำหรือตัวภาษาอักษรพิเศษต่าง ๆ ซึ่งกระบวนการทั้งหมดในการทำความสะอาดข้อมูลมีรายละเอียดและขั้นตอนดังต่อไปนี้

1). ทำการแยกส่วนประกอบของโพสต์ออกเป็น 2 ส่วนตามลักษณะของข้อมูล โดยแยกเป็นเนื้อหาที่อยู่ในโพสต์ที่เป็นคำพูด คำถามทั่วไป และอีกส่วนคือ ส่วนของ code ที่เกี่ยวข้องกับคำถาม

2). ทำการแยกนับคำที่เป็นคำสั่งเฉพาะของภาษาโปรแกรมและระบบระบบปฏิบัติการได้แก่ SQL, Linux และ Windows เช่น select dir mv เป็นต้น แต่การนับจะทำการนับเฉพาะส่วนที่เป็น Tags <Code>

3). ทำการลบ Tags ต่าง ๆ ออกไป เช่น <a>, <code> เป็นต้น

4). ทำการปรับคำรูปสั้น ให้อยู่ในรูปแบบเดียวกัน เช่น "can't" เป็น "cannot"

5). ตัดคำพิเศษที่พบแต่ไม่ส่งผลออกไป คือ ชื่อของผลิตภัณฑ์ เช่น mysql, oracle เป็นต้น

6). ลบพื้นที่ว่าง (Space) และอักขระพิเศษ

7). ลบค่าตัวเลขต่าง ๆ ตัวอักษรตัวเลข

8). ลบเครื่องหมายวรรคตอน (Punctuation) ตัวอย่างเช่น “?”, “@” โดยจะทำในส่วนที่เป็นเนื้อหาของโพสต์เท่านั้น ไม่ทำที่ส่วนของ code

9). แบ่งคำด้วยการทำ Tokenization

10). ทำการปรับตัวอักษรให้อยู่ในรูปแบบของตัวเล็กทั้งหมด

11). ทำการลบคำ Stop Word และปรับคำโดยใช้ Lemmatization อ้างอิงฐานข้อมูล Stop Word จาก Link http://xpo6.com/download-stop-word-list/ และการเปลี่ยนรูปคำจากไลบรารีของ NLTK [23]

6.2 การทำ Feature Extraction และ Feature Transformations

ในส่วนของกระบวนการนี้ เป็นส่วนสำคัญสำหรับการทำงานด้านการแปลงภาษาธรรมชาติให้คอมพิวเตอร์เข้าใจและสามารถตีความภาษาของมนุษย์ได้ ซึ่งวิธีที่งานวิจัยนี้ทำคือการแปลงข้อความเป็นรูปแบบเวกเตอร์คุณลักษณะข้อความ เพื่อทำให้คอมพิวเตอร์สามารถประมวลผลได้ ข้อมูลที่ใช้วิจัย มีการแยกส่วนการทำ Feature Extraction โดยการแบ่งส่วนเป็นดังนี้

- Feature Extraction ส่วนของข้อมูลเนื้อหาโพสต์ที่เป็นคำถามซึ่งไม่มีส่วนของ code
- Feature Extraction ส่วนของข้อมูล code ที่เกี่ยวข้องกับคำถาม

Feature Extraction ข้อมูลเนื้อหาโพสต์ที่ไม่มีส่วนของ code

ในส่วนนี้มีการแปลงและประมวลผลคุณลักษณะที่เกี่ยวกับข้อความ (Textual Feature) ในรูปแบบรายละเอียดจำเพาะ (Feature Engineering) [2],[24] โดยใช้วิธีการดังนี้

1). Term Frequency-Inverse Document Frequency (TF-IDF) คือ การคำนวณความถี่และความสำคัญของคำในเอกสารและปรับใช้ในรูปแบบเวกเตอร์ของข้อมูล

2). Bag of Words (BoW) คือการนับความถี่ของคำที่ปรากฏในเอกสารในรูปแบบจำนวนเต็ม

3). Word2Vec คือการแปลงข้อมูลคำเป็นเวกเตอร์ของข้อมูล โดยใช้โมเดล Continuous Bag of Words (CBOW) สำหรับงานวิจัยนี้

4). สร้างลักษณะเฉพาะจากคำที่พบในเนื้อหาตามกลุ่มต่าง ๆ เช่น กลุ่ม Installation จะมีคำที่พบบ่อยเช่น Setup install เป็นต้น

5). สร้างลักษณะเฉพาะจากการนับคำทั่วไปที่พบในข้อความ

6). สร้างลักษณะเฉพาะจากการนับจำนวนโครงสร้างของประโยคข้อความหรือ Frequency of Part of Speech Tagging (POS) โดยสนใจในส่วนของ คำนาม กริยา

7). สร้างลักษณะเฉพาะจากการนับคำจากฐานข้อมูลของคำสั่งพิเศษที่พบ โดยแยกเป็น 2 ส่วน คือ คำสั่งของระบบปฏิบัติการ และภาษา SQL เช่น mkdir, dir, find เป็นต้น

8). สร้างลักษณะเฉพาะจากการวิเคราะห์ความรู้สึกของประโยค (Sentiment)

9). เฉพาะในส่วนของการทำงานร่วมกับ Deep Learning Convolutional Neural Networks จะสลับไปใช้งาน Word Embeddings Layer ทดแทน TF-IDF

Feature Extraction ข้อมูลส่วนของ code ในส่วนนี้มีการปรับลักษณะบางอย่างออกไป ได้แก่ Frequency of Part of Speech Tagging และ Sentiment

6.3 การพัฒนาโมเดลการเรียนรู้ของเครื่องจากข้อมูล

ในส่วนของกระบวนการนี้ลักษณะงานจะเป็นการทดลองนำข้อมูลที่เตรียมจากกระบวนการก่อนหน้ามาผ่านอัลกอริทึมการเรียนรู้ของเครื่องในวิธีต่าง ๆ แล้วหาผลที่ดีที่สุด กระบวนการนี้มีการทำงาน 2 ส่วนด้วยกัน คือ ส่วนที่ 1 การสร้างโมเดลเพื่อจำแนกปัญหาทั้ง 9 กลุ่มปัญหาที่ประกอบด้วย Development-Design, Development-Discussion, Development-Limitation, Installation-Design, Installation-Discussion, Installation-Limitation, Performance Tuning-Design, Performance Tuning-Discussion, Performance Tuning-Limitation อีกส่วนคือการสร้างโมเดล Topic Modeling หรือ Latent Dirichlet Allocation (LDA) ที่เป็นการเจาะลึกว่าในแต่ละปัญหาพูดถึงหัวข้อใดบ้าง

ในส่วนของการจำแนกปัญหา 9 กลุ่ม เริ่มจากนำข้อมูลข้อความที่ถูกแปลงเป็นเวกเตอร์ข้อความพร้อมทั้งป้ายข้อมูลเข้าสู่อัลกอริทึมการเรียนรู้ของเครื่องซึ่งในงานวิจัยนี้ได้เลือกอัลกอริทึมการเรียนรู้ของเครื่องในลักษณะแบบมีผู้ฝึกสอนชนิด Multi-Class Classifier ซึ่งมีกลุ่มอัลกอริทึมที่ใช้ทดลองได้แก่ Decision Trees, Naive Bayes, Ensemble คือ Random Forest, Linear Models คือ Logistic Regression และ Deep Learning แบบ Convolutional Neural Network (CNN) [25]

ขั้นตอนและวิธีการ การสร้างโมเดลการจำแนกปัญหา 9 กลุ่ม มีลำดับขั้นตอนดังต่อไปนี้

1. ทำการแบ่งข้อมูลตามสัดส่วนจาก 15:85 ไปจนถึง 30:70 โดยแบ่งเป็นข้อมูลสำหรับเรียนรู้และข้อมูลทดสอบ

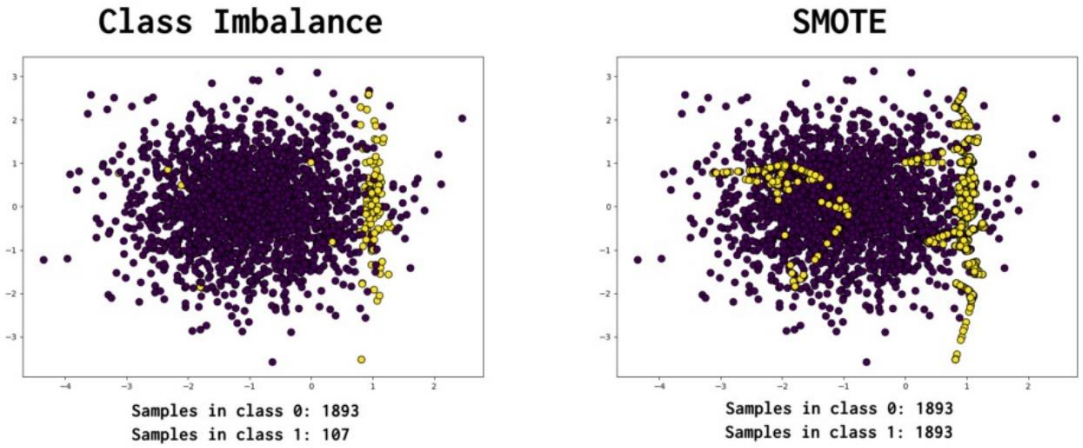
2. ทำการนำข้อมูลมาประมวลผลเบื้องต้น ตามด้วยการทำ Feature Extraction

3. นำข้อมูลเข้าอัลกอริทึมการเรียนรู้ของเครื่อง

6.4 เทคนิคที่ใช้แก้ปัญหาที่พบในการพัฒนาโมเดล

ในการสร้างโมเดลและการทดลองได้พบปัญหาต่าง ๆ และได้มีการใช้เทคนิคต่าง ๆ มาช่วยเรื่องดังกล่าวโดยมีรายละเอียดปัญหาและวิธีแก้ไข กับรายละเอียดเทคนิคที่ใช้ดังนี้

1. ปัญหาความไม่สมมาตรของข้อมูล ความไม่เท่าเทียมกันของข้อมูล (Asymmetric Information) หรือ ข้อมูลที่มีลักษณะสัดส่วนประเภทของข้อมูลไม่เท่ากัน (Data Imbalancing) ในแต่ละกลุ่มปัญหา กล่าวคือในลักษณะของปัญหารูปแบบนี้ เกิดจากเมื่อทำการรวบรวมข้อมูลและแบ่งข้อมูลทดสอบพบว่า ข้อมูลปัญหาแต่ละกลุ่มมีจำนวนที่แตกต่างกันเป็นจำนวนมาก ซึ่งมีความเป็นไปได้ที่เกิดจากลักษณะข้อมูลตั้งต้นที่มาจากระบบถามตอบที่เน้นกลุ่มปัญหาบางกลุ่มมากกว่า ตัวอย่างเช่น กลุ่มปัญหาที่เกี่ยวกับ Development จะมีมากกว่า Installation เป็นต้น
จากปัญหาดังกล่าวที่พบผู้วิจัยได้ทำการทดลองใน 2 รูปแบบคือ (1) ทำการใช้ข้อมูลที่ไม่สมมาตรทำการทดลอง (2) ใช้วิธีจำลองข้อมูลทางสถิติแบบ SMOTE ผู้วิจัยได้เลือกใช้แบบ SMOTE over-sampling เพื่อสร้างข้อมูลปัญหาที่มีน้อยตามจำนวนปัญหาที่มากที่สุดให้มีข้อมูลเป็นจำนวนเท่ากันในแต่ละกลุ่ม การเพิ่มข้อมูลแบบ SMOTE over-sampling จะมีลักษณะตามรูปที่ 6-1



รูปที่ 6-1 ตัวอย่างการทำ SMOTE [26],[27]

สำหรับสัดส่วนของข้อมูลที่นำมาทดลอง ได้เริ่มทดลองจากสัดส่วน 30:70 โดยแบ่งเป็นข้อมูลทดสอบ 30 เปอร์เซ็นต์ (Test Set) และ 70 เปอร์เซ็นต์สำหรับข้อมูลเพื่อเรียนรู้ (Training Set) จนไปถึงจุดสุดท้ายที่สัดส่วน 15:85 สำหรับวิธีการแบ่งข้อมูลได้มีการใช้วิธีแบ่งกลุ่มแบบสุ่มตัวอย่างแบ่งชั้น (Stratified Random Sampling) โดยใช้จำนวนกลุ่มหัวข้อปัญหา 9 กลุ่ม เป็นตัวช่วยในการแบ่งแยกข้อมูล และสุ่มตามสัดส่วน ซึ่งจำนวนที่ให้ผลที่ดีที่สุดในงานวิจัยนี้ พบว่าสัดส่วนข้อมูลแบบ 20:80 และ 15:85 มีความใกล้เคียง ดีกว่าน้อยกว่าสลับ ๆ กันไปแต่มากสุดจะเป็น 20:80 ดังนั้น

ลำดับต่อไปในตารางแสดงผลต่าง ๆ จะใช้ข้อมูลสัดส่วน 20:80 ตามที่เสนอไป ซึ่งจำนวนข้อมูลจะสรุปได้ตามตารางที่ 6-1

ตารางที่ 6-1 ตารางสรุปจำนวนก่อนและหลังทำข้อมูลแบบ SMOTE

ratio	Training Set	Class Average Training per 1 Class	Test Set
70--30	**9279**	**not use SMOTE**	**3978**
80--20	**10605**	**not use SMOTE**	**2652**
85--15	**11268**	**not use SMOTE**	**1989**
70--30 (SMOTE)	**23121**	**2569**	**3978**
80--20 (SMOTE)	**26433**	**2937**	**2652**
85--15 (SMOTE)	**28080**	**3120**	**1989**

2. ปัญหามิติข้อมูลและขนาดเวกเตอร์ข้อมูลที่มีจำนวนมาก (High Dimensional Problem)

กล่าวคือในลักษณะของปัญหานี้นั้นเกิดจากเมื่อทำการแปลงข้อมูลที่เป็นตัวอักษรมาสู่เวกเตอร์ข้อมูล จะพบว่าการนำคำมาใช้กำหนดเวกเตอร์มีความหลากหลายสูงมากหรือเรียกได้ว่ามีมิติข้อมูลที่มากจนเกินไป อันเนื่องจากมีคำหลากหลายที่พบ ในงานวิจัยสามารถแตกเวกเตอร์ที่มีมิติได้มากถึงหลักแสนจนไปถึงหลักล้าน ดังนั้น ผู้วิจัยจึงได้ใช้วิธีการลดมิติของข้อมูลลงเพื่อให้สนใจในกลุ่มที่มีความสำคัญสูงสุดเท่านั้น ปัญหานี้ได้ใช้อัลกอริทึม 3 ชุด ในการทดลองประกอบด้วย TruncatedSVD, Chi-Square, MinMaxScaler

TruncatedSVD ทำหน้าที่ในการลดรูปให้มีจำนวนที่ดีที่สุด โดยในงานวิจัยนี้ได้ทดลอง 10-1,000 มิติ โดยผลที่ดีที่สุดใช้ค่าที่ 200 มิติ

Chi-Square ทำหน้าที่ปรับการลดมิติในลักษณะเดียวกับ TruncatedSVD แต่ให้ผลค่าตัวเลขที่ต่างกัน โดย Chi-Square เป็นค่าบวกเสมอ

MinMaxScaler ทำหน้าที่ปรับค่าของข้อมูลให้อยู่ในช่วง 0 และ 1 เท่านั้น เป็นวิธีในการ Normalization ค่าของข้อมูลจากการลดมิติข้อมูล

3. ปัญหาของการหาค่าตัวแปรที่เหมาะสมสำหรับอัลกอริทึม การใช้งานอัลกอริทึมต่าง ๆ มีค่าตัวแปรที่สามารถปรับได้หลากหลาย ดังนั้นการทดลองได้มีการแก้ปัญหาโดยการใช้ฟังก์ชัน GridSearchCV

GridSearchCV เป็นฟังก์ชันที่ช่วยในการทำ Hyperparameter tuning โดยทำการใส่ชุดของตัวแปรที่ต้องการ ฟังก์ชันดังกล่าวจะทำหน้าที่ในการหาค่าที่เหมาะสมกับโมเดลมากที่สุด

6.5 การประเมินผลประสิทธิภาพโมเดลการจำแนกปัญหา

สำหรับกระบวนการประเมินผลในงานวิจัยนี้ได้ใช้ตัววัดผลจากการคำนวณค่า Precision, Recall, F1-Score, Accuracy ในการเปรียบเทียบ และได้มีการใช้วิธี Stratified Sampling กับ 5-Fold Cross Validation ในการประเมิน

ผลลัพธ์จากการประเมินประสิทธิภาพโมเดลการจำแนกปัญหา 9 กลุ่ม

ในส่วนนี้ ได้มีขั้นตอนการจัดทำ การทดลอง โดยไล่ลำดับและเพิ่มเทคนิคต่าง ๆ เพื่อประสิทธิภาพสูงสุด โดยมีลำดับการทำงานดังนี้

1). ทำการนำข้อมูลที่ผ่านกระบวนการแปลงค่า ปรับปรุงค่าตามตารางที่ 6-2 เพื่อให้คอมพิวเตอร์เรียนรู้เข้าสู่อัลกอริทึม โดยไล่ลำดับอัลกอริทึมในกลุ่มต่าง ๆ ภายใต้ไลบรารี Scikit-learn ประกอบด้วย กลุ่ม Support Vector Machines LinearSVC กลุ่ม Naive Bayes กลุ่ม Decision Trees กลุ่ม Ensemble [10] Random Forest, VotingClassifier และ XGBoost [12] ภายใต้ไลบรารี tensorflow keras [25] จะใช้อัลกอริทึมในกลุ่มการทำงานแบบ Deep Learning โดยใช้แบบConvolutional Neural Network (CNN)

ตารางที่ 6-2 สรุปเทคนิคที่ใช้งาน

Technique	Detail
Feature Extraction	Texts Area (TF-IDF, BoW, DOC2VEC, Word Embeddings Layer) [27]
Feature Selection	Dimensional Reduction Techniques(SVD, MinMaxScaler, Chi-**S**quare)
Feature Transformations	SQL Count, Word Or Token Count, Command_Count (Linux Command or Windows Command), Class Word Count (9 classes), Sentiment, POS
Data lmbalancing	SMOTE (Oversampling)
Other	GridSearchCV, Cross Validation (K = 5)

2). ทำการนำข้อมูลส่วนที่เป็นข้อมูลเพื่อให้คอมพิวเตอร์เรียนรู้เข้าสู่อัลกอริทึม โดยไล่ลำดับอัลกอริทึมในชุดที่ทำงานในรอบแรกที่มีผลที่น่าพอใจสูงสุดไล่ลงมา เข้าสู่การทำงานแบบ K-Fold Cross Validation โดยใช้ค่า k เป็น 5 เพื่อนำผลที่ได้มาหาโมเดลที่ดีที่สุด โดยทดสอบข้อมูล 2 ระดับ คือ

(1) ระดับข้อมูลที่มีคำที่สามารถบอกกลุ่มปัญหาได้ทันทีที่จำนวนโดยประมาณ 5,000 รายการ ซึ่งมีผลแยกตามตารางที่ 6-3 ถึง 6-5

(2) ระดับข้อมูลทั้งหมดที่จำนวนโดยประมาณ 13,000 รายการตามตารางที่ 6-6 ถึงตารางที่ 6-8 ด้านล่าง โดยใช้ Feature ตามตารางด้านบน เหมือนกัน ทุกวิธีโดยในตารางที่ 6-3 ถึง 6-8 จะเป็นการใช้ TF-IDF หรือ Word2Vec ยกเว้น CNN ทำการใช้ Word Embeddings Layer ทดแทน ทั้งนี้สำหรับตารางด้านล่าง Voting (Mix) [28] คือรวมอัลกอริทึมการเรียนรู้ของเครื่องโดยประกอบด้วยอัลกอริทึม Random Forest, Decision Tree, Extra Tree, SGD

โดยข้อมูลในตารางที่ 6-3 ถึง 6-8 จะมีการเน้นตัวหนาสำหรับค่าที่เป็นผลที่ดีที่สุดในตารางนั้น ๆ โดยดูผลจากค่า F1 เป็นลำดับแรก ในกรณีที่มีค่า F1 ที่ใกล้เคียงกัน จะพิจารณาค่าอื่น ๆ ตามมาโดยดูจากความแตกต่างกันของค่าอื่น ๆ ประกอบด้วย

ตารางที่ 6-3 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Installation
ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ

Configuration	Installation			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	0.63	0.49	0.55	0.61
Voting (Random Forest 3 set) (TF-IDF)	0.58	0.49	0.53	0.64
Voting (Mix) (TF-IDF)	0.57	0.49	0.53	0.51
CNN	0.59	0.18	0.27	0.32
XGBoost (TF-IDF)	**0.63**	**0.64**	**0.63**	**0.62**
XGBoost (Word2Vec)	0.58	0.38	0.46	0.58
Limitation				
Random Forest (TF-IDF)	**0.57**	**0.81**	**0.67**	**0.55**
Voting (Random Forest 3 set) (TF-IDF)	0.56	0.75	0.64	0.6
Voting (Mix) (TF-IDF)	0.59	0.73	0.65	0.62
CNN	0.67	0.08	0.14	0.2
XGBoost (TF-IDF)	0.47	0.35	0.40	0.47
XGBoost (Word2Vec)	0.47	0.29	0.36	0.47
Discussion				
Random Forest (TF-IDF)	0.71	0.35	0.46	0.59
Voting (Random Forest 3 set) (TF-IDF)	0.62	0.3	0.4	0.57
Voting (Mix) (TF-IDF)	0.6	0.4	0.48	0.7
CNN	**0.47**	**0.95**	**0.63**	**0.62**
XGBoost (TF-IDF)	0.61	0.66	0.63	0.60
XGBoost (Word2Vec)	0.57	0.77	0.65	0.56

ตารางที่ 6-4 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Development
ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ

Configuration	Development			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	0.73	0.74	0.73	0.69
Voting (Random Forest 3 set) (TF-IDF)	0.69	0.74	0.73	0.69
Voting (Mix) (TF-IDF)	0.67	0.75	0.71	0.69
CNN	0.67	0.57	0.62	0.61
XGBoost (TF-IDF)	**0.72**	**0.80**	**0.75**	**0.71**
XGBoost (Word2Vec)	0.67	0.85	0.75	0.66
Limitation				
Random Forest (TF-IDF)	0.85	0.84	0.84	0.86
Voting (Random Forest 3 set) (TF-IDF)	0.55	0.74	0.63	0.57
Voting (Mix) (TF-IDF)	0.52	0.61	0.56	0.48
CNN	0.84	0.83	0.84	0.82
XGBoost (TF-IDF)	**0.86**	**0.84**	**0.85**	**0.86**
XGBoost (Word2Vec)	0.83	0.83	0.83	0.82
Discussion				
Random Forest (TF-IDF)	0.54	0.76	0.63	0.51
Voting (Random Forest 3 set) (TF-IDF)	**0.85**	**0.85**	**0.85**	**0.83**
Voting (Mix) (TF-IDF)	0.81	0.84	0.82	0.83
CNN	0.44	0.8	0.56	0.52
XGBoost (TF-IDF)	0.58	0.69	0.63	0.58
XGBoost (Word2Vec)	0.57	0.54	0.56	0.57

ตารางที่ 6-5 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Performance Tuning ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 5,000 รายการ

Configuration	Performance Tuning			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	0.66	0.55	0.6	0.7
Voting (Random Forest 3 set) (TF-IDF)	0.55	0.54	0.57	0.59
Voting (Mix) (TF-IDF)	0.56	0.63	0.59	0.51
CNN	0.55	0.32	0.4	0.41
XGBoost (TF-IDF)	**0.69**	**0.60**	**0.64**	**0.68**
XGBoost (Word2Vec)	0.48	0.54	0.51	0.47
Limitation				
Random Forest (TF-IDF)	0.51	0.73	0.6	0.51
Voting (Random Forest 3 set) (TF-IDF)	0.82	0.48	0.61	0.81
Voting (Mix) (TF-IDF)	0.83	0.5	0.62	0.83
CNN	0.95	0.32	0.48	0.42
XGBoost (TF-IDF)	**0.78**	**0.54**	**0.64**	**0.78**
XGBoost (Word2Vec)	0.73	0.43	0.54	0.72
Discussion	-	-	-	-
Random Forest (TF-IDF)	**0.87**	**0.48**	**0.62**	**0.86**
Voting (Random Forest 3 set) (TF-IDF)	0.5	0.62	0.55	0.49
Voting (Mix) (TF-IDF)	0.6	0.4	0.48	0.6
CNN	0.39	0.78	0.52	0.55
XGBoost (TF-IDF)	0.56	0.72	0.63	0.55
XGBoost (Word2Vec)	0.47	0.45	0.46	0.47

ที่ระดับข้อมูลทั้งหมด 13,000 รายการ ของการทดลองสร้างโมเดล ทุกตัวแปรใช้ค่าเดิมโดยปรับปรุงแค่จำนวนที่ใช้ในการเรียนรู้และทดสอบเท่านั้น โดยใช้สัดส่วนที่ 80:20 โดยได้ผลตามตารางที่ 6-6 ถึง 6-8

ตารางที่ 6-6 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Installation ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 13,000 รายการ

Configuration	Installation			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	**0.42**	**0.42**	**0.42**	**0.41**
Voting (Random Forest 3 set) (TF-IDF)	0.45	0.28	0.34	0.45
XGBoost(TF-IDF)	0.41	0.35	0.38	0.41
CNN	0.07	0.19	0.1	0.07
XGBoost (Word2Vec)	0.58	0.09	0.16	0.57
Limitation				
Random Forest (TF-IDF)	0.32	0.24	0.27	0.31
Voting (Random Forest 3 set) (TF-IDF)	0.34	0.19	0.25	0.34
XGBoost(TF-IDF)	**0.37**	**0.24**	**0.29**	**0.36**
CNN	0.09	0.2	0.13	0.09
XGBoost (Word2Vec)	0.01	0.01	0.01	0.1
Discussion				
Random Forest (TF-IDF)	0.38	0.61	0.47	0.38
Voting (Random Forest 3 set) (TF-IDF)	**0.43**	**0.65**	**0.52**	**0.43**
XGBoost(TF-IDF)	0.48	0.55	0.51	0.47
CNN	0.16	0.26	0.2	0.25
XGBoost (Word2Vec)	0.54	0.31	0.39	0.54

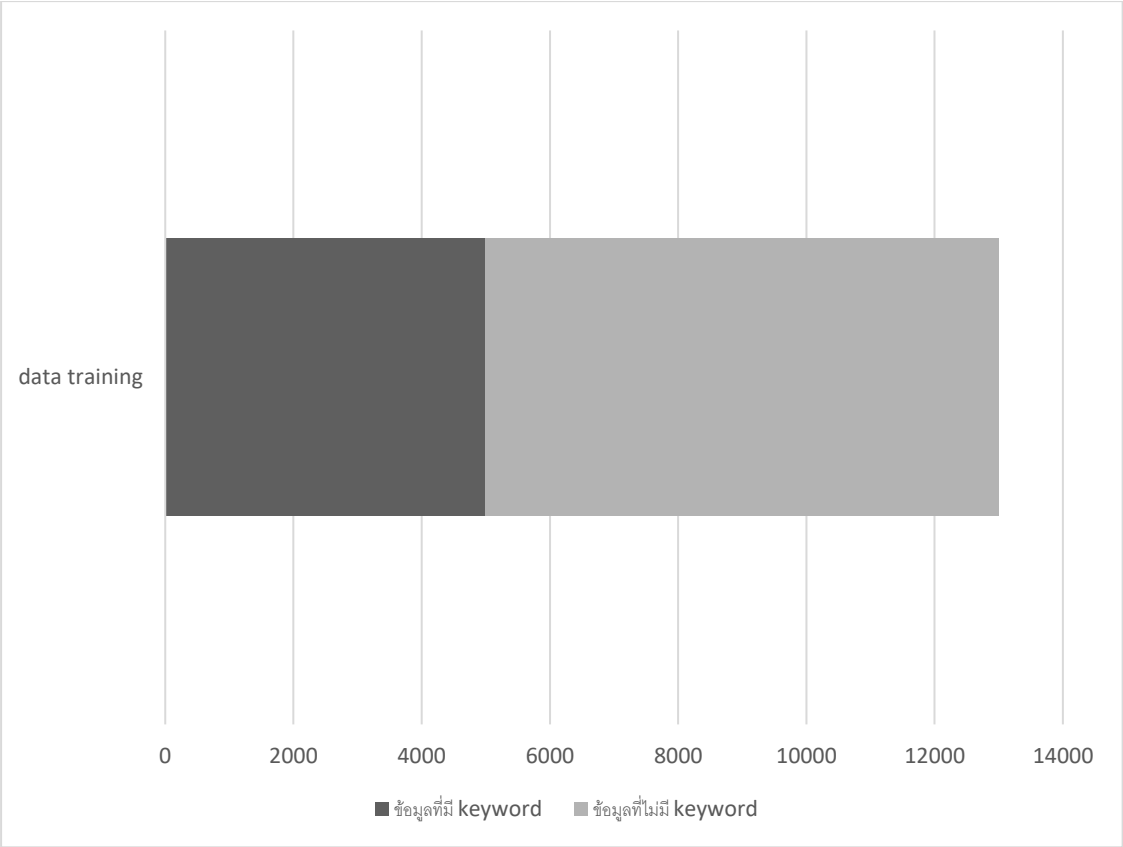
ตารางที่ 6-7 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Development
ในแต่ละปัญหาย่อย ระดับข้อมูลที่ 13,000 รายการ

Configuration	Development			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	0.52	0.57	0.55	0.52
Voting (Random Forest 3 set) (TF-IDF)	0.51	0.57	0.54	0.51
XGBoost(TF-IDF)	**0.55**	**0.6**	**0.57**	**0.54**
CNN	0.4	0.13	0.2	0.39
XGBoost (Word2Vec)	0.46	0.58	0.52	0.46
Limitation				
Random Forest (TF-IDF)	0.75	0.57	0.65	0.74
Voting (Random Forest 3 set) (TF-IDF)	0.75	0.57	0.65	0.74
XGBoost(TF-IDF)	0.73	0.62	0.67	0.72
CNN	0.36	0.54	0.43	0.36
XGBoost (Word2Vec)	**0.83**	**0.59**	**0.69**	**0.82**
Discussion				
Random Forest (TF-IDF)	0.57	0.56	0.56	0.56
Voting (Random Forest 3 set) (TF-IDF)	0.53	0.61	0.57	0.52
XGBoost(TF-IDF)	**0.55**	**0.64**	**0.59**	**0.55**
CNN	0.43	0.49	0.46	0.42
XGBoost (Word2Vec)	0.45	0.78	0.57	0.44

ตารางที่ 6-8 ตารางประสิทธิภาพของโมเดลการจำแนกปัญหา Performance Tuning ในแต่ละปัญหาย่อยระดับ ข้อมูลที่ 13,000 รายการ

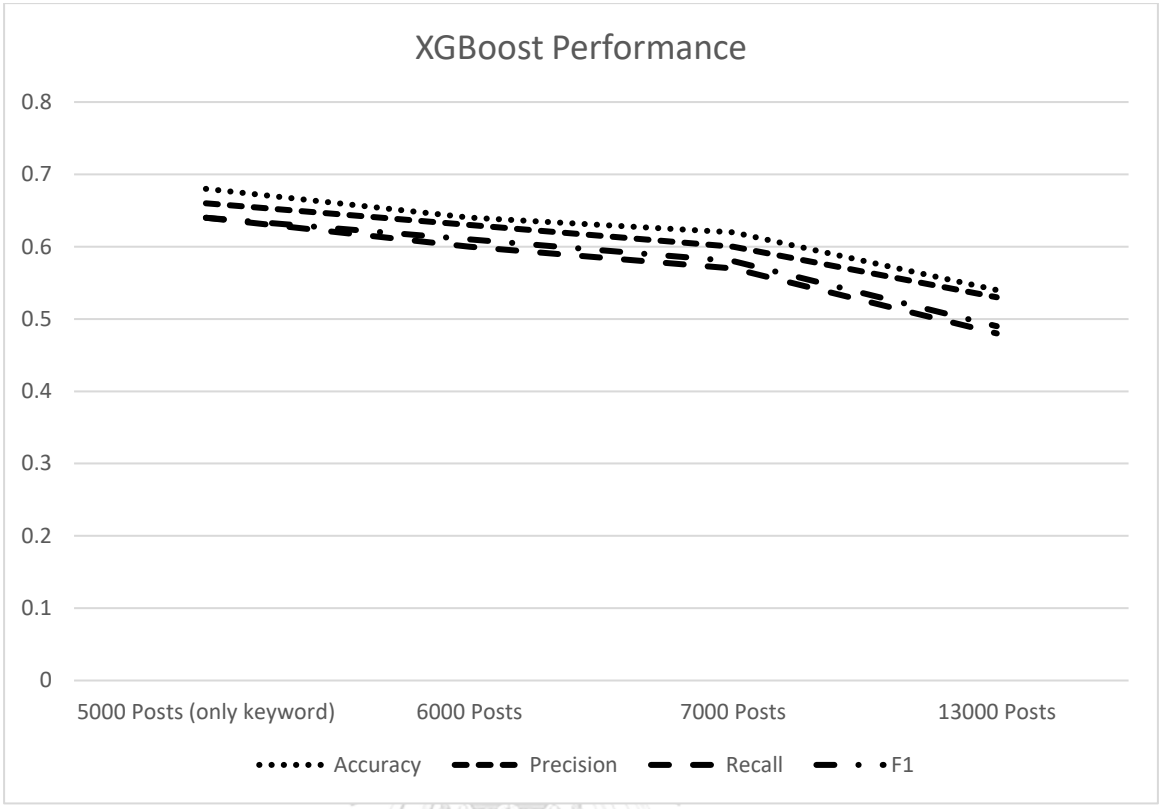
Configuration	Performance Tuning			
	Precision	Recall	F1	Accuracy
Design				
Random Forest (TF-IDF)	**0.51**	**0.52**	**0.52**	**0.5**
Voting (Random Forest 3 set) (TF-IDF)	0.55	0.44	0.49	0.54
XGBoost(TF-IDF)	0.55	0.41	0.47	0.53
CNN	0.26	0.52	0.34	0.25
XGBoost (Word2Vec)	0.44	0.29	0.35	0.44
Limitation				
Random Forest (TF-IDF)	**0.75**	**0.52**	**0.61**	**0.75**
Voting (Random Forest 3 set) (TF-IDF)	0.67	0.58	0.62	0.66
XGBoost(TF-IDF)	0.7	0.51	0.59	0.7
CNN	0.1	0.1	0.1	0.1
XGBoost (Word2Vec)	0.93	0.32	0.47	0.92
Discussion				
Random Forest (TF-IDF)	0.41	0.37	0.39	0.41
Voting (Random Forest 3 set) (TF-IDF)	**0.51**	**0.34**	**0.54**	**0.51**
XGBoost(TF-IDF)	0.46	0.41	0.43	0.45
CNN	0.45	0.04	0.07	0.45
XGBoost (Word2Vec)	0.45	0.08	0.14	0.45

จากกระบวนการข้างต้นพบข้อสังเกตเกี่ยวกับจำนวนข้อมูลที่น่าสนใจคือ เมื่อจำนวนมีมากขึ้นและข้อมูลที่นำมาใช้เรียนรู้มีความหลายหลายมากยิ่งขึ้น พบว่าความสามารถในการเรียนรู้ของเครื่องประสิทธิภาพของโมเดลลดลงอย่างเห็นได้ชัด โดยมีตัวแปรที่น่าสนใจในข้อมูลคือ ข้อมูลส่วนแรกประมาณ 5,000 โพสต์ เป็นข้อมูลที่มีลักษณะพิเศษ คือมี keyword จำพวกคำที่สามารถบอกแยกกลุ่มปัญหาต่าง ๆ เช่น “Install”, “setup” เป็นต้น ส่วนที่เหลือเป็นโพสต์ที่ไม่มีคำ keyword ที่แยกเรื่องดังกล่าวชัดเจน โดยแบ่งสัดส่วนตามรูปที่ 6-2



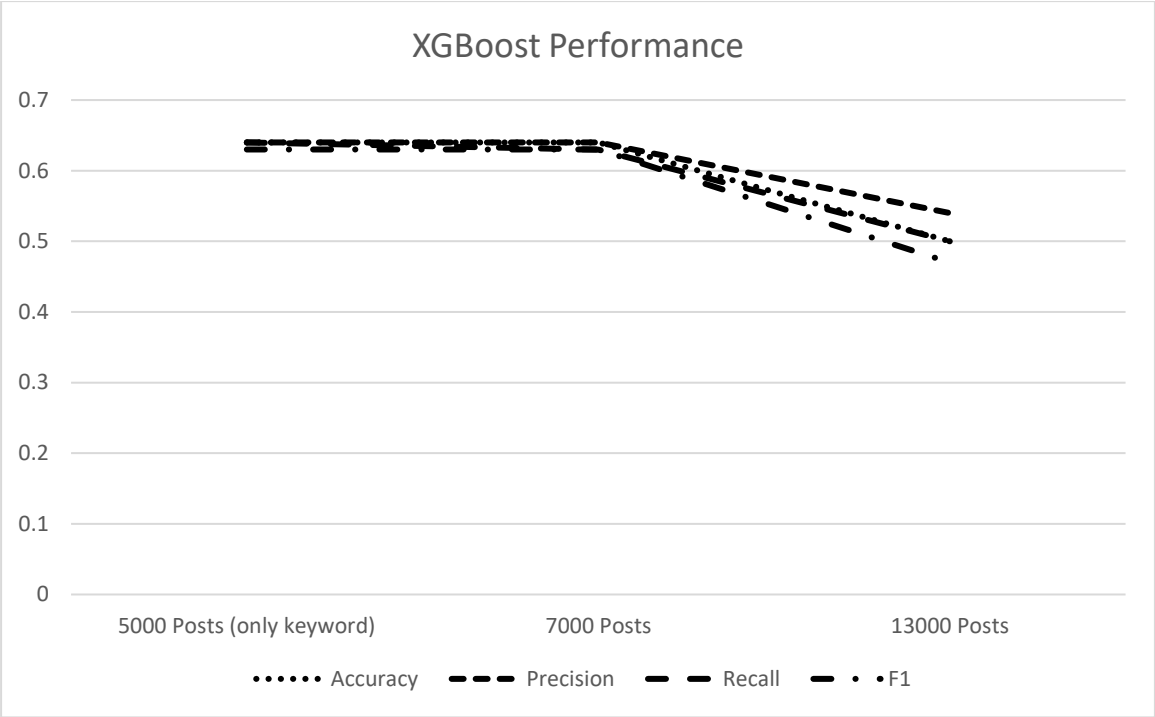
รูปที่ 6-2 สัดส่วนของข้อมูลที่มีคำศัพท์ที่สามารถแยกกลุ่มได้ทันทีกับคำทั่วไป

ในเรื่องดังกล่าวพบว่ามีปัญหาต่อประสิทธิภาพของโมเดลพอสมควร โดยลักษณะคือการเรียนรู้ข้อมูลเพื่อสร้างโมเดลในข้อมูลชุดแรก 5,000 โพสต์ จะได้ค่าที่ดีที่สุด โมเดลดีที่สุด จากนั้นเมื่อเริ่มเติมข้อมูลเพิ่มเข้าไป โดยเริ่มจาก 5,000 โพสต์ ไปเป็น 6,000 โพสต์ พบว่าค่าต่าง ๆ ที่ใช้วัดผลประสิทธิภาพมีค่าลดลงโดยมีการทดลองสลับข้อมูลหลาย ๆ ชุด ชุดละ 1,000 โพสต์เติมไปในการทำงาน ซึ่งผลที่ได้มีแนวโน้มที่ค่าประสิทธิภาพลดลง ในจุดนี้ผู้วิจัยคิดว่าเกิดจากการที่ลักษณะของข้อมูลมีความหลากหลายมากเกินไปในรูปแบบความถี่ของคำที่พบ ทำให้เมื่อยิ่งเพิ่มคำที่หลากหลายมากค่าประสิทธิภาพก็ยิ่งลดมาก บวกกับปัญหาเดิมที่ข้อมูลแต่ละกลุ่มปัญหามีความแตกต่างกันในเรื่องของจำนวนในระดับที่มีนัยสำคัญ ผลการทดสอบเรื่องดังกล่าวได้นำมาแสดงเป็นกราฟดังตัวอย่างของวิธีการ XGBoost ดังรูปกราฟที่ 6-3 ด้านล่าง โดยจะเห็นว่ากราฟ มีแนวโน้มลดลงยิ่งจำนวนข้อมูลมาก ประสิทธิภาพที่ได้ยิ่งน้อยลง



รูปที่ 6-3 รูปกราฟแสดงตัวอย่างแนวโน้มของผลการทดลองต่อจำนวนข้อมูลที่ใช้ทดลองด้วย TF-IDF

ข้อสังเกตเพิ่มเติมในส่วนของการทดลองด้วยการเปลี่ยนจาก TF-IDF ไปใช้ Word2Vec พบว่าค่าเฉลี่ยโดยรวมไม่ต่างกันมาก แต่มีจุดที่น่าสนใจในเรื่องของจำนวนข้อมูลพบว่าในช่วงข้อมูลที่ 5,000 ถึง 7,000 รายการ ค่าประสิทธิภาพที่วัดได้พบว่ามีความใกล้เคียงกันสูงมากซึ่งจุดนี้ต่างกับ TF-IDF มาก เพราะ TF-IDF จะทยอยลดลงตามจำนวนข้อมูลที่เพิ่มขึ้นมา ซึ่งทำให้ขนาดมิติข้อมูลมีความใหญ่ขึ้น แต่เมื่อใส่ข้อมูลเพิ่มขึ้นจาก 7,000 จนถึง 13,000 รายการ พบว่าค่าประสิทธิภาพที่ได้มีแนวโน้มต่ำลง จุดนี้อาจจะมองได้ว่าข้อมูลที่เพิ่มเข้ามามากขึ้นทำให้เกิดความหลากหลายของคำมากขึ้น ก่อให้เกิดเวกเตอร์ข้อมูลที่ขนาดใหญ่มากขึ้นส่งผลต่อค่าประสิทธิภาพมากขึ้นดังรูปที่ 6-4



รูปที่ 6-4 รูปกราฟแสดงตัวอย่างแนวโน้มของผลการทดลองต่อจำนวนข้อมูลที่ใช้ทดลองด้วย Word2Vec

ตารางที่ 6-9 แสดงประสิทธิภาพโดยรวมของโมเดล Multiclass Classification เมื่อใช้ในการจำแนกคำถามตามกลุ่มปัญหา-ปัญหาย่อยทั้ง 9 กลุ่ม โดยใช้ข้อมูล 5,000 โพสต์ เป็นข้อมูลในการเรียนรู้ ซึ่งจะได้ว่าโมเดล XGBoost (TF-IDF) เป็นโมเดลที่มีประสิทธิภาพโดยรวมในการจำแนก 9 กลุ่มปัญหาได้ดีที่สุด และจะถูกนำไปใช้ในการพัฒนาเครื่องมือต่อไป

ตารางที่ 6-9 ตารางแสดงประสิทธิภาพโดยรวมของโมเดล Multiclass Classification โดยใช้ข้อมูล 5,000 โพสต์

Configuration	Accuracy	Precision	Recall	F1
XGBoost (TF-IDF)	**0.68**	**0.65**	**0.64**	**0.64**
Random Forest (TF-IDF)	0.65	0.66	0.63	0.61
Voting (Mix) (TF-IDF)	0.67	0.66	0.63	0.63
Extra Trees (TF-IDF)	0.64	0.62	0.61	0.60
XGBoost (Word2Vec)	0.64	0.60	0.57	0.57

6.6 การสรุปผลการประเมินผลประสิทธิภาพโมเดลการจำแนกปัญหาและแนวทางการปรับปรุงในอนาคต

จากตารางผลประสิทธิภาพของโมเดลและกราฟแนวโน้มจากการทดลองพบปัญหาในเรื่องประสิทธิภาพของโมเดล ซึ่งจากตารางที่ 6-9 พบว่าโมเดลที่ดีที่สุดมีค่า F1 เพียงแค่ 64% เท่านั้น ซึ่งจุดนี้ทำให้ผู้วิจัยสามารถสรุปแนวทางสาเหตุของประสิทธิภาพที่ยังมีค่าไม่สูงมากนัก โดยปัญหาสรุปเป็นข้อ ๆ ได้ดังนี้

1). ลักษณะของข้อมูลกับป้ายกำกับการแยกประเภทของข้อมูล

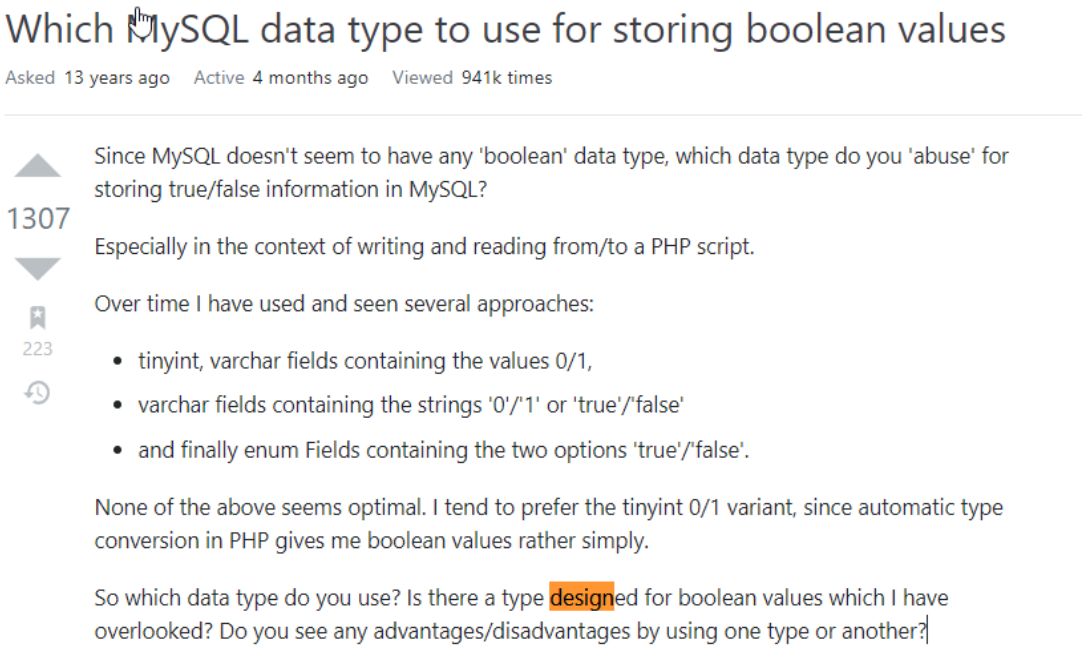
ลักษณะของข้อมูลจากการสำรวจและการทดลองที่ผ่านมาพบว่า ข้อมูลคำถาม ในกลุ่มเทคโนโลยีฐานข้อมูลมักมีโอกาสพบรูปแบบของประโยคที่มีความคล้ายคลึงกันสูง และมักจะเจอกลุ่มคำที่ซ้ำ ๆ ในหลายส่วน ดังตัวอย่างหัวข้อโพสต์เรื่องของการติดตั้งโปรแกรม เช่น “How to install MySQL in Fedora2 0 ?”, ”How to install node-sqlserver”, “How to install mongoDB on windows?” เป็นต้น โดยทั้ง 3 ประโยคได้ดึงข้อมูลมาจาก 3 แท็ก คือ “mysql”, “sqlserver” และ ”mongodb” ซึ่งจะพบว่ามีโอกาสที่จะพบคำหรือประโยคในลักษณะดังกล่าวอีก ในกรณีนี้อ่านจากหัวข้อสามารถสรุปขั้นต้นได้ว่าเป็นกลุ่มปัญหาประเภท Installation แต่เมื่อพิจารณาเนื้อหาด้านในจะพบว่าเนื้อหาสามารถจำแนกแยกได้อีกทั้ง 3 กลุ่มย่อยลงไปทั้งกลุ่ม Design, Discussion และ Limitation ทำให้มีโอกาสที่จะแยกความแตกต่างของเนื้อหาได้ยากขึ้น และมีคำซ้ำ ๆ ได้ในทั้ง 3 กลุ่มย่อย จุดนี้ส่งผลให้การทำโมเดลมีประสิทธิภาพลดลง เพราะ คำที่โมเดลจะใช้ในการจำแนกประเภทอาจจะมีความสับสนมากขึ้น

ข้อมูลอีกลักษณะที่พบได้คือภาษา SQL ตามแต่ละระบบปฏิบัติการ และรายละเอียดคำสั่งการทำงานต่าง ๆ หรือ Log output สำหรับข้อมูลดังกล่าวจะมีแนวโน้มของปัญหาที่เกิดจากการที่โพสต์ส่วนมากจะมีคำพวกนี้อยู่เป็นจำนวนมากและมีการใช้ซ้ำกัน แต่การซ้ำกันกลับมีความหมายของการซ้ำที่แตกต่างกัน เช่น คำสั่ง “sudo apt-get install- mysql-community-server” ซึ่งเป็นประโยคเดียวอาจจะสามารถตีความได้ว่าเป็นปัญหาที่พบในกลุ่ม Installation-Discussion แต่เมื่อรวมกับกลุ่มประโยค อาจจะสามารถตีความได้หลายกลุ่มปัญหาจากคำสั่งดังกล่าว เช่น ในบริบทของ Development-Design อาจจะพบการใช้คำสั่งนี้ในคำถามที่เกี่ยวกับการเขียนโปรแกรมอัตโนมัติและใช้คำสั่งนี้เป็นเงื่อนไขในการตรวจสอบสถานะการติดตั้งของโปรแกรม ในขณะที่ในบริบทของ Installation-Discussion จะกล่าวถึงคำสั่งนี้เพื่อต้องการติดตั้งโปรแกรม

ดังนั้นจึงเห็นได้ว่ามีโอกาสที่ประโยคชุดเดียวกันจะสามารถมีหลายกลุ่มได้เช่นกัน จึงเป็นปัญหาของข้อมูลที่อาจจะก่อให้เกิดความสับสนขึ้นมาได้

2). ป้ายข้อมูลทำให้เกิดการตีความได้หลายลักษณะ

ป้ายข้อมูลที่งานวิจัยใช้มี 9 กลุ่มปัญหา แต่บางกลุ่มอาจจะมีความหมายกว้าง ทำให้สามารถตีความได้หลายลักษณะ เช่นกลุ่มที่เป็นลักษณะ Design กับ Discussion โดย 2 กลุ่มดังกล่าวหลังจากที่มีการติดป้ายข้อมูลก็พบว่ามีแนวโน้มที่ผู้ติดป้ายทั้ง 5 ท่านมีความสับสน ดังตัวอย่างข้อมูลในรูปที่ 6-5 ผู้ติดป้ายข้อมูล 2 ท่านได้ติดป้ายทั้ง Development-Design และ Development-Discussion ทำให้ต้องมาหาข้อสรุปอีกครั้ง จึงมีโอกาสเป็นอย่างมากที่โมเดลที่สร้างได้จะมีประสิทธิภาพในการจำแนกประเภทได้ไม่ดีมากนัก จากการที่ลักษณะของข้อมูลที่ถูกติดป้ายสำหรับใช้ในการเรียนรู้มีลักษณะที่ไม่แตกต่างกันอย่างชัดเจนนัก



Which MySQL data type to use for storing boolean values

Asked 13 years ago Active 4 months ago Viewed 941k times

1307

223

Since MySQL doesn't seem to have any 'boolean' data type, which data type do you 'abuse' for storing true/false information in MySQL?

Especially in the context of writing and reading from/to a PHP script.

Over time I have used and seen several approaches:

- tinyint, varchar fields containing the values 0/1,
- varchar fields containing the strings '0'/'1' or 'true'/'false'
- and finally enum Fields containing the two options 'true'/'false'.

None of the above seems optimal. I tend to prefer the tinyint 0/1 variant, since automatic type conversion in PHP gives me boolean values rather simply.

So which data type do you use? Is there a type designed for boolean values which I have overlooked? Do you see any advantages/disadvantages by using one type or another?

รูปที่ 6-5 รูปตัวอย่างโพสต์ที่สามารถตีความได้ 2 กลุ่มปัญหา

3). ขนาดข้อมูลในแต่ละกลุ่มปัญหามีความแตกต่างกัน

เป็นอีกข้อสังเกตที่เกิดขึ้นหลังจากการทำการติดป้ายข้อมูล จากการที่มี 9 ป้ายข้อมูลกลุ่มปัญหา ตามสมมติฐานคาดไว้ว่าจะมีข้อมูลที่แตกต่างกันจำนวนไม่มาก แต่ผลที่ได้กลับพบว่ามีความแตกต่างกันพอควรตัวอย่างเช่น กลุ่ม Development-Design และ กลุ่ม Development-Discussion มีจำนวนข้อมูล 1,000-2,000 โพสต์ แต่กลุ่ม Performance-Tuning Limitation และ Installation-Discussion มีจำนวนข้อมูลไม่เกิน 500-800 โพสต์ ซึ่งมีความแตกต่างกันพอควร การแก้ปัญหาขั้นต้นได้ทำการใช้ SMOTE มาช่วยในการปรับปรุงให้จำนวนข้อมูลมีความใกล้เคียงแต่ยังไม่เพียงพอเมื่อเทียบกับผลประสิทธิภาพที่ออกมา จุดนี้จึงเป็นอีกจุดที่ควรต้องปรับปรุงเพราะในหลาย ๆ ตัวอย่างปัญหาในเรื่องดังกล่าวมักส่งผลต่อประสิทธิภาพของโมเดลโดยตรง

จาก 3 สาเหตุที่กล่าวมาข้างต้นทางผู้วิจัยได้มีข้อเสนอแนะแนวทางเพิ่มเติมในอนาคตที่สามารถแก้ไขได้ โดยจากสาเหตุที่ 1 และ สาเหตุที่ 2 อาจจะจำเป็นต้องปรับโครงสร้างหรือหรือนิยามความหมายของป้ายข้อมูลใหม่ให้สามารถแยกได้ชัดเจนมากยิ่งขึ้นและต้องลดปัญหาความซ้ำซ้อนของข้อมูลที่เกิดจากกลุ่มมีความใกล้เคียงกันให้มากที่สุด ส่วนในสาเหตุที่ 3 ถ้าสามารถแก้ไขปัญหาทั้ง 2 สาเหตุก่อนหน้าได้แล้วสาเหตุดังกล่าวอาจจะบรรเทาลง แต่ถ้าในกรณีแก้ไขแล้วแต่ยังเกิดสาเหตุดังกล่าว อาจจะต้องแก้ไขด้วยการทำ SMOTE ข้อมูลก่อนในลำดับแรก หรือ กรณีสามารถหาข้อมูลมาเพิ่มให้มีจำนวนที่ใกล้เคียงกันก็อาจจะแก้ไขปัญหาดังกล่าวได้เช่นกัน แต่ในกรณีที่ไม่สามารถปรับปรุงแก้ไข 2 สาเหตุแรกได้ไม่ว่าเหตุใดก็ตาม อาจจะต้องทำการปรับแนวทางการจำแนกหมวดหมู่ใหม่ (Classification) โดยจากเดิมได้ใช้รูปแบบ Multi-Class Classification (การจำแนกประเภทหลายคลาส) ที่ 1 โมเดลจำแนกปัญหาออกมาได้หลายคลาส ให้มาเป็นในลักษณะ Multi-Label Classification (การจำแนกประเภทหลายป้ายข้อมูล) ซึ่งอาจจะให้ผลที่แตกต่างกัน เช่น โพสต์ตัวอย่าง “How to install MySQL in Fedora20?” สามารถจำแนกได้เป็น Installation-Discussion และ Installation-Limitation เป็นต้น

ในด้านของเทคโนโลยีและอัลกอริทึมที่ใช้อาจจะมีการปรับปรุงเพิ่มเติมวิธีการเข้าไป เช่น ในส่วนการทดลองของ Deep Learning ให้ปรับวิธีการจาก Convolution Neural Network (CNN) ไปเป็น Recurrent Neural Networks (RNN) หรือ Artificial Neural Network (ANN) เป็นต้น

6.7 การสร้างโมเดล Topic Modeling โดยใช้เทคนิค Latent Dirichlet Allocation (LDA)

การสร้างโมเดลหัวข้อหรือ Topic Modeling [29] โดยใช้เทคนิค LDA เป็นการค้นหาหัวข้อหรือเนื้อหาบางอย่างที่ซ่อนอยู่ในกลุ่มเอกสาร โดยอาจจะเป็นคำกลาง ๆ ที่พบจำนวนมากในกลุ่มเอกสาร ลักษณะของวิธีการดังกล่าวคือการค้นหาการเกิด หรือความน่าจะเป็นของเนื้อหา คำ กลุ่มคำในเอกสารที่พบ โดยคำดังกล่าวจะเรียกว่า Latent Variable หรือ Hidden Variable ซึ่งจะมีความหมายซ่อนอยู่ โดยอาจจะสามารถตีความได้ หรือไม่ได้ก็ได้ ขึ้นอยู่กับลักษณะสำคัญของเอกสาร

สำหรับโมเดลดังกล่าวทางผู้วิจัยเห็นช่องทางในการหาหัวข้อหรือประเด็นที่มีโอกาสซ่อนอยู่ในเนื้อหาของแต่ละโพสต์ ทำให้เข้าใจกลุ่มปัญหา หรือสาระสำคัญของปัญหานั้น ๆ ได้ดีขึ้นว่าในกลุ่มปัญหา-ปัญหาย่อยนั้น มีหัวข้อที่เป็นประเด็นปัญหาอะไรบ้าง ทางผู้วิจัยได้ทำการสร้างโมเดล Topic Modeling (LDA) โดยแยกตามกลุ่มปัญหาเป็น 9 กลุ่มปัญหาที่ประกอบด้วย Development-Design, Development-Discussion, Development-Limitation, Installation-Design, Installation-Discussion, Installation-Limitation, Performance Tuning Design, Performance Tuning-Discussion, Performance Tuning-Limitation

การสร้างโมเดลเริ่มจากการนำข้อมูลแยกตามกลุ่มต่าง ๆ 9 กลุ่มปัญหาข้างต้น ข้อมูลจะได้รับการทำความสะอาดมาแล้ว และอยู่ในรูปแบบเวกเตอร์ข้อมูลที่ผ่านการทำ tokenization ลำดับถัดมาทำการนำข้อมูลที่เตรียมไว้เข้าสู่กระบวนการสร้างคลังคำศัพท์ของกลุ่มคำ โดยใช้ไลบรารีของ genism ในการช่วยสร้าง เมื่อได้คลังคำศัพท์ นำข้อมูลมาเข้าโมเดล LDA ในไลบรารีของ genism (LdaModel) เพื่อสร้างโมเดล LDA

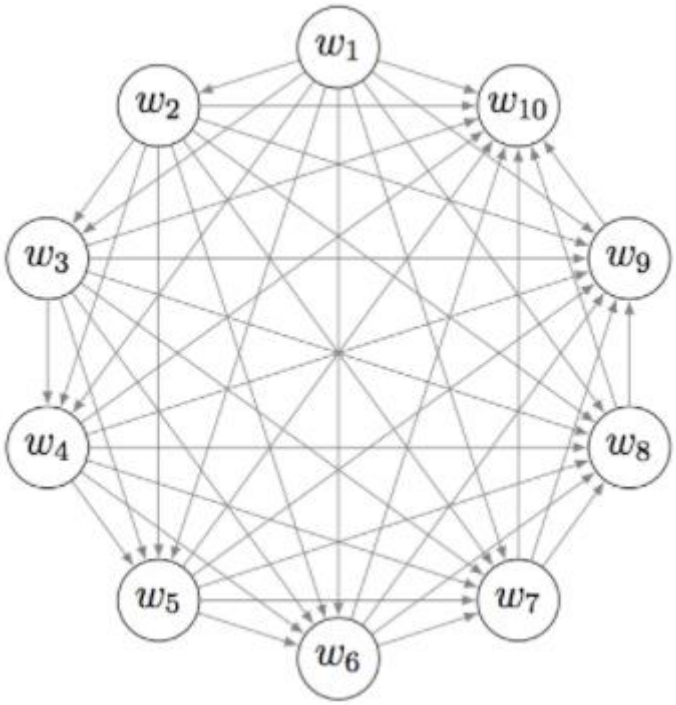
6.8 การประเมินประสิทธิภาพ Topic Modeling ซึ่งใช้เทคนิค Latent Dirichlet Allocation (LDA) [29]

การประเมินโมเดล LDA ในงานวิจัยนี้จะแยกออกเป็น 2 ส่วน ประกอบด้วยส่วนที่ 1 จะทำการประเมินความเหมาะสมของค่า K หรือ จำนวน Topics ที่เหมาะสมที่สุดในแต่ละกลุ่มปัญหา ส่วนที่ 2 คือการประเมินความแม่นยำในการนำไปใช้งาน [30]

<u>ส่วนที่ 1</u> การประเมินความเหมาะสมของจำนวน Topics สามารถวัดผลได้หลาย ๆ วิธี เช่น ค่า Coherence [31] และ ค่า Perplexity ในงานวิจัยนี้จะใช้ค่า Coherence เป็นหลัก

ค่า Coherence คือ ค่าความสอดคล้องของหัวข้อ เป็นค่าที่วัดความใกล้เคียงระหว่างคำ ว่ามีความใกล้เคียงหรือสามารถจัดอยู่ในกลุ่มเดียวกันได้หรือไม่ โดยมีลักษณะคือการคำนวณระยะห่าง

ของแต่ละคำภายในหัวข้อนั้น ๆ ตามรูปที่ 6-5 ด้านล่าง โดยในภาพจะเป็นการยกตัวอย่างว่าใน 1 กลุ่ม หรือ 1 topic จะแทนด้วย 10 คำ แต่ละคำจะมีค่ารวมตามเส้นต่าง ๆ โดยผลรวมทั้งหมดจะเป็นค่า Coherence ซึ่งหากมีค่ามาก จะแสดงว่าคำต่าง ๆ มีความเหมาะสมที่จะสื่อถึงหัวข้อเดียวกัน



รูปที่ 6-6 การหาค่า Coherence จาก Wn-Wn โดยที่ W คือ Word [31]

การหาค่า Coherence ในงานวิจัย จะใช้เครื่องมือหาค่าผ่านไลบรารีของ genism ซึ่งสามารถปรับค่า parameter ได้ 5 ค่าหลัก ๆ คือ Validation set, Alpha, Beta, Topics (K) และ word [3] โดยรายละเอียดทั้ง 5 ค่ามีดังนี้

ค่า Validation set เป็นค่าจำนวนคำศัพท์จากคลังคำศัพท์ที่ใช้ในการคำนวณหาค่า Topics (K) ที่เหมาะสมที่สุด

ค่า Alpha [32] เป็นค่าความหนาแน่นของหัวข้อ (K) ในเอกสาร ยิ่งค่ามากแสดงว่าปริมาณหัวข้อจะต้องมากตาม

ค่า Beta [33] เป็นค่าความหนาแน่นของคำที่อยู่ในหัวข้อ (K) ยิ่งค่ามากคำ ที่อยู่ภายใต้หัวข้อจะยิ่งมากตาม แต่ในงานวิจัยนี้ได้มีการกำหนดคำในหัวข้อเบื้องต้นไว้เรียบร้อยเพื่อทำให้คำมีความเป็นไปได้ในการตีความได้ง่ายขึ้น

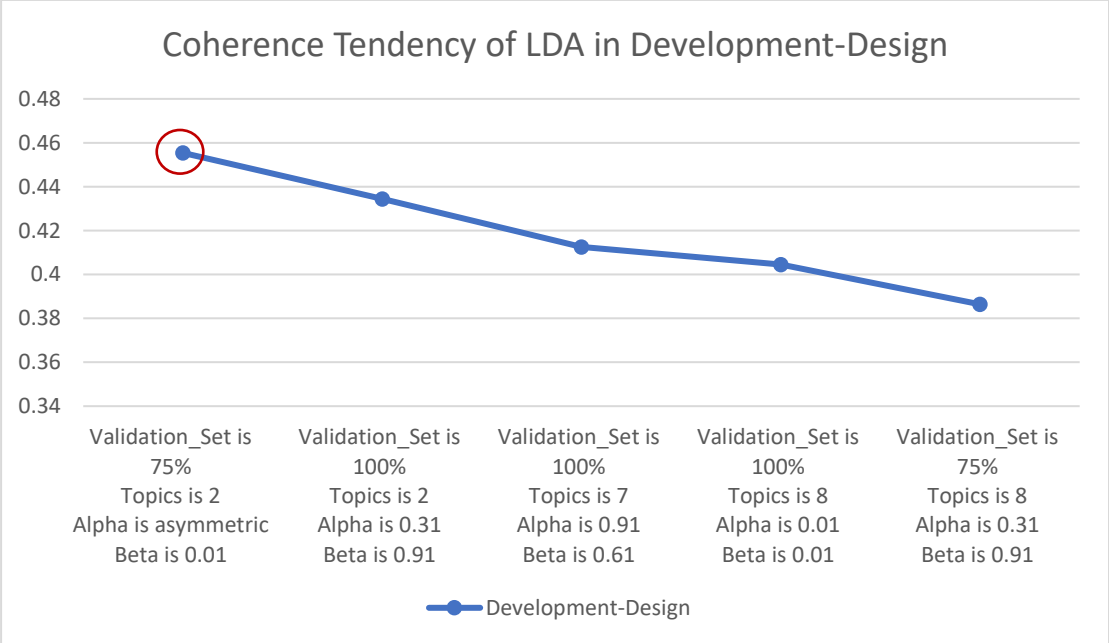
ค่า Topics (K) ค่าจำนวนหัวข้อที่เหมาะสม

ค่า Word เป็นค่าที่ใช้บอกจำนวนคำในหัวข้อที่เหมาะสม หรือที่ถูกกำหนดในการใช้งาน

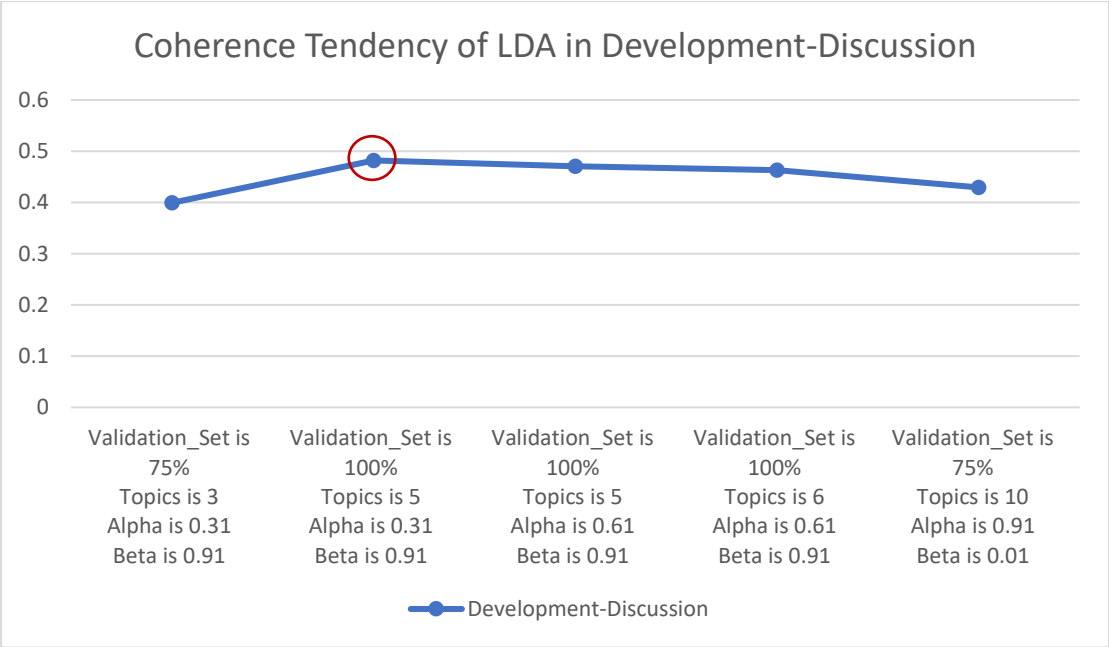
ค่าพิเศษในตัวแปร Beta และ Alpha คือ symmetric เป็นค่าที่คำนวณด้วยสูตร 1.0 หารด้วยจำนวน **Topics (K)** เช่น k = 4 ค่า symmetric จะเป็น 0.25 asymmetricจะเป็นค่าตรงกันข้ามซึ่งคำนวณด้วยสูตร 1 หารด้วย รากที่สอง (Square root) ของจำนวน Topics (K)

ในการทดลองครั้งนี้ได้ทำการปรับพารามิเตอร์ต่าง ๆ เพื่อหาค่า Coherence ที่สูง ซึ่งจะช่วยบ่งบอกถึงจำนวน Topic ที่เหมาะสมสำหรับแต่ละกลุ่มปัญหาโดยกำหนดให้จำนวน word ต่อ 1 Topic คือ 10 คำ

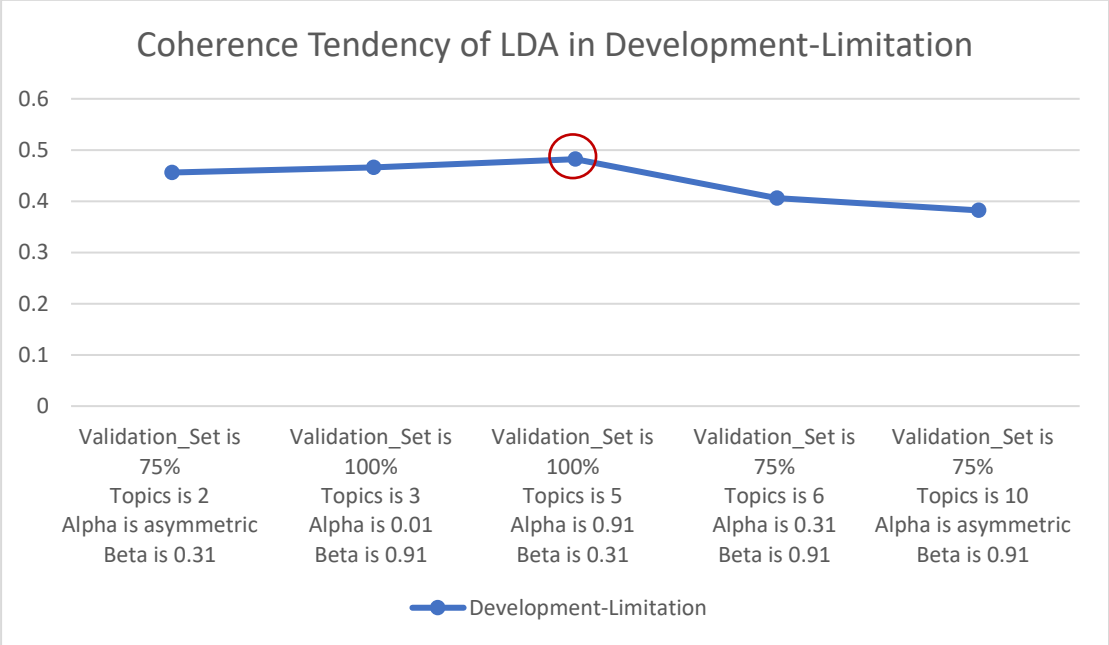
ผลของค่า Coherence ที่คำนวณได้จากการทดลองสำหรับแต่ละกลุ่มปัญหาแสดงดังรูปที่ 6-7 ถึงรูปที่ 6-15



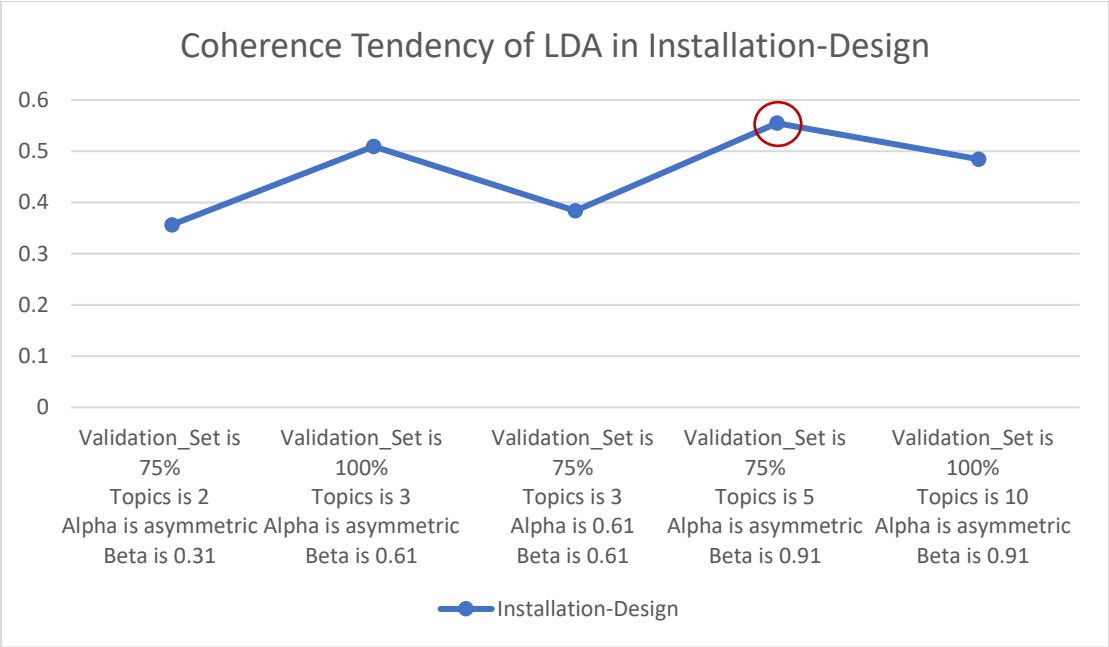
รูปที่ 6-7 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Design



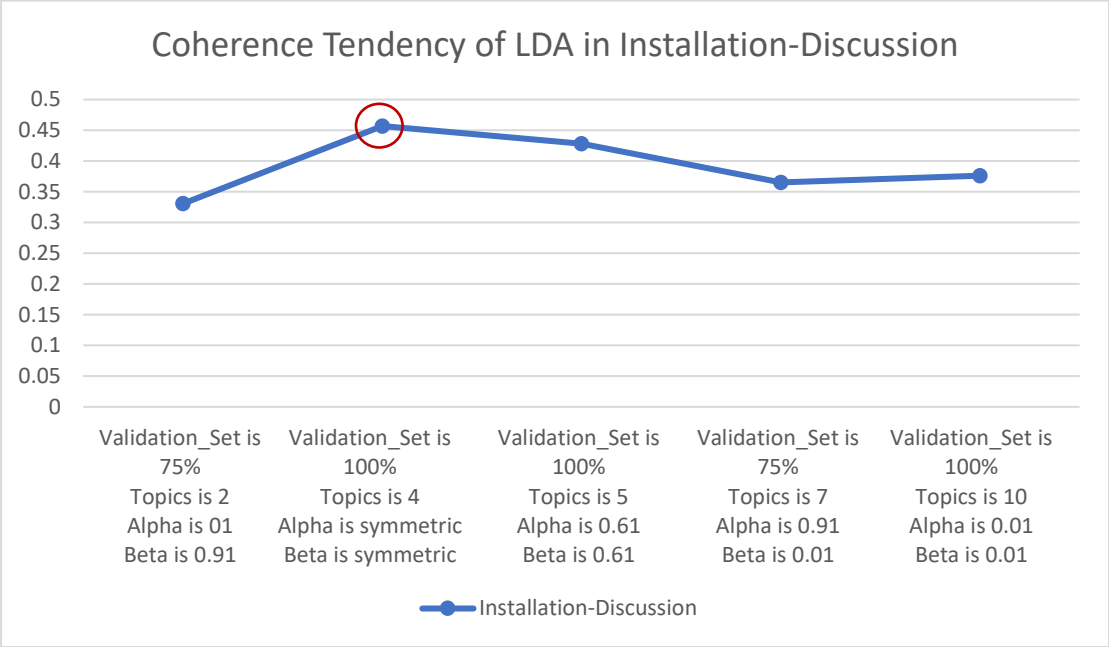
รูปที่ 6-8 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Discussion



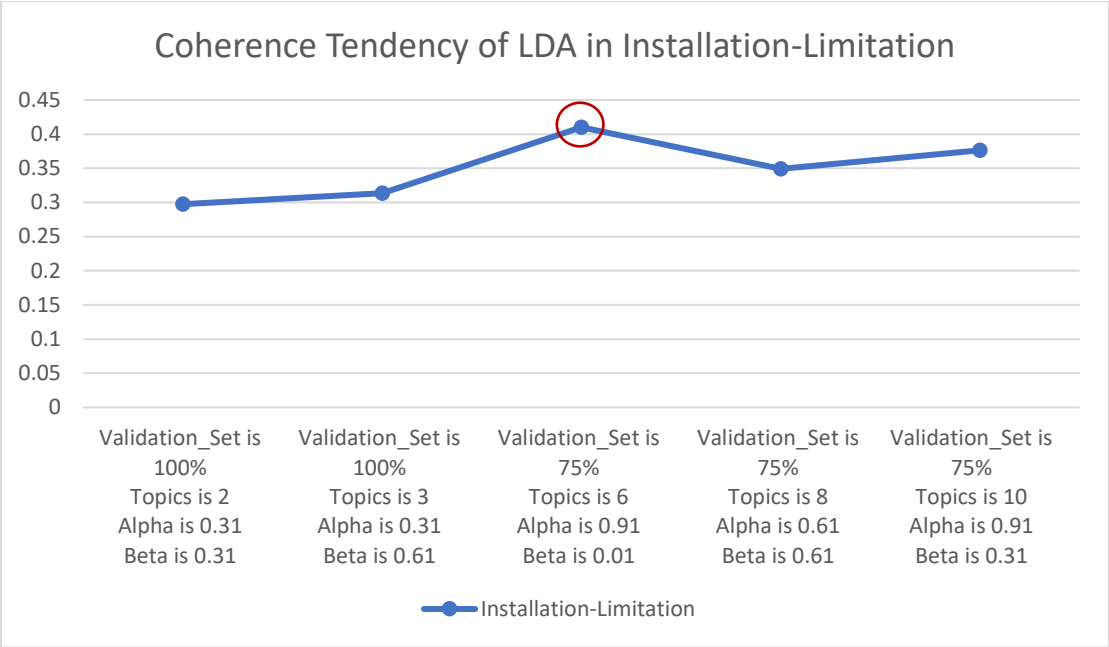
รูปที่ 6-9 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Development-Limitation



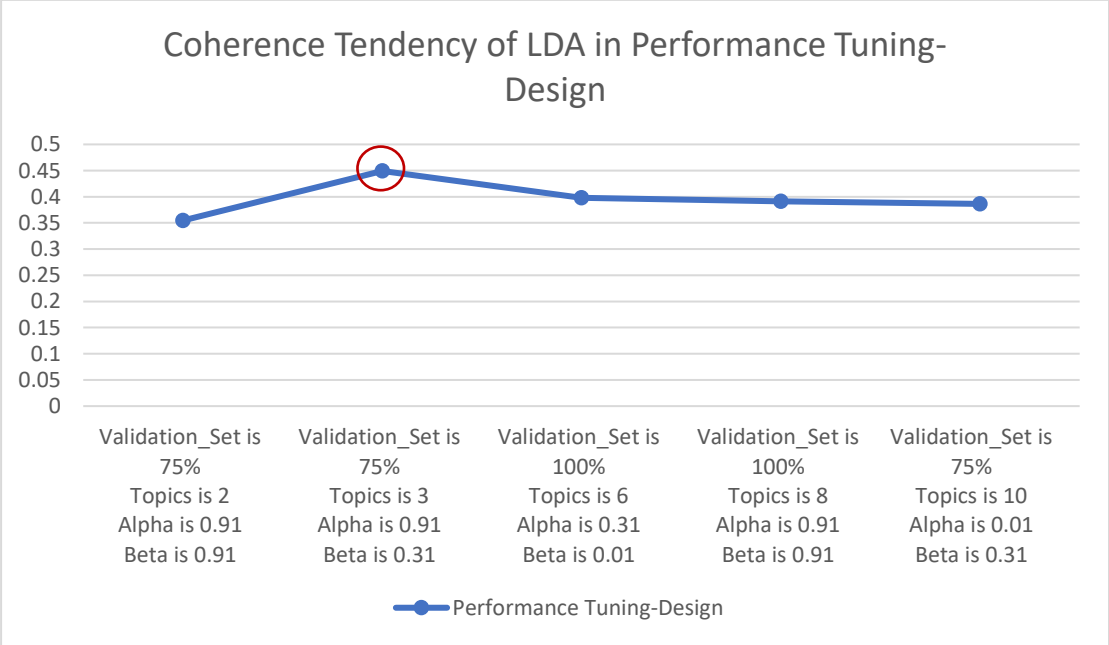
รูปที่ 6-10 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Design



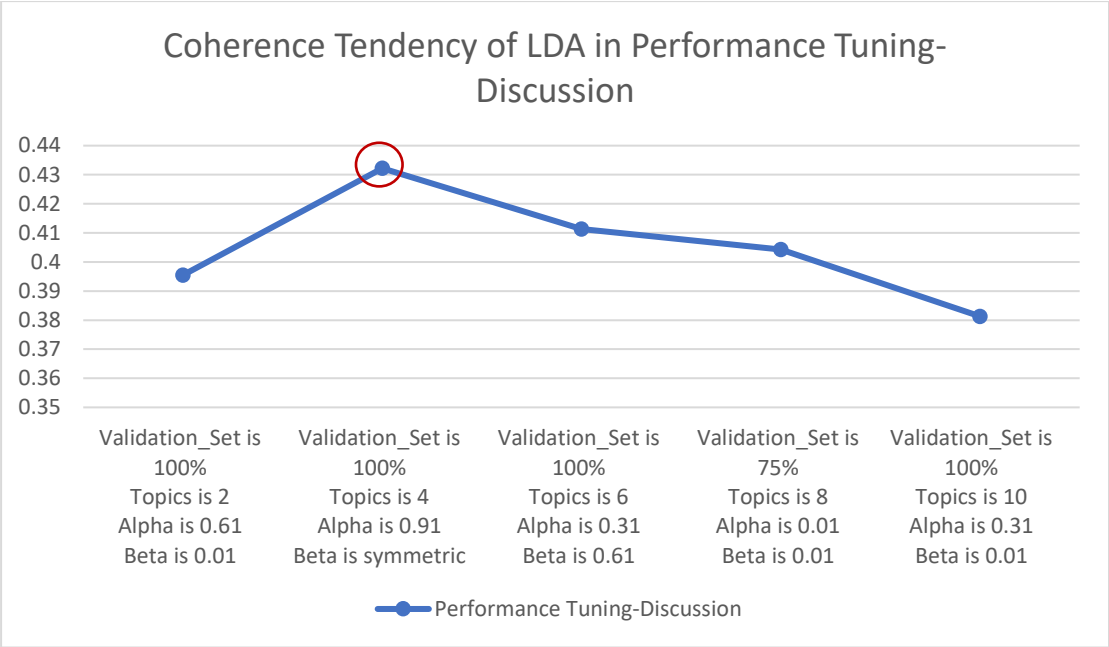
รูปที่ 6-11 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Discussion



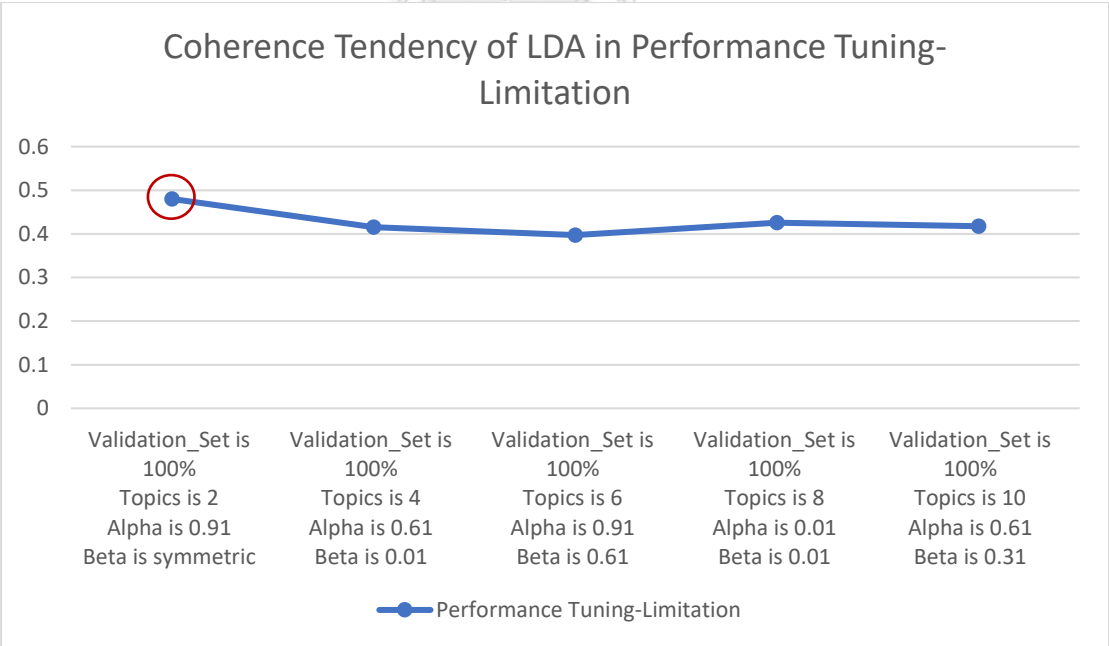
รูปที่ 6-12 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Installation-Limitation



รูปที่ 6-13 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Design



รูปที่ 6-14 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Discussion



รูปที่ 6-15 รูปกราฟแสดงผลการหาค่า Coherence ของกลุ่มปัญหา Performance Tuning-Limitation

จากรูปภาพกราฟทั้ง 9 กลุ่มปัญหาสามารถนำข้อมูลที่ได้มาสรุปเป็นข้อมูลในตารางที่ 6-10 ด้านล่างโดยแยกเป็นตัวแปร Beta, Alpha, Topics(K), Validation Set และค่า Coherence ที่สูงที่สุดจากการทดลองซึ่งบ่งบอกถึงจำนวน Topic ที่เหมาะสมสำหรับแต่ละกลุ่มปัญหา

ตารางที่ 6-10 ตารางสรุปผลค่า Coherence ของ Topics และ ตัวแปรที่ใช้

Class	Validation Set	Beta	Alpha	Coherence	Topics (K)
1	75% Corpus	0.01	asymmetric	0.45548861	2
2	100% Corpus	0.91	0.31	0.482153034	5
3	100% Corpus	0.91	0.31	0.482153034	5
4	75% Corpus	0.91	asymmetric	0.554846033	5
5	100% Corpus	symmetric	symmetric	0.456966243	4
6	75% Corpus	0.01	0.91	0.410113142	6
7	75% Corpus	0.31	0.91	0.449222108	3
8	100% Corpus	symmetric	0.91	0.432270214	4
9	100% Corpus	symmetric	0.91	0.480087689	2

โดยกำหนดให้ค่า class มีค่า 1 ถึง 9 ทำหน้าที่แทนค่าดังต่อไปนี้คือ Development-Design เป็นค่า 1, Development-Discussion เป็นค่า 2, Development-Limitation เป็นค่า 3, Installation-Design เป็นค่า 4, Installation-Discussion เป็นค่า 5, Installation-Limitation เป็นค่า 6, Performance Tuning-Design เป็นค่า 7, Performance Tuning-Discussion เป็นค่า 8, และ Performance Tuning-Limitation เป็นค่า 9

จากรูปกราฟและข้อมูลสรุปในตารางด้านบนพบว่าจะมีค่า k ที่มีค่าระหว่าง 2-6 topic มีการเรียนรู้จากกลุ่มคำเริ่มที่จำนวน 75% ของเอกสารถึง 100% และพบอีกว่าค่า Coherence จะอยู่ในช่วง 0.41-0.55 โดยในแต่ละ topic ได้มีการกำหนดจำนวนคำไว้ที่ 10 คำต่อ topic สำหรับค่า Coherence นั้นจะมีค่าสูงสุดคือ 1.00 ซึ่งค่าที่ได้จากการทดลองจะพบว่าอยู่ที่ 40-55% ซึ่งค่ายังไม่สูงมากนัก แต่จากตัวอย่างงานวิจัยและบทความที่เกี่ยวข้อง [34],[35] พบว่ามีค่า Coherence อยู่ระหว่าง 0.3-0.7 หรือ 30-70 % ผู้วิจัยเห็นว่าเนื่องจากการทำโมเดล LDA เป็นรูปแบบ Probabilistic Model และค่าความแม่นยำมีตัวแปรสำคัญมาจากคำศัพท์ที่ใช้ในการหาหัวข้อซึ่งส่วนนี้จะได้รับผลกระทบมาจากปัญหาก่อนหน้า หากมีการใช้คำที่ซ้ำ ๆ กันและคำจำนวนมากไม่มีความ

เฉพาะเจาะจง จะทำให้ความสัมพันธ์ของคำใน Topic เดียวกัน ไม่สูงมากนักและคำในต่าง Topic กันก็ไม่แตกต่างกันมากนัก ทำให้ค่า Coherence ไม่สูงมากนักด้วย การเพิ่มจำนวนข้อมูลอาจจะส่งผลให้ค่า Coherence สูงขึ้นได้หรือในอนาคตการเปลี่ยนวิธีการเช่น Latent Semantic-Analysis (LSA) หรือ Non-Negative Matrix Factorization (NMF) อาจจะเป็นอีกทางเลือกในการทำ Topic Modeling ที่แม่นยำมากกว่านี้

หลังจากทำการคำนวณค่าจำนวน topic ที่ดีที่สุดของแต่ละกลุ่มปัญหา (class) ทางผู้วิจัยได้ทำการสร้างโมเดลจากผลดังกล่าว โดยได้คำที่มาจากการทำ topic modeling (LDA) สำหรับแต่ละกลุ่มปัญหาในตารางที่ 6-11 ถึงตารางที่ 6-19 และจากกลุ่มคำที่ได้ผู้ใช้ต้องทำการตีความเพื่อให้สามารถเข้าใจความหมายของแต่ละ topic และนำไปใช้งานได้ [36]

ตารางที่ 6-11 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Design

Topic	1	2	3	4	5	6	7	8	9	10
1	table	select	from	as	query	where	and	column	join	like
2	server	data	date	table	time	get	like	need	try	would

ในตารางที่ 6-11 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Development-Design โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 2 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Development of query design for select“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Development of table design“

ตารางที่ 6-12 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Discussion

Topic	1	2	3	4	5	6	7	8	9	10
1	index	function	create	view	use	explain	or	procedure	exists	name
2	select	from	as	date	query	where	and	count	table	join
3	error	server	run	command	try	get	set	use	file	user
4	table	column	insert	id	one	want	like	update	value	data
5	query	find	like	collection	field	get	array	want	document	data

ในตารางที่ 6-12 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Development-Discussion โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 5 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Development discussion about SQL function“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “ Development discussion about SQL select“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Development discussion about system-command”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Development discussion about SQL update”

topic 5 สามารถตีความหมายได้เป็นหัวข้อ “Development discussion about NOSQL”

ตารางที่ 6-13 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Development-Limitation

Topic	1	2	3	4	5	6	7	8	9	10
1	and	time	or	like	date	query	as	from	select	where
2	table	constraint	create	key	column	insert	default	add	delete	update
3	as	select	set	from	and	date	procedure	string	function	case
4	server	data	file	error	run	station	command	use	get	user
5	select	from	table	qurey	limit	where	insert	get	join	as

ในตารางที่ 6-13 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Development-Limitation โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 5 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Development limitation of SQL select (related to -time)“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “ Development limitation of NOSQL“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Development limitation of SQL function”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Development limitation of system command”

topic 5 สามารถตีความหมายได้เป็นหัวข้อ “Development limitation of SQL select (related to -insert)”

ตารางที่ 6-14 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Design

Topic	1	2	3	4	5	6	7	8	9	10
1	table	system	select	from	query	insert	command	like	set	get
2	as	select	table	sum	from	query	date	join	get	and
3	table	data	server	one	user	want	would	like	set	create
4	server	error	try	get	connect	run	file	install	user	version
5	in	or	at	as	database	default	create	alter	try	password

ในตารางที่ 6-14 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Installation-Design โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 5 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “SQL select (related to insert) in installation design“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “SQL select (related to time) in installation design“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Create system table in installation design”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Install and test in installation design”

topic 5 สามารถตีความหมายได้เป็นหัวข้อ “Admin Command in installation design”

ตารางที่ 6-15 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Discussion

Topic	1	2	3	4	5	6	7	8	9	10
1	error	try	get	run	user	start	work	command	password	data
2	server	error	connect	try	instance	connection	run	install	studio	get
3	table	server	query	like	would	get	column	need	insert	one
4	set	replica	version	select	and	update	see	node	index	error

ในตารางที่ 6-15 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Installation-Discussion โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 4 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ“Installation discussion about admin command“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Installation discussion about install and test“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Installation discussion about NOSQL”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Installation discussion about SQL select (related to update)”

ตารางที่ 6-16 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Installation-Limitation

Topic	1	2	3	4	5	6	7	8	9	10
1	default	by	date	server	error	run	name	table	import	unique
2	password	try	server	instance	connect	version	like	root	work	table
3	server	error	know	get	try	create	database	write	key	tutorial
4	table	default	query	and	from	select	join	set	as	create
5	server	install	want	use	application	service	folder	start	query	also
6	server	file	error	user	run	get	work	start	try	machine

ในตารางที่ 6-16 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Installation-Limitation โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 6 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to programming - (not SQL)“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to admin - command“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to SQL create”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to SQL select”

topic 5 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to install and - test”

topic 6 สามารถตีความหมายได้เป็นหัวข้อ “Installation limitation related to system”

ตารางที่ 6-17 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Design

Topic	1	2	3	4	5	6	7	8	9	10
1	table	select	update	insert	set	from	query	as	count	value
2	and	select	as	from	query	table	where	join	index	time
3	table	data	server	like	time	one	query	use	would	get

ในตารางที่ 6-17 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Performance Tuning-Design โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 3 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning design for SQL update“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning design for SQL index“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning design for system”

ตารางที่ 6-18 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Discussion

Topic	1	2	3	4	5	6	7	8	9	10
1	update	insert	set	value	want	like	document	array	field	new
2	as	join	and	select	query	default	from	table	where	date
3	table	select	query	from	where	index	time	column	like	and
4	table	server	data	would	error	run	one	try	file	get

ในตารางที่ 6-18 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Performance Tuning-Discussion โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 4 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning discussion about NOSQL“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning discussion about SQL select (related to time)“

topic 3 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning discussion about SQL index”

topic 4 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning discussion about system”

ตารางที่ 6-19 ตารางสรุปคำศัพท์สำหรับ topic ที่อยู่ภายใต้ปัญหาเรื่อง Performance Tuning-Limitation

Topic	1	2	3	4	5	6	7	8	9	10
1	update	insert	set	value	want	like	document	array	field	new
2	as	join	and	select	query	default	from	table	where	date

ในตารางที่ 6-19 จะเป็นคำที่ได้จากปัญหาในกลุ่ม Performance Tuning-Limitation โดยในกลุ่มนี้จำนวน topic ที่เหมาะสมคือ 2 โดยที่

topic 1 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning limitation related to programming (not SQL)“

topic 2 สามารถตีความหมายได้เป็นหัวข้อ “Performance tuning limitation related to SQL select“

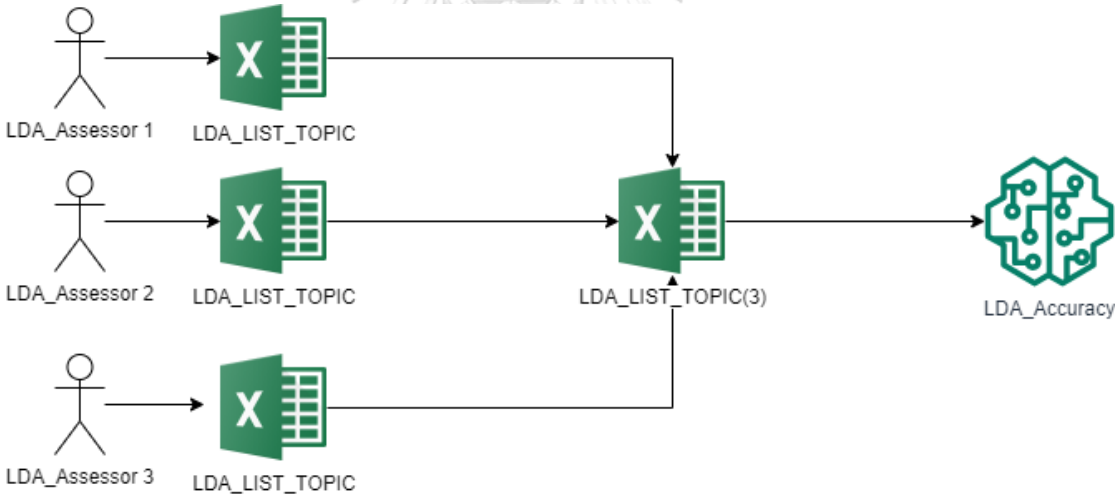
<u>ส่วนที่ 2</u> การประเมินความแม่นยำในการนำไปใช้งานจริง จากการสร้างและตีความ topic ทั้ง 9 กลุ่มปัญหา ลำดับถัดมาทางผู้วิจัยได้ทำการประเมินความแม่นยำในการนำไปใช้งาน โดยการใช้ผู้มี

ประสบการณ์ด้านฐานข้อมูลเป็นเวลา 1-2 ปี จำนวน 3 ท่าน ในการประเมินความแม่นยำ โดยใช้วิธีการสุ่มนำข้อมูล 4 โพสต์ ต่อ 1 topic เพื่อมาประเมินเนื้อหาของโพสต์ว่าตรงตาม topic ตามการตีความจาก LDA หรือไม่ โดยรายละเอียดแต่ละกลุ่มปัญหาและสรุปจำนวนโพสต์ที่ใช้ทดสอบดังตารางที่ 6-20 ด้านล่าง

ตารางที่ 6-20 ตารางสรุปจำนวน topic และ post ที่ใช้ทดสอบโมเดล

Class	1	2	3	4	5	6	7	8	9
No. of topics	2	5	5	5	4	6	3	4	2
No. of posts under topics	8	20	20	20	16	24	12	16	8

วิธีการในขั้นตอนนี้จะทำการแจกข้อมูลที่ใช้ทดสอบทั้งหมดให้กับผู้ประเมินทั้ง 3 ท่าน เพื่อให้ทำการประเมินโดยการอ่านและตีความหมายว่า โพสต์มีความหมายตรงกับหัวข้อ topic ที่ตีความจาก LDA หรือไม่ ซึ่งมีลำดับการทำงานตามรูปที่ 6-16 ด้านล่าง



รูปที่ 6-16 flow การประเมินผลโมเดล LDA กับผู้ประเมิน

ตัวอย่างข้อมูลโพสต์ที่สุ่มพร้อมจับคู่กับ topic เป็นข้อมูลจากกลุ่มปัญหา Development-Design หรือ Class 1 โดย topic 1 หมายถึง “Development of query design for select“ และ topic 2 หมายถึง “Development of table design“ ในส่วนนี้ผู้ประเมินจะทำการประเมินจากข้อความทั้งหมดในโพสต์ดังตัวอย่างในตารางที่ 6-21 ว่าสอดคล้องกับ topic ที่แนะนำหรือไม่

ตารางที่ 6-21 ตารางแสดงตัวอย่างข้อมูลที่ส่งให้ผู้เชี่ยวชาญอ่านและประเมินผลโดยเป็นข้อมูลจากกลุ่มปัญหา Development-Design (แสดงเฉพาะส่วน Post Header)

Topic	Post Header
1	How can I avoid calling a stored procedure from a UDF in SQL Server
1	How can I get the data with select query with not writing all columns extenally and date formating is possible
1	Finding similar substrings in two separate strings
1	Grouping rows by time span
2	Can I logically reorder columns in a table?
2	Can a record with a higher IDENTITY value be inserted before one with a lower IDENTITY value in SQL Server with large number of user?
2	How can I get only one row if only one column is different
2	How can I aggregate sales by Month in MongoDB to Scale / Report

โดยผลสรุปจากทั้ง 3 ท่านเมื่อนำมารวบรวมจะได้ผลตามตารางที่ 6-22 ด้านล่างและสามารถสรุปหาค่าความแม่นยำได้โดยการคำนวณจำนวนโพสต์ที่ผู้ประเมินเห็นด้วยกับหัวข้อที่ LDA แนะนำหารด้วยจำนวนโพสต์ที่ใช้ในแต่ละกลุ่มปัญหาและ ทั้งหมดหารด้วยจำนวนผู้ประเมิน ซึ่งในที่นี้คือ 3 โดยได้ผลตามตารางที่ 6-23

ตารางที่ 6-22 ตารางสรุปผลการประเมินเปรียบเทียบผลจากโมเดลและมุมมองของผู้เชี่ยวชาญ

	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class 9
No. of topics	2	5	5	5	4	6	3	4	2
No. of Posts under topic	8	20	20	20	16	24	12	16	8
No. of posts under topics distributed to evaluators	24	60	60	60	48	72	36	48	24
No. of posts under topics agreed by evaluators	14	24	23	35	17	36	27	28	15

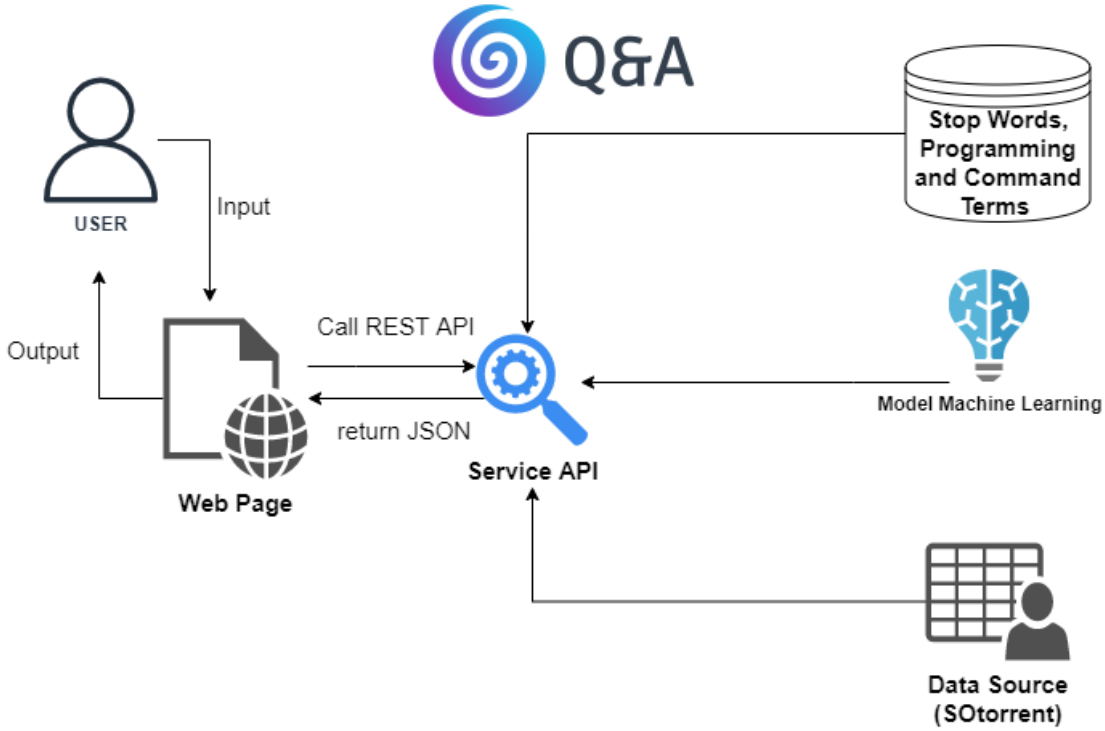
ตารางที่ 6-23 ตารางค่าความแม่นยำของโมเดลจากการเฉลี่ยผลการประเมิน

Class no	1	2	3	4	5	6	7	8	9
LDA Accuracy	58%	55%	38%	58%	35%	50%	75%	58%	62%

จากผลการประเมินจะพบว่าการระบุหัวข้อ topic ตามวิธีการของ LDA ให้กับโพสต์คำถามที่เกี่ยวข้องกับกลุ่มปัญหาต่าง ๆ ส่วนใหญ่มีความแม่นยำปานกลางที่ระดับ 50-62% กลุ่มปัญหา Performance Tuning (Class 7-9) มีความแม่นยำค่อนข้างสูงพอสมควรที่ระดับ 58-75% เมื่อเทียบกับกลุ่มปัญหากลุ่มอื่น ๆ ทั้งนี้น่าจะเป็นผลมาจากการที่เนื้อหาโพสต์ของ Performance Tunning ค่อนข้างมีคำที่แยกปัญหาได้ชัดเจนกว่า และหัวข้อ Topic ที่วิเคราะห์ได้ค่อนข้างมีความเฉพาะเจาะจงและตีความได้ค่อนข้างชัดเจนกว่า เมื่อเทียบกับโพสต์ในกลุ่มปัญหา Development และ Installation โดยเฉพาะกลุ่ม Development-Limitation และกลุ่ม Installation-Discission มีความแม่นยำน้อยสุดที่ 38% และ 35% ตามลำดับ ซึ่งน่าจะสืบเนื่องจากการที่เนื้อหาในโพสต์มีความหลากหลายมาก ครอบคลุมประเด็นต่าง ๆ กว้างมาก ทำให้การตีความเนื้อหาคลาดเคลื่อนไปได้ การวิเคราะห์ด้วยวิธีด้วย LDA ที่ให้กลุ่มคำออกมาเป็น topic และต้องอาศัยการตีความโดยมนุษย์-อีกครั้ง กลุ่มคำเหล่านั้นหมายถึง topic อะไร จึงยังมีความคลาดเคลื่อนอยู่มากได้ หากข้อมูลที่ใช้วิเคราะห์มีลักษณะ เนื้อหาที่หลากหลายมาก ทำให้คำที่ประกอบกันเป็นหัวข้อ topic จึงยังไม่ชัดเจนและทำให้การตีความหัวข้อทำได้ยากขึ้น

บทที่ 7
การพัฒนาเครื่องมือช่วยเหลือเจ้าของผลิตภัณฑ์และเครื่องมือช่วยค้นหาตัวอย่างกลุ่มปัญหาที่พบ

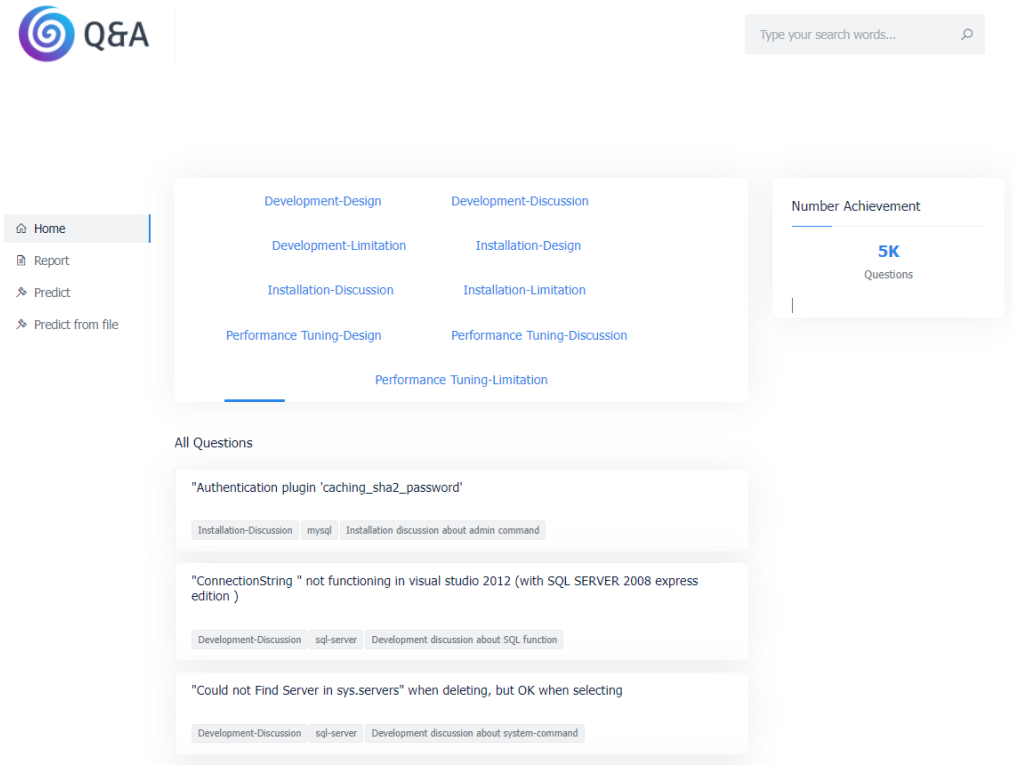
ในหัวข้อนี้หลังจากที่ผู้วิจัยได้ทำการรวบรวมข้อมูล สร้างโมเดลในการจำแนกปัญหา ทั้งในระดับปัญหา 9 กลุ่ม และวิเคราะห์หัวข้อภายใต้ปัญหาแต่ละกลุ่มจากการทำ topic modeling ลำดับถัดมาจะเป็นส่วนของการนำไปทดลองใช้งาน โดยในงานวิจัยได้พัฒนาระบบต้นแบบในการออกรายงาน และตัวทดลองในการจำแนกโพสต์ตามแต่ละกลุ่มปัญหาต่าง ๆ ในรูปแบบของเว็บแอปพลิเคชัน และมีระบบต้นแบบเว็บเซอร์วิซในที่นี้ได้เลือกใช้แบบ RESTful API เบื้องต้นสำหรับการนำไปพัฒนาต่อยอดการใช้โมเดลผ่านเว็บเซอร์วิซ สำหรับโครงสร้างการทำงานของตัวระบบต้นแบบจะมีรูปแบบและ สถาปัตยกรรม ตามรูปที่ 7-1 ด้านล่าง



รูปที่ 7-1 ภาพโครงสร้างระบบและการทำงานเบื้องต้น

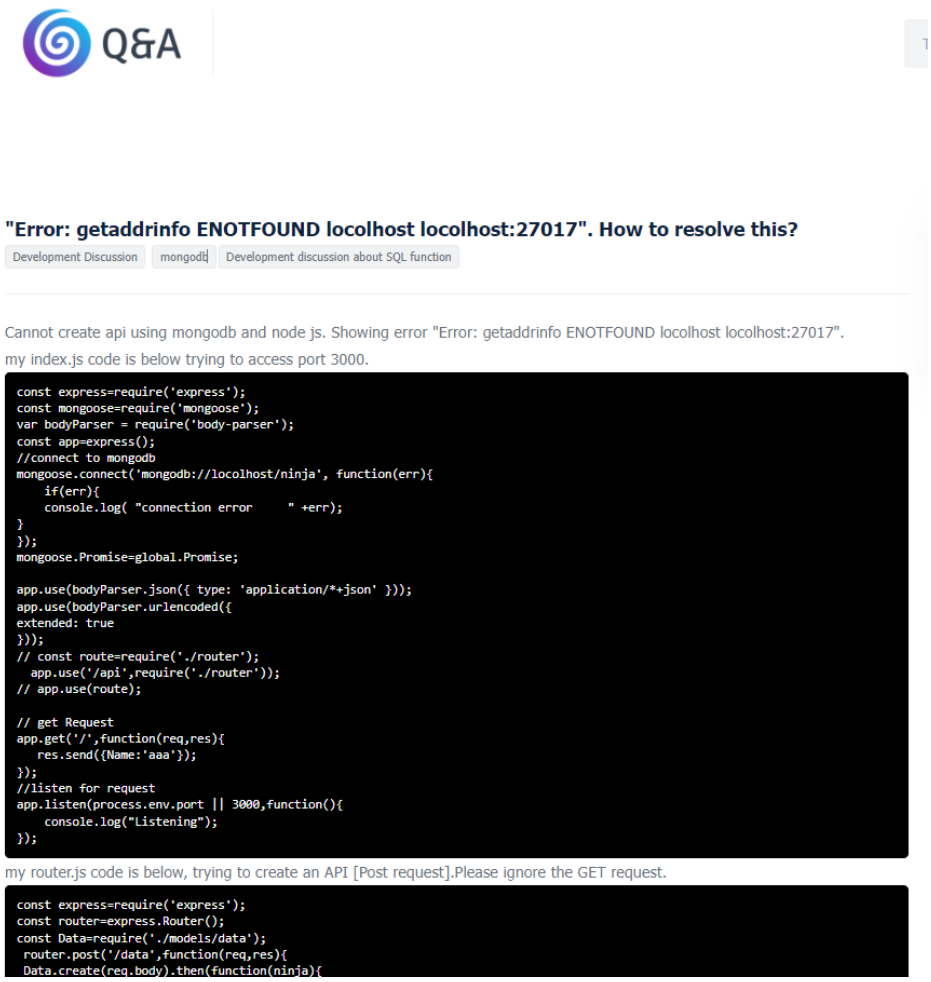
การทำงานจะแยกเป็น 3 ส่วนหลัก ๆ ประกอบด้วยส่วนที่ 1 คือหน้าจอแสดงผลและสั่งการโดยนำเสนอในรูปแบบเทคโนโลยีเว็บ 1 ส่วนที่ 2 เป็นตัวเว็บเซอร์วิซ หรือ เซอร์วิซเอพีไอ โดยจะทำหน้าที่คอยรับส่งข้อมูล เป็นตัวกลางสำหรับการประมวลผลเพื่อส่งข้อมูลไปหาผู้ใช้งาน และสามารถใช้งานโมเดลได้โดยไม่จำเป็นต้องสั่ง หรือส่งข้อมูล หรือเขียนโปรแกรมเสริมเข้าไป ส่วนที่ 3 จะเป็นส่วนที่มีไว้เพื่อการเรียกใช้งานผ่านทางเว็บเซอร์วิซเท่านั้น โดยมีฐานข้อมูลจากเว็บไซต์สแต็กโอเวอร์โฟลว์ มีคลังคำศัพท์ของคำหยุดและคำทั่ว ๆ ไป ที่พบบ่อย ๆ ในประโยคโดยจะเป็นคำจากภาษาคอมพิวเตอร์และคำสั่งคอมพิวเตอร์ เช่น mkdir เป็นต้น และสุดท้าย โมเดล โดยจะแยกเป็นสองส่วนย่อย คือ โมเดลในการจำแนกกลุ่มปัญหาทั้ง 9 กลุ่ม และโมเดลวิเคราะห์หัวข้อภายใต้ 9 กลุ่มปัญหา (Topic Modeling)

ในส่วนของเทคโนโลยีที่ใช้ประกอบด้วยตัว NodeJS เป็นตัวช่วยในการจัดการหน้าจอแสดงผลผู้ใช้, Python Flask WSGI จะทำหน้าที่เป็นเว็บเซิร์ฟเวอร์และ Rest Service API Provider และภาษาที่ใช้จะเป็นภาษา JavaScript, HTML และ Python



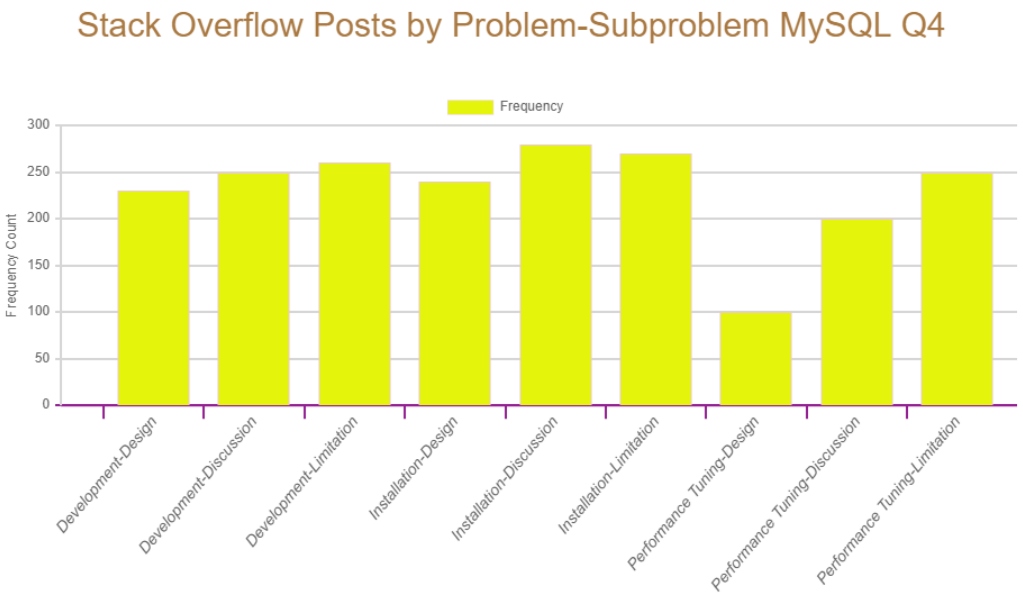
รูปที่ 7-2 ภาพตัวอย่างหน้าจอระบบในหน้าแรก

จากรูปที่ 7-2 จะเป็นตัวอย่างหน้าจอระบบในหน้าแรก ในหน้าดังกล่าวจะแสดงโพสต์ทั่ว ๆ ไปออกมาซึ่งจะมี tag ของกลุ่มปัญหาและหัวข้อปรากฏอยู่และสามารถค้นหาโพสต์ต่าง ๆ ได้ที่ช่องค้นหา และเมื่อคลิกเข้าไปในแต่ละโพสต์ จะเข้าไปพบหน้ารายละเอียดของโพสต์นั้น ๆ ตามรูปที่ 7-3 โดยการออกแบบจะทำให้ใกล้เคียงกับตัวต้นฉบับ Stack Overflow ให้มากที่สุด ทั้งในแง่การแสดงผลและการค้นหา เป็นต้น

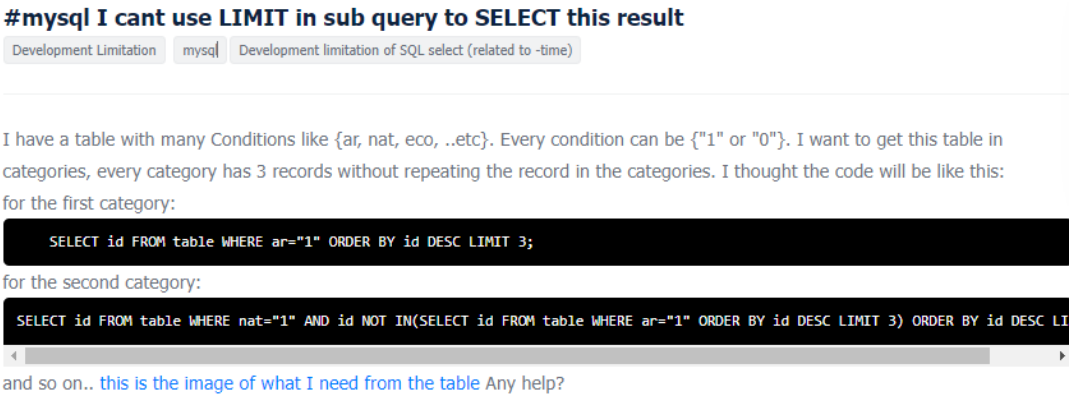


รูปที่ 7-3 ภาพตัวอย่างหน้ารายละเอียดของโพสต์

ในส่วนถัดมาจะมีระบบรายงานผลหรือ Report ต่าง ๆ ตามภาพตัวอย่างรูปที่ 7-4 สามารถแยกตามแต่ละแท็กได้เนื่องจากจุดประสงค์หลักจะเน้นเพื่อให้กลุ่มผู้ใช้ที่เป็นเจ้าของผลิตภัณฑ์ฐานข้อมูลสามารถดูรายละเอียดได้ว่าในแต่ละกลุ่มปัญหามีอะไรบ้าง มีการพูดถึงหรือสอบถามในมุมใด เพื่อในอนาคตสามารถนำไปปรับปรุงแก้ไขได้



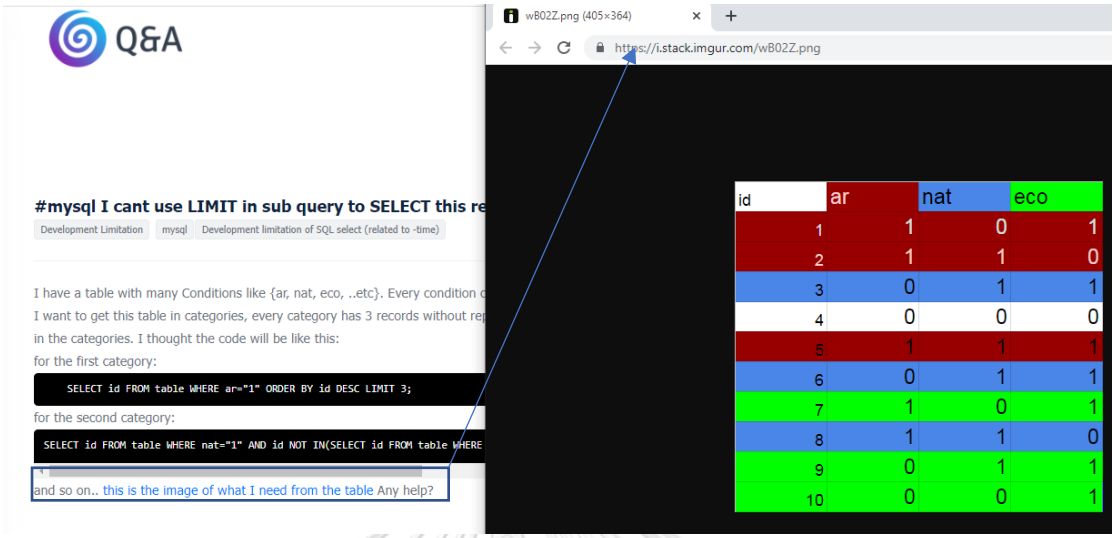
รูปที่ 7-4 ตัวอย่างกราฟแท่ง (Bar) แสดงความถี่ของแต่ละปัญหาของฐานข้อมูล MySQL ในช่วง-Q4 ปี 2019



รูปที่ 7-5 ตัวอย่างการแสดงผลที่แตกต่างกันระหว่างเนื้อหากับภาษาโปรแกรม

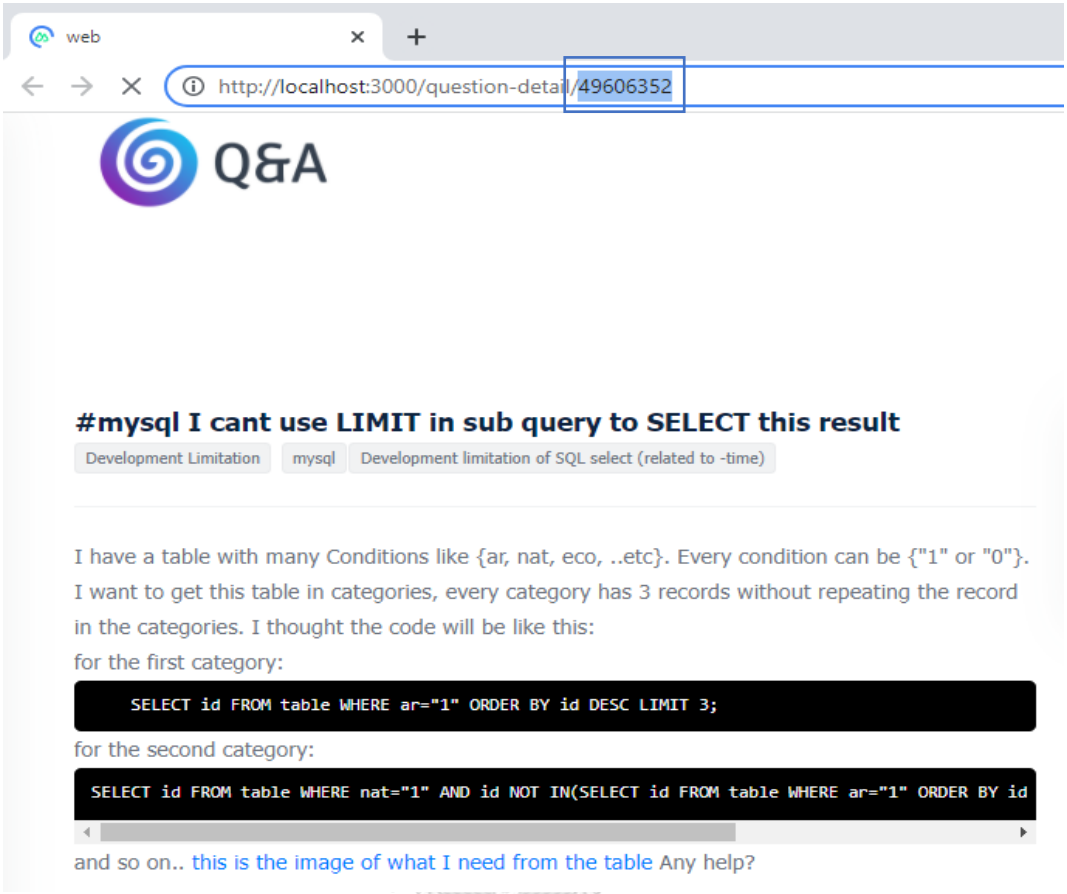
หลังจากที่ใช้โมเดลจำแนกปัญหา 9 กลุ่ม ระบบจะทำนายและนำเสนอในรูปแบบของแท็ก (Tags) ดังในภาพตัวอย่างที่รูปที่ 7-5 จากโพสต์เป็นกลุ่มปัญหาของ Development-Limitation และสำหรับข้อมูลการแสดงผลทางผู้วิจัยได้ทำการออกแบบการแสดงผลให้ใกล้เคียงเว็บไซต์ดั้งเดิมของข้อมูลคือ เว็บสแต็กโอเวอร์โฟลว์ ซึ่งจะเห็นว่าหน้าการแสดงผลได้แยกส่วนของเนื้อหาออกคือ เนื้อหาที่เป็นคำพูดทั่วไปที่เป็นการถามปัญหา และเนื้อหาที่เป็นภาษาคอมพิวเตอร์ดังเช่นในรูปที่ 7-5 เป็น

ภาษา SQL โดยจะมีแยกส่วนเป็นพื้นหลังสีดำชัดเจน และในโพสต์ต้นฉบับของเนื้อหา ทางผู้วิจัยได้ทำการเก็บลิงค์เชื่อมโยงข้อมูลไว้ เพื่อทำการสืบค้นได้ง่ายขึ้น ดังรูปที่ 7-6

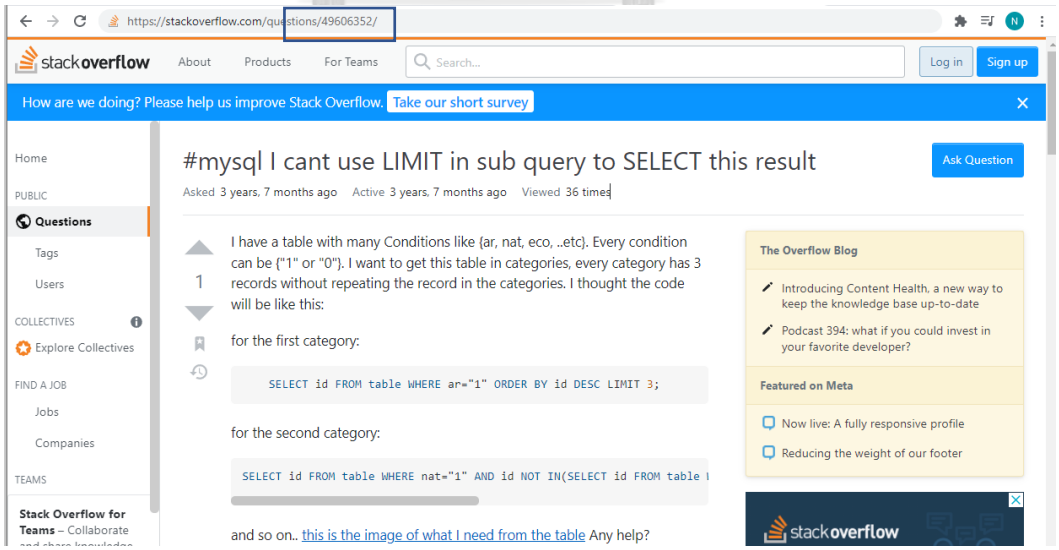


รูปที่ 7-6 ตัวอย่างการแสดงผล และทดสอบการเชื่อมโยง link ของข้อมูล

ผู้วิจัยได้ทำการเก็บเลขเรียกหรือเลขประจำโพสต์ (ID) ที่เป็นตัวเดียวกับที่ใช้ในเว็บไซต์สแต็กโอเวอร์โฟลว์เพื่อให้สามารถสืบค้นย้อนกลับไปที่ต้นฉบับได้ เพราะเนื่องจากข้อมูลในระบบที่นำเสนอไม่ได้นำข้อมูลทั้งหมดของโพสต์นั้นออกมาจัดเก็บ ทำให้จำเป็นต้องใช้และจัดเก็บรหัสดังกล่าวให้เหมือนเดิมโดยมีตัวอย่างเป็นเลข 49606352 และรูปด้านล่างทั้งสองรูป โดยรูปที่ 7-7 จะเป็นรูประบบงานวิจัย รูปที่ 7-8 เป็นเว็บไซต์ดังเดิม ซึ่งจะเห็นได้ว่าใช้รหัสชุดเดียวกัน



รูปที่ 7-7 ตัวอย่างการใช้เลขรหัสเจ้าของโพสต์ในระบบของงานวิจัย



รูปที่ 7-8 ตัวอย่างการใช้เลขรหัสเจ้าของโพสต์ในเว็บไซต์ต้นฉบับ

ในส่วนของการเข้าถึงข้อมูลและการนำไปใช้งานต่อได้ในอนาคตทางผู้วิจัยได้ทำการเตรียมเว็บ API (Application Programming Interface) ไว้เบื้องต้นสำหรับการนำไปใช้งานโดยมีรายละเอียดสำคัญดังตารางที่ 7-1

ตารางที่ 7-1 สรุปรายการ API ที่เป็นส่วนประกอบสำคัญของระบบ

HTTP Methods	API Name	Input	Output	note
GET	list_all_post	none	all post by relevance	
GET	list_by_tags	tags	list of results (post by tags)	
GET	list_by_post	post id	list of results (post by post id)	
GET	search	keyword	list of results	ค้นหา post และ tags
POST	prediction _post	post name , post details	post tags problem	
POST	prediction _post_lda	post name , post details , post tags problem	Topics under post tags problem	
GET	report_summary_count	none	summary count of problem	

ในตารางจะมีฟังก์ชันการทำงานที่สำคัญได้แก่ “list_all_post” ตัวชุด API ดังกล่าวจะทำหน้าที่สำคัญคือเรียกข้อมูลทั้งหมดที่มีออกมาจากคลังระบบผู้วิจัย สามารถนำไปประยุกต์ใช้งาน หรือปรับการเรียกใช้แสดงผลได้โดยใช้ฟังก์ชัน “list_by_tags” เพื่อดึงข้อมูลที่มี tag ที่ต้องการหรือ “list_by_post” เพื่อดึงข้อมูล post id ที่ต้องการ นอกจากนี้ฟังก์ชัน “search” จะทำหน้าที่สำคัญคือ การค้นหาคำที่ต้องการในคลังข้อมูลโดยค้นหาในเอกสารทุก ๆ ตำแหน่งหรือใกล้เคียงที่สุด ส่วนฟังก์ชัน “prediction_post” จะใช้เพื่อทำนายกลุ่มปัญหา-ปัญหาย่อยของโพสต์ ฟังก์ชัน “prediction_post_lda” จะใช้เพื่อทำนายหัวข้อ topic ที่กล่าวถึงในโพสต์และฟังก์ชัน “report_summary_count” จะรายงานจำนวนปัญหาที่มีโพสต์เข้ามา

บทที่ 8
สรุปผลงานวิจัยและข้อเสนอแนะ

8.1 สรุปผลงานวิจัย

สำหรับงานวิจัยนี้ผู้วิจัยได้นำเสนอแนวคิด วิธีการ และต้นแบบของเครื่องมือ ที่ช่วยในการแก้ไขปัญหาและการสำรวจกลุ่มปัญหาในเทคโนโลยีกลุ่มฐานข้อมูล โดยมีการยกเทคโนโลยีฐานข้อมูล 5 ตัวประกอบด้วย MySQL, Oracle, SQLSERVER, PostgreSQL, MongoDB โดยใช้ข้อมูลการถามคำถามจากกระดานถามตอบของเว็บไซต์สแต็กโอเวอร์โฟลว์ ที่เป็นกระดานถามตอบที่มีขนาดใหญ่ของวงการเทคโนโลยี

ในเป้าหมายของงานวิจัยนี้ในครั้งแรกตั้งใจสำรวจและค้นหาอินไซด์ของข้อมูลในเบื้องต้นเท่านั้น แต่หลังจากที่ได้ทำการสำรวจขั้นต้นและมองเห็นปัญหาบางส่วนที่ข้อมูลในเว็บไซต์ดังกล่าวอาจจะหาคำตอบได้ จากปัญหาจำพวกปัญหาการใช้งานทั่วไป ปัญหาการเปรียบเทียบข้อดี ข้อเสียของแต่ละเทคโนโลยี และปัญหาอื่น ทางผู้วิจัยจึงได้ศึกษาและทดลองในแบบต่าง ๆ เพื่อสร้างโมเดลการเรียนรู้ของเครื่องให้สามารถจำแนกปัญหาต่าง ๆ ได้โดยไม่ต้องใช้ผู้ใช้งานทั่วไปในการแยกกลุ่มโดยมีการจำแนกกลุ่มปัญหาออกเป็น 3 กลุ่มได้แก่ ปัญหาด้าน Development, Installation, Performance Tuning ในแต่ละกลุ่มปัญหายังแบ่งออกเป็นปัญหาย่อยอีก 3 ด้าน ได้แก่ Design, Discussion, และ Limitation รวมทั้งหมดเป็น 9 กลุ่มปัญหา และผู้วิจัยยังได้เสนอแนวทางพร้อมโมเดลในการหาหัวข้อของประเด็นปัญหาในเชิงลึกจากปัญหาขั้นต้น และสุดท้ายผู้วิจัยได้นำเสนอต้นแบบระบบในการช่วยค้นหาและแยกกลุ่มปัญหาตามที่งานวิจัยนี้ได้จำแนกไว้

จากผลการทดลองในงานวิจัยในส่วนของโมเดลการเรียนรู้ของเครื่องในการจำแนกปัญหา 9 กลุ่ม ปัญหา พบว่าการใช้อัลกอริทึมสำหรับการเรียนรู้ของเครื่องในกลุ่ม Ensemble Learning ให้ประสิทธิภาพโดยรวมที่ดีที่สุดในทุกกลุ่มปัญหาทั้ง 9 กลุ่ม โมเดลที่ได้จากการเรียนรู้ของเครื่องในงานวิจัยนี้ เป็นโมเดลในลักษณะการจำแนกประเภทหลายคลาส (Multiclass Classifier) อัลกอริทึมที่ใช้เป็น Ensemble แบบ Boost ชื่อ XGBoost รองลงมาที่มีประสิทธิภาพใกล้เคียงกันจะเป็น Random Forest แบบปกติ และแบบ VotingClassifier โดยใช้ Random Forest 3 ชุดปรับตัวแปรที่แตกต่างกันมาประกอบ

ผู้วิจัยได้ทดลองกับชุดข้อมูลหลายขนาดในช่วง 5,000-13,000 โพสต์ พบว่า เมื่อใช้ข้อมูลประมาณ 5,000 โพสต์ ทำให้ได้ค่าประสิทธิภาพโดยรวมดีที่สุด โดยโมเดลที่ดีที่สุดที่นำไปใช้พัฒนา

เครื่องมือต่อคือ XGBoost (TF-IDF) โดยมีค่าประสิทธิภาพ Precision เป็น 64% Accuracy เป็น 68% Precision เป็น 65% Recall เป็น 64% และ F1 เป็น 64%

ลำดับถัดมาจะเป็นการสร้างโมเดล Topic Modeling ด้วยวิธี LDA เพื่อใช้ค้นหาหัวข้อที่เป็นประเด็นปัญหาที่มีการพูดถึงภายใต้กลุ่มปัญหาที่จำแนกมาแล้วก่อนหน้าจากข้อมูลที่ใช้ในการทดลอง ทำให้ได้หัวข้อสำหรับแต่ละกลุ่มปัญหาดังนี้

- Development-Design แบ่งหัวข้อปัญหาได้อีก 2 topic
- Development-Discussion แบ่งหัวข้อปัญหาได้อีก 5 topic
- Development-Limitation แบ่งหัวข้อปัญหาได้อีก 5 topic
- Installation-Design แบ่งหัวข้อปัญหาได้อีก 5 topic
- Installation-Discussion แบ่งหัวข้อปัญหาได้อีก 4 topic
- Installation-Limitation แบ่งหัวข้อปัญหาได้อีก 6 topic
- Performance Tuning-Design แบ่งหัวข้อปัญหาได้อีก 3 topic
- Performance Tuning-Discussion แบ่งหัวข้อปัญหาได้อีก 4 topic
- Performance Tuning-Limitation แบ่งหัวข้อปัญหาได้อีก 2 topic

การประเมินความแม่นยำสำหรับการนำ topic modeling มาใช้ในงานวิจัยนี้ได้ทำการให้ผู้ประเมิน 3 ท่านทำการประเมินโดยได้ผลดังนี้

- หัวข้อของปัญหา Development-Design มีค่าความแม่นยำที่ 58%
- หัวข้อของปัญหา Development-Discussion มีค่าความแม่นยำที่ 55%
- หัวข้อของปัญหา Development-Limitation มีค่าความแม่นยำที่ 38%
- หัวข้อของปัญหา Installation-Design มีค่าความแม่นยำที่ 58%
- หัวข้อของปัญหา Installation-Discussion มีค่าความแม่นยำที่ 35%
- หัวข้อของปัญหา Installation-Limitation มีค่าความแม่นยำที่ 50%
- หัวข้อของปัญหา Performance Tuning-Design มีค่าความแม่นยำที่ 75%
- หัวข้อของปัญหา Performance Tuning-Discussion มีค่าความแม่นยำที่ 58%
- หัวข้อของปัญหา Performance Tuning-Limitation มีค่าความแม่นยำที่ 62%

ความแม่นยำในโมเดลแบบ LDA ในหลาย ๆ งานวิจัยเอง มักจะได้ผลความแม่นยำที่ไม่สูงมากและต้องอาศัยการตีความ ในการทดลองและงานวิจัยนี้ก็เช่นกัน จะเห็นได้ว่าค่าที่ได้ไม่สูงมากนักส่วนใหญ่อยู่ในระดับปานกลางทางผู้วิจัยคิดว่าอาจจะมาจากสาเหตุที่กลุ่มคำหรือคำศัพท์มีการกระจายหรือหลากหลายเป็นจำนวนมาก ๆ ทำให้โมเดลแบบดังกล่าวมีความแม่นยำในการสุ่มค่าคำและการเรียนรู้ไม่สูงมากนัก และเนื่องจากโมเดล LDA เป็นโมเดลที่มีการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ทำให้ทิศทางหรือความแม่นยำอาจจะต่ำลงไปได้

หลังจากที่ได้โมเดลทั้ง 2 แบบแล้วในส่วนสุดท้ายของงานวิจัยคือการนำโมเดลมาทดลองใช้โดยในงานวิจัยนี้ได้สร้างต้นแบบระบบพร้อมเชื่อมต่อโมเดลดังกล่าวเพื่อทดสอบและช่วยจำแนกปัญหาต่าง ๆ และค้นหาปัญหาได้ง่ายขึ้น

8.2 ข้อจำกัดในงานวิจัยและปัญหาที่พบ

1). จำนวนข้อมูลและความหลากหลาย โดยที่จำนวนข้อมูลเมื่อคัดแยกแล้วจะพบปัญหาข้อมูลไม่สมมาตร จำนวนข้อมูลแตกต่างกันมากเกินไป ทำให้ในที่สุดต้องใช้เทคนิคการจัดการข้อมูลไม่สมมาตรร่วมด้วย

2). เนื้อหาในเว็บไซต์สแต็กโอเวอร์โฟลว์ในส่วนของเทคโนโลยีซอฟต์แวร์ที่เป็น open source จะมีข้อมูลโพสต์จำนวนมากกว่าพวกที่ไม่ใช่ open source เช่นกรณี MongoDB จะมีโพสต์คำถามมากกว่า Oracle ซึ่งอาจส่งผลต่อเนื้อหาของข้อความปัญหาที่นำมาวิเคราะห์

3). โมเดลที่สร้างขึ้นจากงานวิจัยนี้มีประสิทธิภาพดีพอควรแต่อาจจะต้องทำการปรับข้อมูลหรือหาข้อมูลมาเพิ่มเติม เพื่อเพิ่มประสิทธิภาพ

4). เนื้อหาของโพสต์ที่ผู้วิจัยใช้เป็นแค่เพียงบางส่วนคือส่วนของชื่อโพสต์ และเนื้อหาเท่านั้น

5). โมเดลประเภท LDA มีความแม่นยำไม่มากนัก อาจเป็นเพราะยังมีข้อมูลจำกัดในการเรียนรู้

6). มิติของข้อมูลมีจำนวนมาก ๆ โดยเฉพาะอย่างยิ่งขั้นตอนการสร้างเวกเตอร์คุณลักษณะของข้อมูล มีโอกาสสูงถึง 2-3 ล้านมิติ ทำให้เครื่องที่ใช้สร้างโมเดลไม่สามารถทำงานได้ และการใช้การลดมิติ บางครั้งอาจจะทำให้ค่าความแม่นยำลดน้อยลงไปได้เช่นกัน

8.3 ข้อเสนอแนะ

1). ทดลองเพิ่มประสิทธิภาพโมเดลด้วยการเพิ่มข้อมูลในส่วนอื่น ๆ เข้าไป เช่น ส่วนของคำตอบในโพสต์ และคำตอบต่อเนื่องที่เป็นลักษณะซ้อน ๆ กันในเว็บ

2). เพิ่มคำศัพท์ในฐานข้อมูล สำหรับภาษา programming และคำสั่ง Computer command

3). ขยายความสามารถของเครื่องมือในการนำเข้าข้อมูลจากสแต็กโอเวอร์โฟลว์ได้อย่างอัตโนมัติ หรือนำข้อมูลจากแหล่งอื่น ๆ นอกเหนือจากสแต็กโอเวอร์โฟลว์มารวมด้วยได้เช่นกัน

บรรณานุกรม

1. Delip Rao and Brian McMahan, *Natural Language Processing with PyTorch Build Intelligent Language Applications Using Deep Learning,*. 2019.
2. Benjamin Bengfort, R.B., and Tony Ojeda *Applied Text Analysis with Python Enabling Language-Aware Data Products with Machine Learning* 2018, O'Reilly Media, Inc.
3. Kapadia, S. *Evaluate Topic Models: Latent Dirichlet Allocation (LDA)*. 2019 [cited 2019 22 aug]; Available from: https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.
4. Mitchell, T.M., *Machine Learning*. 1997: McGraw-Hill Science 1-432.
5. *naive_bayes* : 20 Nov 2019 [cited 2019 20]; Available from: https://uc-r.github.io/naive_bayes.
6. *diagrams decision-tree*.
7. สงวนสัตย์, ร.ด.ป., ปัญญาประดิษฐ์ด้วยการเรียนรู้ของเครื่อง ฉบับภาษา *Python* 2019. 375.
8. empirical-software.engineering, *SOTorrent*.
9. *Stack Overflow*. [cited 2019. 20 Nov 2019.]; Available from: https://stackoverflow.com/.
10. *ensemble-methods-for-deep-learning-neural-networks/*. [cited 2019 20 Nov 2019]; Available from: https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/.
11. *types-of-ensemble-methods-in-machine-learning*. [cited 2020 Oct 2020]; Available from: https://towardsdatascience.com/types-of-ensemble-methods-in-machine-learning-4ddaf73879db
12. *XGBoost*. Available from: https://XGBoost.ai/
13. Arshad Ahmad , C.F., Kan Li ,Syed Mohamad Asim ,Tingting Sun,, *Toward Empirically Investigating Non-Functional Requirements of iOS Developers on Stack Overflow*, in IEEE
14. Wang, H.W., Bei & Li, Can & Xu, Ling & He, JianJun & Yang, Mengning., *SOTagRec: A Combined Tag Recommendation Approach for Stack Overflow*, in *ICMAI 2019*

Chegndu, China.

15. Bandeira, A.C., Alberto & Medeiros, Matheus & Paixao, Paulo & Maia, Paulo., *We Need to Talk about Microservices: an Analysis from the Discussions on Stack Overflow,* , in *MSR.* 2019
16. SebastianBaltes, L., ChristophTreud, StephanDiehl, *SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts*. 2018: Gothenburg, Sweden
17. Pingyi Zhou, J.L., Zijiang Yang, and Guangyou Zhou, *Scalable Tag Recommendation for Software Information Sites,*, in *IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)* 2017.
18. IEEE, *IEEE 24765 Systems and software engineering — Vocabulary*

ed. S. edition. 2017-09.

19. Mike Loukides, G.-P.D.M., *System Performance Tuning*, ed. n. Edition.
20. Pierre Bourque, É.d.t.s.É.R.E.D.F., *SWEBOK Guide to the Software Engineering Body of Knowledge* ed. 3. 2014

Software and Systems Engineering Associates (S2EA) , .

21. *dictionary cambridge limitation*. Available from: https://dictionary.cambridge.org/dictionary/english/limitation.
22. *dictionary cambridge Design.*; Available from: https://dictionary.cambridge.org/dictionary/english/Design.
23. *nltk* [cited 2019 20 Nov 2019.]; Available from: https://www.nltk.org/index.html.
24. Hauck, T., *scikit-learn Cookbook* 2014: , Packt Publishing.
25. Chollet, F., *Deep Learning with Python*

2017: Manning Publications

26. *smote-synthetic-data-augmentation-for-tabular-data*. Available from: https://towardsdatascience.com/smote-synthetic-data-augmentation-for-tabular-data-1ce28090debc.
27. Fernando Nogueira , G.L., Dayvid Victor, and Christos Aridas. *SMOTE*. [cited 2019 20 oct]; Available from: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html.
28. *how-voting-classifiers-work*. 2020 1 Oct 2020]; Available from:

https://towardsdatascience.com/how-voting-classifiers-work-f1c8e41d30ff.

29. *overcoming-the-limitations-of-topic-models-with-a-semi-supervised-approach*. [cited 2020 1 Oct 2020]; Available from: https://medium.com/pew-research-center-decoded/overcoming-the-limitations-of-topic-models-with-a-semi-supervised-approach-b947374e0455.
30. *topic-modeling-and-latent-dirichlet-allocation-in-python*. [cited 2020 1 Oct 2020]; Available from: https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24.
31. Blei, D.C., Lawrence & Dunson, David. , *Probabilistic Topic Models: A focus on graphical model design and applications to document and image analysis.* 2010, IEEE signal processing magazine.
32. Michael Röder, A.B., and Alexander Hinneburg. , *Exploring the Space of Topic Coherence Measures.* , in *ACM International Conference on Web Search and Data Mining (WSDM '15)*. 2015, Association for Computing Machinery: New York, NY, USA,. p. 399–408.
33. *topic-coherence-to-evaluate-topic-models*. [cited 2020 1 Oct 2020]; Available from: http://qpleple.com/topic-coherence-to-evaluate-topicmodels/
34. *evaluate-topic-model-in-python-latent-dirichlet-allocation-lda*. [cited 2021 1 Oct 2021]; Available from: https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.
35. Bose, S. *How Stuff Works: A Comprehensive Topic Modelling Guide with NMF, LSA, PLSA, LDA & lda2vec*. 2018 [cited 2019 20 aug]; Available from: https://medium.com/@souravboss.bose/comprehensive-topic-modelling-with-nmf-lsa-plsa-lda-lda2vec-part-1-20002a8e03ae.
36. Chang, J.B.-G., Jordan & Gerrish, Sean & Wang, Chong & Blei, David. , *Reading Tea Leaves: How Humans Interpret Topic Models. Neural Information Processing Systems*. 2009. p. 288-296.

ประวัติผู้เขียน

ชื่อ-สกุล	นาย ณัฐนัย สุวรรณชูชิต
วัน เดือน ปี เกิด	28 ธันวาคม 2535
สถานที่เกิด	กรุงเทพมหานคร ประเทศไทย
วุฒิการศึกษา	ปีการศึกษา 2557 หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ จากคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ปีการศึกษา 2560 เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ จากคณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	639 เพชรเกษม 63แยก 1 แขวงหลักสอง เขตบางแค 10160
ผลงานตีพิมพ์	ปี พ.ศ. 2564 “Classification of Database Technology Problems on Stack Overflow” โดย ณัฐนัย สุวรรณชูชิต และ รศ. ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา ในงานประชุมวิชาการ IEEE/ACIS 19th International Conference On Software Engineering Research, Management Applications (SERA) ในช่วงระหว่างวันที่ 20-22 มิถุนายน พ.ศ. 2564 ณ เมืองคานาซาว่า ประเทศญี่ปุ่น