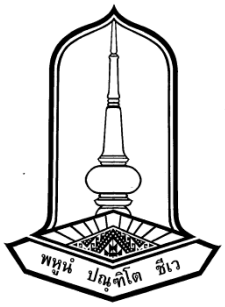


การสกัดข้อมูลการรักษาโรคข้าวจากเอกสารวิจัยที่ตีพิมพ์ใน PubMed

Extraction of Rice Disease Treatments from Published Research in PubMed

โครงงานปริญญานิพนธ์

ของ

นายจันปอง ฉั่น

นายดลสันติ์ สิงหาคำ

อนุญาตขึ้นสอบ Project 1

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2567

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

การสกัดข้อมูลการรักษาโรคข้าวจากเอกสารวิจัยที่ตีพิมพ์ใน PubMed

Extraction of Rice Disease Treatments from Published Research in PubMed

โครงงานปริญญานิพนธ์

ของ

นายจันปอง ฉั่น

นายดลสันติ์ สิงหาคำ

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ปีการศึกษา 2567

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบโครงงานปริญญานิพนธ์ ได้พิจารณาปริญญานิพนธ์ของ [ชื่อเจ้าของปริญญานิพนธ์] แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบโครงงานปริญญานิพนธ์

..... ประธานสอบ
(รองศาสตราจารย์ ดร. พนิดา ทรงรัมย์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สำรวน เวียงสมุทร)

..... ที่ปรึกษาโครงงานปริญญานิพนธ์หลัก
(รองศาสตราจารย์ ดร. จันทิมา พลพินิจ)

หลักสูตรวิทยาการคอมพิวเตอร์อนุมัติให้รับโครงงานปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

....
(อาจารย์พชระ พฤกษะศรี) (ผู้ช่วยศาสตราจารย์พิมลรัตน์ อ้วนศรีเมือง)
อาจารย์ผู้ประสานงานวิชาโครงงานปริญญานิพนธ์
วันที่ [ใช้วันที่สอบ] เดือน [ชื่อเดือน] พ.ศ. [ปี พ.ศ.]

บทคัดย่อ

ชื่อโครงงาน	การสกัดข้อมูลการรักษาโรคข้าวจากเอกสารวิจัยที่ตีพิมพ์ใน PubMed
ผู้จัดทำ	65011212003 นายจันปอง ฉั่น
	65011212011 นายดลสันติ์ สิงหาคำ
อาจารย์ที่ปรึกษา	รศ.ดร. จันทิมา พลพินิจ
หลักสูตร	วิทยาศาสตรบัณฑิต (สาขาวิชาวิทยาการคอมพิวเตอร์)
คณะ	วิทยาการสารสนเทศ
มหาวิทยาลัย	มหาวิทยาลัยมหาสารคาม
ปีที่พิมพ์	[ปีที่ส่งเล่มสมบูรณ์]

งานวิจัยนี้มุ่งเน้นการสกัดข้อมูลสำคัญเกี่ยวกับการรักษาโรคข้าวในประเทศไทย โดยใช้เทคนิคการเรียนรู้แบบไม่มีผู้สอนและการประมวลผลภาษาธรรมชาติ (NLP) เพื่อวิเคราะห์เอกสารวิจัยที่เผยแพร่ในฐานข้อมูล PubMed การดำเนินงานประกอบด้วยการสร้างตัวแทนเชิงตัวเลขของบทคัดย่อด้วยโมเดล SciBERT และการลดมิติข้อมูลด้วย UMAP จากนั้นนำข้อมูลที่ได้มาจัดกลุ่มด้วย k-means clustering โดยมีการประเมินผลด้วยดัชนีต่าง ๆ เช่น silhouette score, Calinski-Harabasz score และ Davies-Bouldin score นอกจากนี้ยังใช้เทคนิค Named Entity Recognition (NER) ในการสกัดชื่อโรค อาการ และวิธีการรักษาออกมาจากข้อมูล ผลการวิจัยแสดงให้เห็นว่าการผสมผสานระหว่างการประมวลผลภาษาธรรมชาติและการเรียนรู้แบบไม่มีผู้สอนสามารถสกัดข้อมูลที่เกี่ยวข้องกับโรคข้าวได้อย่างเป็นระบบและแม่นยำ ซึ่งจะเป็นฐานข้อมูลสำคัญในการพัฒนาวิธีการควบคุมและรักษาโรคข้าวในอนาคต

คำสำคัญ: การสกัดข้อมูล, โรคข้าว, การประมวลผลภาษาธรรมชาติ, คลัสเตอริ่ง, Named Entity Recognition (NER), การเรียนรู้แบบไม่มีผู้สอน

กิตติกรรมประกาศ

โครงงานปริญญานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ได้ด้วยความกรุณาและความช่วยเหลืออย่างสูงยิ่งจาก รองศาสตราจารย์ ดร. จันทิมา พลพินิจ อาจารย์ที่ปรึกษาโครงงานปริญญานิพนธ์เรื่อง “การสกัดข้อมูลการรักษาโรคพืชในประเทศไทยจากเอกสารวิจัยที่ตีพิมพ์ใน PubMed” และกรรมการควบคุมโครงงานปริญญานิพนธ์ [ชื่อประธานกรรมการสอบ] ประธานกรรมการสอบ และ [ชื่อกรรมการสอบ] กรรมการสอบ

ขอขอบพระคุณ [ชื่อผู้เชี่ยวชาญ] (ขอบคุณผู้เชี่ยวชาญที่ช่วยตรวจ หรือช่วยให้คำแนะนำ ถ้ามี)

ขอขอบพระคุณ (ขอบคุณผู้ให้การช่วยเหลือสนับสนุนอื่นๆ ถ้ามี)

จันปอง ฉั่น

ดลสันติ์ สิงหาคำ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ข้าวเป็นพืชเศรษฐกิจหลักที่มีความสำคัญอย่างยิ่งต่อประเทศไทยและภูมิภาคเอเชีย [1-3] โดยเฉพาะอย่างยิ่งในประเทศที่ประชากรส่วนใหญ่บริโภคข้าวเป็นอาหารหลัก ความสำคัญของข้าวไม่ได้มีเพียงด้านโภชนาการเท่านั้น แต่ยังส่งผลโดยตรงต่อเศรษฐกิจ ความมั่นคงทางอาหาร และวิถีชีวิตของเกษตรกรทั่วภูมิภาค ทว่าการผลิตข้าวมักประสบกับปัญหาจากโรคข้าวที่หลากหลาย ซึ่งเป็นอุปสรรคสำคัญต่อผลผลิตและคุณภาพของข้าว โรคข้าวหลากหลายชนิด [4-9] เช่น โรคไหม้คอรวง (Rice Blast) [10-11] โรคใบขีดสีน้ำตาล (Brown Spot) [6] โรคขอบใบแห้ง (Bacterial Leaf Blight) [12-13] โรคใบหงิก (Tungro Disease) [14-15] โรคข้าวแดง (Red Stripe Disease) [16] และโรคเมล็ดด่าง (Dirty Panicle) สามารถทำให้ต้นข้าวเกิดความเสียหายรุนแรง ส่งผลให้ผลผลิตลดลงและคุณภาพของข้าวเสื่อมถอย หากการแพร่ระบาดของโรคไม่ได้รับการควบคุมอย่างเหมาะสมและทันเวลา จะส่งผลกระทบต่อระบบนิเวศการเกษตรและลดความมั่นคงทางอาหารในประเทศที่ข้าวเป็นพืชอาหารหลัก ด้วยเหตุนี้ การมีวิธีการรักษาโรคข้าวที่มีประสิทธิภาพจึงเป็นสิ่งสำคัญอย่างยิ่งในการช่วยลดความสูญเสียที่เกิดจากโรคข้าว และเพิ่มศักยภาพในการผลิตข้าวให้มีคุณภาพสูงขึ้น

แหล่งข้อมูลที่เกี่ยวข้องกับการรักษาโรคข้าวและการวิจัยที่เป็นประโยชน์อย่างยิ่งต่อการพัฒนาวิธีการควบคุมโรคได้ถูกรวบรวมในฐานข้อมูลที่ครอบคลุมและน่าเชื่อถือเช่น PubMed ซึ่งเป็นแหล่งรวบรวมบทความวิจัยทางการแพทย์และชีววิทยาจากทั่วโลก [18] ข้อมูลจาก PubMed มีความน่าเชื่อถือเนื่องจากผ่านการตรวจสอบคุณภาพโดยนักวิจัยและผู้เชี่ยวชาญ ทำให้ฐานข้อมูลนี้เป็นแหล่งข้อมูลสำคัญสำหรับการศึกษาวิจัยที่เกี่ยวข้องกับการรักษาโรคข้าวในแง่มุมที่มีความหลากหลาย ตั้งแต่การวินิจฉัยอาการ ผลกระทบของโรค จนถึงการเลือกใช้วิธีการรักษาที่เหมาะสม อย่างไรก็ตาม ด้วยจำนวนบทความที่มีอยู่มากและเนื้อหาที่มีความซับซ้อน การค้นหาข้อมูลที่เกี่ยวข้องกับโรคข้าวอาจใช้เวลานานและต้องการการวิเคราะห์เชิงลึกเพื่อดึงข้อมูลที่สำคัญออกมาได้อย่างมีประสิทธิภาพ

งานวิจัยในปัจจุบันได้มุ่งเน้นการประยุกต์ใช้เทคโนโลยีปัญญาประดิษฐ์และการเรียนรู้ของเครื่องเพื่อยกระดับการตรวจจับและจำแนกโรคข้าวให้มีประสิทธิภาพมากยิ่งขึ้น โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ได้แสดงให้เห็นถึงศักยภาพในการช่วยระบุโรคข้าวอย่างแม่นยำและรวดเร็วในหลายภูมิภาค [19] โดยเทคนิคการประมวลผลภาพและอัลกอริธึมการเรียนรู้ของเครื่องถูกนำมาใช้เพื่อสกัดคุณลักษณะที่เกี่ยวข้องกับโรคและวิเคราะห์รูปแบบในภาพถ่ายต้นข้าว [20] ซึ่งวิธีการเหล่านี้มีข้อ

ได้เปรียบเหนือกว่าวิธีการสังเกตแบบดั้งเดิมที่ต้องอาศัยเวลาและอาจมีความคลาดเคลื่อนสูง [21] งานวิจัยหลายชิ้นพบว่าการใช้โมเดลการเรียนรู้เชิงลึกในการจำแนกประเภทของโรคให้ผลลัพธ์ที่แม่นยำกว่าวิธีการแบบเดิม [22] นอกจากนี้ ยังมีการศึกษาการบูรณาการความรู้เกี่ยวกับความต้านทานของพืชและความสามารถในการก่อโรคของเชื้อโรคเพื่อพัฒนาสายพันธุ์ข้าวที่มีความต้านทานต่อโรคมากยิ่งขึ้น [23] การพัฒนาเหล่านี้ในด้านการตรวจจับโรคข้าวมีเป้าหมายเพื่อเพิ่มผลิตภาพทางการเกษตรและสนับสนุนการเกษตรที่ยั่งยืน [24-25]

งานวิจัยนี้จึงมุ่งเน้นการใช้กระบวนการสกัดข้อมูลที่เกี่ยวข้องกับโรคข้าวโดยอัตโนมัติ โดยใช้กระบวนการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ร่วมกับเทคนิคด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) เพื่อนำข้อมูลที่สำคัญ ได้แก่ ชื่อโรค อาการของโรค และวิธีการรักษาโรคข้าว ออกมาจากเอกสารวิจัยที่ตีพิมพ์ใน PubMed อย่างมีประสิทธิภาพ วิธีการนี้จะช่วยประหยัดเวลาและแรงงานในการค้นหาและสรุปสาระสำคัญจากข้อมูลจำนวนมาก ตลอดจนสร้างฐานความรู้ที่มีคุณค่าในการพัฒนาวิธีการควบคุมและรักษาโรคข้าวที่เหมาะสมยิ่งขึ้นในอนาคต

1.2 วัตถุประสงค์ของโครงงาน

เพื่อสกัดข้อมูลที่สำคัญเกี่ยวกับการรักษาโรคข้าวในประเทศไทยจากเอกสารวิจัยที่ตีพิมพ์ในฐานข้อมูล PubMed โดยใช้เทคนิคการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ร่วมกับการประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) เพื่อดึงข้อมูลสำคัญ ได้แก่ ชื่อโรค อาการของโรค และวิธีการรักษาโรคข้าวออกมาได้อย่างมีประสิทธิภาพและเป็นระบบ

1.3 ขอบเขตของโครงงาน

1. นำเสนอกระบวนการวิจัยเพื่อสกัดข้อมูลที่สำคัญเกี่ยวกับการรักษาโรคข้าวในประเทศไทยจากเอกสารวิจัยที่ตีพิมพ์ในฐานข้อมูล PubMed โดยใช้เทคนิคการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ร่วมกับการประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) เพื่อดึงข้อมูลสำคัญ ได้แก่ ชื่อโรค อาการของโรค และวิธีการรักษาโรคข้าวออกมาได้อย่างมีประสิทธิภาพและเป็นระบบ
2. โรคข้าวที่ศึกษามี 6 โรคคือ โรคไหม้คอรวง (Rice Blast) โรคใบขีดสีน้ำตาล (Brown Spot) โรคขอบใบแห้ง (Bacterial Leaf Blight) โรคใบหงิก (Tungro Disease) โรคข้าวแดง (Red Stripe Disease) และโรคเมล็ดด่าง (Dirty Panicle)
3. ใช้ข้อมูลงานวิจัยเกี่ยวกับข้าวจาก PubMed

4. มีผู้เชี่ยวชาญในสาขาเกษตรหรือโรคพืชมาช่วยตรวจสอบและทำ Ground Truth ให้กับข้อมูลที่สกัดได้ เพื่อให้มั่นใจว่าข้อมูลที่ได้มีความถูกต้อง และสามารถใช้งานได้ในบริบทการเกษตร

5. ใช้ Entity Recognition (ER) ในการดึงข้อมูลที่เกี่ยวข้องกับชื่อโรค อาการ และวิธีการรักษา ซึ่งช่วยระบุข้อมูลที่เกี่ยวข้องกับโรคข้าวในเอกสารวิจัย

6. ใช้ SciBERT Embedding ในการทำ Text Representation

7. ใช้เทคนิค Clustering ในการจัดกลุ่มหัวข้อที่เกี่ยวข้องกับโรคข้าวในเอกสารวิจัย เพื่อค้นหาและจัดหมวดหมู่ข้อมูลเกี่ยวกับการรักษาโรคข้าว

8. ประเมินประสิทธิภาพในการจัดกลุ่มเอกสารด้วยค่า Silhouette Score, Calinski-Harabasz Index และ Davies-Bouldin Index

1.3 ความสำคัญของโครงงาน

1. การสนับสนุนเกษตรกรและเพิ่มประสิทธิภาพการผลิตข้าว: งานวิจัยนี้จะช่วยให้เกษตรกรสามารถเข้าถึงข้อมูลเกี่ยวกับโรคข้าว อาการ และวิธีการรักษาที่เหมาะสมได้อย่างง่ายดายและรวดเร็ว ซึ่งมีความสำคัญอย่างยิ่งต่อการควบคุมการแพร่ระบาดของโรคในช่วงเวลาที่เหมาะสม ช่วยลดความสูญเสียและเพิ่มประสิทธิภาพในการผลิตข้าวให้ได้คุณภาพสูงขึ้น

2. การเกษตรที่ยั่งยืนและการรักษาความมั่นคงทางอาหาร: การพัฒนาวิธีการรักษาและควบคุมโรคข้าวแบบอัตโนมัตินี้มีบทบาทสำคัญในการสนับสนุนการเกษตรที่ยั่งยืน ข้อมูลที่แม่นยำและทันสมัยจากงานวิจัยที่ตีพิมพ์ใน PubMed ช่วยให้เกษตรกรและนักวิจัยสามารถพัฒนาแนวทางที่ปลอดภัยและคุ้มค่าในการจัดการโรคพืช ซึ่งส่งผลดีต่อความมั่นคงทางอาหารในประเทศไทยและภูมิภาคเอเชีย

3. การพัฒนาการสกัดข้อมูลอัตโนมัติจากเอกสารวิจัยจำนวนมาก: งานวิจัยนี้ใช้เทคโนโลยีปัญญาประดิษฐ์และการประมวลผลภาษาธรรมชาติ (NLP) ร่วมกับการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ซึ่งจะพัฒนาแนวทางในการสกัดข้อมูลสำคัญจากเอกสารจำนวนมากที่ตีพิมพ์ในฐานข้อมูลขนาดใหญ่ เช่น PubMed ช่วยลดระยะเวลาและความซับซ้อนในการค้นคว้าข้อมูล ช่วยให้ได้ข้อมูลที่แม่นยำและมีประสิทธิภาพสูงขึ้น

1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

1.5.1 ฮาร์ดแวร์

- Processor 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz
- Installed RAM 32.0 GB
- System type 64-bit operating system, x64-based processor

1.5.2 ซอฟต์แวร์

- Visual Studio Code
- XAMPP
- SpaCy (NLP)

1.5.3 เครื่องมือสำหรับการค้นหาและรวบรวมข้อมูล

- PubMed API

1.6 แผนการดำเนินงาน

โครงงานปริญญานิพนธ์ฉบับนี้ ดำเนินงาน ณ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคามระหว่างเดือน xxxxx 25xx ถึง xxxxx 25xx

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	เดือน											
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาและรวบรวมข้อมูล												
2. วิเคราะห์และกำหนดขอบเขต												
3. ออกแบบระบบ												
4. พัฒนาโปรแกรม												
5. ทดสอบระบบ												
6. ทำรายงานสรุป												
7. นำเสนอโครงงาน												

บทที่ 2
ทฤษฎีและระบบงานที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 PubMed และ PubMed API

PubMed [26-27] คือฐานข้อมูลฟรีของบทความวิชาการด้านชีวการแพทย์ ชีววิทยา ชีวสารสนเทศ และสาขาที่เกี่ยวข้อง จัดทำโดย National Center for Biotechnology Information (NCBI) ภายใต้ National Library of Medicine (NLM) ของสหรัฐอเมริกา PubMed เป็นที่นิยมสำหรับการสืบค้นบทความวิจัย วารสาร และข้อมูลที่ผ่านการทบทวนโดยผู้เชี่ยวชาญ ซึ่งครอบคลุมเนื้อหาจากสาขาการแพทย์ ชีววิทยา ชีวเวชศาสตร์ เภสัชศาสตร์ และสาธารณสุขวิทยา

PubMed มีบทความวิจัยและข้อมูลทางวิชาการเกี่ยวกับโรคข้าว รวมถึงที่มาของโรคและวิธีการรักษาที่เกี่ยวข้อง โดยครอบคลุมหัวข้อต่าง ๆ เช่น:

โรคข้าว (Rice Diseases) – รวมถึงโรคที่พบในข้าว เช่น โรคไหม้ข้าว (rice blast), โรคขอบใบแห้ง (bacterial blight) และโรคอื่นๆ ที่เกิดจากเชื้อรา แบคทีเรีย หรือไวรัส

สาเหตุของโรคข้าว (Etiology of Rice Diseases) – สาเหตุการเกิดโรค ซึ่งครอบคลุมถึงเชื้อก่อโรค ปัจจัยแวดล้อมที่กระตุ้นการเกิดโรค เช่น ความชื้น อุณหภูมิ และการจัดการแปลงข้าว

การจัดการและการรักษาโรคข้าว (Treatment and Management of Rice Diseases) – วิธีการจัดการโรคข้าว รวมถึงการใช้สารเคมี การจัดการศัตรูพืชอย่างบูรณาการ (Integrated Pest Management: IPM) การพัฒนาสายพันธุ์ข้าวที่ทนทานต่อโรค และการปรับปรุงวิธีการเกษตร

PubMed API (หรือเรียกอย่างเป็นทางการว่า NCBI E-utilities หรือ Entrez Programming Utilities) เป็นอินเทอร์เฟซที่ช่วยให้นักพัฒนาหรือผู้ใช้สามารถเข้าถึงข้อมูลในฐานข้อมูล PubMed ได้โดยอัตโนมัติผ่านโปรแกรมหรือสคริปต์ API นี้สามารถใช้ในการค้นหา ดึงข้อมูลบทความ (เช่น ชื่อเรื่อง ผู้แต่ง บทคัดย่อ) และเรียกข้อมูลในรูปแบบที่สะดวกสำหรับการประมวลผลเชิงวิเคราะห์ในโครงการวิจัยหรือระบบที่ต้องการข้อมูลทางการแพทย์ โดยสามารถค้นหาบทความที่เกี่ยวข้องใน PubMed โดยใช้ Keywords เช่น “rice diseases,” “rice blast treatment,” “bacterial blight in rice,” หรือ “management of rice diseases”

การใช้ PubMed API (Entrez Programming Utilities) เพื่อดึงข้อมูลจาก PubMed ด้วย Keywords เช่น “rice diseases,” “rice blast treatment,” “bacterial blight in rice,” หรือ “management of rice diseases” สามารถทำได้โดยใช้คำสั่ง esearch เพื่อค้นหา PubMed IDs

(PMIDs) และคำสั่ง efetch เพื่อดึงข้อมูลบทคัดย่อหรือข้อมูลอื่นๆ ของเอกสารนั้นมาได้ ตัวอย่างการใช้ API สามารถแสดงโปรแกรมตัวอย่างได้ดังภาพที่ 2.1

```
import requests
from xml.etree import ElementTree as ET

# คีย์เวิร์ดสำหรับค้นหาบทความ
query = "rice diseases OR rice blast treatment OR bacterial blight in rice OR
management of rice diseases"

# ใช้คำสั่ง esearch เพื่อค้นหา PMIDs ของบทความที่เกี่ยวข้อง
esearch_url =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term={query
}&retmax=5&retmode=xml"
response = requests.get(esearch_url)
root = ET.fromstring(response.content)

# ดึง PMIDs จากผลลัพธ์การค้นหา
pmid_list = [id_elem.text for id_elem in root.findall(".//Id")]
print("PMIDs ที่ค้นพบ:", pmid_list)

# ใช้คำสั่ง efetch เพื่อดึงข้อมูลของแต่ละบทความ
for pmid in pmid_list:
    efetch_url =
f"https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed&id={pmid}&re
tmode=xml"
    response = requests.get(efetch_url)
    root = ET.fromstring(response.content)

    # ดึงข้อมูลจาก XML เช่น ชื่อเรื่อง และบทคัดย่อ
    title = root.find(".//ArticleTitle").text
    abstract = root.find(".//AbstractText").text if root.find(".//AbstractText")
is not None else "No abstract available"

    print("\n--- บทความ ---")
    print("PMID:", pmid)
    print("ชื่อเรื่อง:", title)
    print("บทคัดย่อ:", abstract)
```

ภาพที่ 2.1 ตัวอย่างโปรแกรมการดึงข้อมูลด้วย Python และ PubMed API

สามารถอธิบายตัวอย่างโปรแกรมได้ดังนี้

1. ขั้นตอนการค้นหา (esearch):
   - กำหนด URL ของ esearch โดยใช้คีย์เวิร์ดที่ต้องการ เช่น “rice diseases OR rice blast treatment OR bacterial blight in rice OR management of rice diseases”
   - กำหนดให้ retmax = 5 เพื่อจำกัดจำนวนบทความที่ค้นหา (ปรับเพิ่มหรือลดได้ตามต้องการ)

- ผลลัพธ์ XML ที่ได้จะถูกแปลงเป็นโครงสร้างข้อมูล XML เพื่อดึง PMID ของแต่ละบทความ

2. ขั้นตอนการดึงข้อมูลบทความ (efetch):
   - ใช้ PMID แต่ละตัวในการดึงข้อมูลบทความผ่าน efetch โดยกำหนด retmode=xml เพื่อให้ผลลัพธ์เป็น XML
   - ดึงข้อมูลที่ต้องการ เช่น ชื่อเรื่อง (ArticleTitle) และบทคัดย่อ (AbstractText)

จากโปรแกรมข้างต้น การค้นหาใน PubMed ด้วยคีย์เวิร์ด เช่น “*rice diseases*,” “*rice blast treatment*,” “*bacterial blight in rice*,” หรือ “*management of rice diseases*” จะสืบค้นข้อมูลบทความตัวอย่างมาในรูปแบบที่ได้ดังภาพที่ 2.2

--- บทความ ---
PMID: 12345678
ชื่อเรื่อง: Efficacy of Novel Fungicides in Managing Rice Blast Disease in Southeast Asia
บทคัดย่อ: This study investigates the efficacy of several novel fungicides in controlling rice blast disease, a major threat to rice production. Field trials were conducted in multiple locations to evaluate fungicidal effects on disease severity and yield improvement. Results indicate significant reduction in rice blast symptoms with specific fungicides, highlighting potential for integrated disease management approaches.

--- บทความ ---
PMID: 23456789
ชื่อเรื่อง: Bacterial Blight in Rice: Pathogenicity and Management Strategies
บทคัดย่อ: Bacterial blight, caused by Xanthomonas oryzae pv. oryzae, presents a substantial challenge for rice cultivation globally. This review summarizes current knowledge on pathogenicity, environmental factors affecting disease prevalence, and recent advancements in management practices, including resistant cultivar development and biological control agents.

--- บทความ ---
PMID: 34567890
ชื่อเรื่อง: Impact of Climate Change on Rice Disease Incidence and Management Practices
บทคัดย่อ: Climate change significantly influences the incidence and severity of rice diseases. This article discusses how rising temperatures and altered precipitation patterns affect pathogen spread and crop susceptibility. Adaptation strategies, such as modifying planting schedules and breeding for climate-resilient varieties, are proposed to mitigate adverse effects on rice production.

--- บทความ ---
PMID: 45678901
ชื่อเรื่อง: The Role of Integrated Pest Management in Combating Rice Sheath Blight
บทคัดย่อ: Rice sheath blight is a severe disease affecting rice yields in many Asian countries. This study explores the use of integrated pest management (IPM) to reduce disease impact through a combination of cultural, biological, and chemical controls. Findings suggest IPM provides a sustainable approach for reducing reliance on chemical treatments.

ภาพที่ 2.2 ตัวอย่างเอกสารที่สืบค้นได้

2.1.2 การทำความสะอาดเอกสารข้อความ (Text Cleaning)

การทำความสะอาดเอกสารข้อความ (Text Cleaning) [28-30] เป็นกระบวนการที่เตรียมข้อความดิบ (Raw Text) ให้อยู่ในรูปแบบที่เหมาะสมสำหรับการวิเคราะห์ข้อมูลหรือนำไปใช้งานด้าน NLP (Natural Language Processing) เนื่องจากข้อมูลดิบมักมีอักขระที่ไม่จำเป็น ข้อมูลที่ซ้ำซ้อน หรือการจัดรูปแบบที่ไม่เหมาะสม การทำความสะอาดช่วยลดปัญหาเหล่านี้ ทำให้การวิเคราะห์ข้อมูลแม่นยำยิ่งขึ้น โดยทั่วไปขั้นตอนหลักในการทำความสะอาดข้อความจะมีดังนี้

(1) การลบอักขระพิเศษ (Removing Special Characters) คือ การลบอักขระที่ไม่จำเป็น เช่น @, #, &, *, (), % เป็นต้น เพื่อให้เหลือเฉพาะข้อความที่ต้องการ ตัวอย่างสามารถแสดงได้ดังนี้

ข้อความเดิม: “Hello @everyone! Meet me at 5pm :)”

ผลลัพธ์หลังลบอักขระพิเศษ: “Hello everyone Meet me at 5pm”

(2) การลบช่องว่างที่เกินมา (Removing Extra Whitespace) คือ การลบช่องว่างซ้ำซ้อนที่อาจมีมากเกินหนึ่งช่อง รวมถึงการเว้นวรรคตอนต้นและตอนท้ายของข้อความ ตัวอย่างสามารถแสดงได้ดังนี้

ข้อความเดิม: “ Machine learning is powerful . ”

ผลลัพธ์หลังลบช่องว่างเกิน: “Machine learning is powerful.”

(3) การแปลงข้อความให้เป็นตัวพิมพ์เล็ก (Lowercasing) คือ การทำให้ข้อความทั้งหมดเป็นตัวพิมพ์เล็ก เพื่อป้องกันการนับคำซ้ำจากการใช้ตัวพิมพ์ใหญ่และเล็ก ตัวอย่างสามารถแสดงได้ดังนี้

ข้อความเดิม: “Machine Learning is Interesting”

ผลลัพธ์: “machine learning is interesting”

(4) การลบเครื่องหมายวรรคตอน (Removing Punctuation) คือ การลบเครื่องหมายวรรคตอน เช่น ,, ., ;, :, !, ? เพื่อให้เหลือเฉพาะคำที่สำคัญ ตัวอย่างสามารถแสดงได้ดังนี้

ข้อความเดิม: “Data science, machine learning, and AI are related fields.”

ผลลัพธ์: “Data science machine learning and AI are related fields”

2.1.3 การเตรียมเอกสารข้อความ (Text Preparation)

เป็นกระบวนการที่จัดเตรียมข้อความให้พร้อมสำหรับการวิเคราะห์ข้อมูลหรือการทำงานด้าน NLP (Natural Language Processing) [28-30] โดยเน้นการปรับปรุงและแปลงข้อความดิบให้อยู่ในรูปแบบที่สามารถประมวลผลได้ง่ายและแม่นยำมากขึ้น รวมถึงการแปลงโครงสร้าง และการเลือกคุณลักษณะต่างๆ ของข้อความ ขั้นตอนการเตรียมเอกสารข้อความ เช่น

(1) การทำ Tokenization คือ การแยกข้อความเป็นคำ หรือประโยคเล็ก ๆ ทำให้ง่ายต่อการนับคำหรือวิเคราะห์โครงสร้างข้อความ เช่น

ข้อความต้นฉบับ: “Natural Language Processing is fun”

ผลลัพธ์: [“Natural”, “Language”, “Processing”, “is”, “fun”]

(2) การลบคำที่ไม่สำคัญ (Removing Stop Words) คือ ลบคำที่พบได้บ่อยแต่ไม่มีความหมายสำคัญ เช่น and, or, but, is, the, a เป็นต้น

ข้อความต้นฉบับ: “Machine learning is a branch of artificial intelligence.”

ผลลัพธ์: “Machine learning branch artificial intelligence”

(3) การแปลงคำให้เป็นรูปฐาน (Lemmatization หรือ Stemming) คือ การลดคำให้เป็นรูปฐานหรือรากเดิม เพื่อให้ง่ายต่อการประมวลผล เช่น เปลี่ยน running เป็น run

ข้อความต้นฉบับ: “She was running and he runs”

ผลลัพธ์หลัง Lemmatization: “She be run and he run”

(4) การแปลงเป็นเวคเตอร์ (Vectorization) และการให้น้ำหนักคำ (Term Weighting) คือ การแปลงข้อความให้อยู่ในรูปของตัวเลขที่ใช้ประมวลผลได้ง่าย เช่น การใช้ Bag of Words, TF-IDF หรือ Word Embedding เช่น Word2Vec

2.1.4 คลัสเตอริ่ง (Clustering)

Clustering [31] คือเทคนิคการวิเคราะห์ข้อมูลที่จัดกลุ่มข้อมูลให้เป็นกลุ่ม (clusters) โดยที่ข้อมูลในแต่ละกลุ่มมีความคล้ายคลึงกันมากกว่าข้อมูลในกลุ่มอื่น ๆ การทำ Clustering เป็นส่วนหนึ่งของ การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เพราะไม่ได้มีคำตอบที่กำหนดไว้ล่วงหน้า แต่จะค้นหาความสัมพันธ์หรือรูปแบบที่ซ่อนอยู่ในข้อมูล ประโยชน์ของ Clustering ได้แก่

1. ช่วยในการค้นหารูปแบบที่ซ่อนอยู่ในข้อมูล
2. ทำให้การแยกประเภทข้อมูลง่ายขึ้นโดยไม่ต้องใช้ป้ายกำกับ
3. ใช้ในการวิเคราะห์ลูกค้าหรือผู้ใช้เพื่อแบ่งกลุ่มตลาด (Market Segmentation)
4. ใช้ในการจัดการและค้นหาข้อมูล เช่น การค้นหาเอกสารที่คล้ายคลึงกันในระบบข้อมูลขนาดใหญ่

วิธีการ Clustering ที่นิยมใช้ ได้แก่

1. K-Means Clustering [32]**:** แบ่งข้อมูลเป็นกลุ่มตามจำนวนกลุ่ม (K) ที่กำหนดล่วงหน้า โดยใช้ตำแหน่งของค่าเฉลี่ย (centroid) เป็นศูนย์กลางของแต่ละกลุ่ม ข้อมูลแต่ละจุดจะถูกจัดให้อยู่ในกลุ่มที่มีศูนย์กลางที่ใกล้ที่สุด ขั้นตอนการทำงานของ K-Means Clustering คือ

(1) กำหนดจำนวนกลุ่ม (K): กำหนดค่า K หรือจำนวนกลุ่มที่ต้องการแบ่งข้อมูล เช่น K=3 สำหรับแบ่งข้อมูลออกเป็น 3 กลุ่ม

(2) สุ่มตำแหน่งจุดศูนย์กลางเริ่มต้น: เริ่มต้นด้วยการสุ่มเลือกจุดศูนย์กลาง (centroids) ของแต่ละกลุ่มตามจำนวน K

(3) จัดกลุ่มข้อมูลตามระยะห่าง: สำหรับแต่ละจุดข้อมูล คำนวณระยะห่างระหว่างจุดข้อมูลกับจุดศูนย์กลางของแต่ละกลุ่ม จากนั้นจัดข้อมูลให้อยู่ในกลุ่มที่มีศูนย์กลางใกล้ที่สุด (โดยทั่วไปใช้การคำนวณระยะห่างแบบ Euclidean Distance)

ในการจัดกลุ่ม x_i (จุดข้อมูล) ให้ใกล้เคียงกับจุดสูนย์กลางของแต่ละกลุ่ม μ_j สมการระยะทางแบบ Euclidean Distance

$$d(x_i, \mu_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - \mu_{jk})^2} \tag{2.1}$$

โดยที่ x_i คือเวกเตอร์ของจุดข้อมูล μ_j คือเวกเตอร์ของจุดสูนย์กลางของกลุ่มที่ *j* และ *n* คือจำนวนคุณลักษณะ (Features) ของข้อมูล

(4) อัปเดตตำแหน่งจุดศูนย์กลางใหม่ (Update Centroids): เมื่อได้ข้อมูลที่อยู่ในแต่ละกลุ่มแล้ว จะคำนวณจุดศูนย์กลางใหม่ของแต่ละกลุ่ม โดยคำนวณจากค่าเฉลี่ยของจุดข้อมูลทั้งหมดในกลุ่มนั้น

เมื่อจัดข้อมูลทั้งหมดในแต่ละกลุ่มเสร็จแล้ว จุดศูนย์กลางใหม่ของกลุ่ม μ_j จะคำนวณจากค่าเฉลี่ยของจุดข้อมูลทั้งหมดในกลุ่ม C_j ดังนี้

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \tag{2.2}$$

โดยที่ μ_j คือจุดศูนย์กลางใหม่ของกลุ่ม *j* ในขณะที่ C_j คือกลุ่มของจุดข้อมูลที่อยู่ในกลุ่ม *j* และ $|C_j|$ คือจำนวนจุดข้อมูลในกลุ่ม C_j

สมการเป้าหมาย (Objective Function) ของ *k*-means clustering คือการลด Sum of Squared Errors (SSE) หรือการลดระยะห่างรวมระหว่างจุดข้อมูลแต่ละจุดกับจุดศูนย์กลางของกลุ่มนั้น โดยสมการเป้าหมายเป็นดังนี้:

$$SSE = \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \tag{2.3}$$

โดยที่ K คือจำนวนกลุ่ม ในขณะที่ x_i คือจุดข้อมูลที่อยู่ในกลุ่ม C_j และ C_j คือจุดศูนย์กลางของกลุ่ม j ซึ่ง k-means clustering จะพยายามหาจุดสูนย์กลางที่ทำให้ค่า SSE ค่าต่ำที่สุด ซึ่งแสดงถึงการจัดกบุ่มที่เหมาะสม

(5) ทำซ้ำกระบวนการ (Iteration): ทำการจัดกลุ่มใหม่ตามจุดศูนย์กลางที่คำนวณใหม่ในขั้นตอนที่ 4 ทำซ้ำขั้นตอน 3 และ 4 จนกว่าจุดศูนย์กลางจะไม่เปลี่ยนแปลง หรือจำนวนรอบที่กำหนดไว้ครบ

(6) หยุดการทำงานและสรุปผล: เมื่อจุดศูนย์กลางหยุดเปลี่ยนแปลง จะได้กลุ่มที่แบ่งไว้อย่างเหมาะสมตามค่า K ที่กำหนดไว้

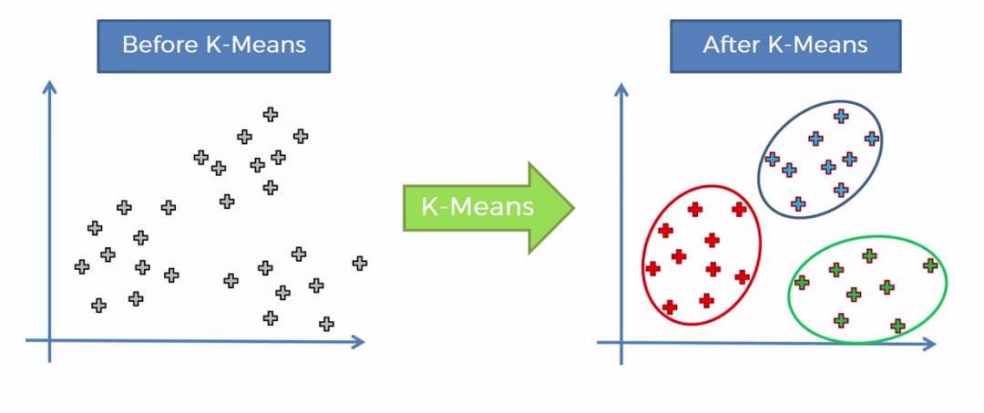
ข้อดีของ K-Means Clustering คือ

- เข้าใจง่ายและคำนวณรวดเร็ว: อัลกอริทึมนี้มีโครงสร้างที่ไม่ซับซ้อนและทำงานรวดเร็ว โดยเฉพาะเมื่อต้องการแบ่งข้อมูลขนาดใหญ่
- การจัดกลุ่มที่ชัดเจน: ข้อมูลจะถูกจัดให้อยู่ในกลุ่มที่มีความคล้ายคลึงกันสูง ทำให้กลุ่มที่ได้มีความแตกต่างกันอย่างชัดเจน

ข้อเสียของ K-Means Clustering คือ

- ต้องกำหนดจำนวนกลุ่มล่วงหน้า: ต้องกำหนดค่า K ล่วงหน้า ซึ่งอาจไม่ทราบจำนวนกลุ่มที่เหมาะสมได้ทันที
- ไวต่อค่าเริ่มต้นของ Centroids: การสุ่มจุดเริ่มต้นอาจทำให้ผลลัพธ์แตกต่างกันไป ค่าที่ได้อาจจะติดอยู่ที่ค่าที่ไม่ใช่ค่าที่ดีที่สุด (local optimum)
- เหมาะกับข้อมูลที่มีรูปทรงกลม: ไม่เหมาะกับข้อมูลที่มีโครงสร้างซับซ้อนหรือมีรูปทรงที่ไม่ใช่วงกลม

การเลือกจำนวนกลุ่ม (K) ที่เหมาะสม - หนึ่งในวิธีการเลือก K คือ Elbow Method โดยการวาดกราฟระหว่างค่า K กับค่า Sum of Squared Errors (SSE) ซึ่งคำนวณจากผลรวมของระยะห่างระหว่างจุดข้อมูลกับจุดศูนย์กลางของกลุ่มที่อยู่ เมื่อค่า K เพิ่มขึ้น ค่า SSE จะลดลง แต่จะมีจุดที่การลดลงเริ่มไม่มากหรือเรียกว่าจุดข้อศอก (Elbow) ซึ่งสามารถเลือกเป็นค่า K ที่เหมาะสมได้



ภาพที่ 2.3 ตัวอย่างการทำ K-Means Clustering[51]

2. Hierarchical Clustering [33]**:** เป็นเทคนิคการจัดกลุ่มข้อมูลที่สร้างลำดับชั้นของกลุ่มข้อมูลโดยการรวมกลุ่มเข้าด้วยกันทีละขั้นตอน เทคนิคนี้สามารถแบ่งออกเป็น 2 วิธีหลัก คือ:

(1) Agglomerative (Bottom-Up): เริ่มจากการจัดให้แต่ละจุดข้อมูลเป็นกลุ่มของตนเอง จากนั้นรวมกลุ่มที่ใกล้เคียงกันจนได้กลุ่มเดียว

(2) Divisive (Top-Down): เริ่มจากกลุ่มเดียวที่มีจุดข้อมูลทั้งหมด จากนั้นแยกกลุ่มออกทีละกลุ่มจนเหลือกลุ่มย่อย ๆ ของแต่ละจุดข้อมูล

โดยทั่วไปจะนิยมแบบ Agglomerative Hierarchical Clustering เพราะว่า

- กระบวนการคำนวณที่ง่ายกว่าและมีประสิทธิภาพมากกว่า: Agglomerative Clustering เริ่มจากการรวมจุดข้อมูลทีละคู่ ซึ่งทำให้คำนวณได้ง่ายและค่อนข้างตรงไปตรงมาในเชิงโครงสร้าง ขณะที่ Divisive Clustering เริ่มจากการแยกกลุ่มใหญ่สุด ซึ่งต้องประเมินทุกกลุ่มย่อยและค้นหาการแยกที่ดีที่สุดในทุกขั้นตอน ทำให้การคำนวณซับซ้อนและใช้เวลามากกว่า
- มีอัลกอริทึมที่ปรับใช้ได้หลากหลาย: Agglomerative Clustering รองรับวิธีการเชื่อมโยง (Linkage Criteria) หลากหลาย เช่น Single Linkage, Complete Linkage, Average Linkage และ Centroid Linkage ซึ่งทำให้สามารถปรับแต่งการรวมกลุ่มได้หลากหลายและเหมาะสมกับข้อมูลประเภทต่าง ๆ ขณะที่ Divisive Clustering มักมีความยืดหยุ่นน้อยกว่า
- ทำงานได้ดีกับข้อมูลขนาดเล็กถึงขนาดกลาง: Agglomerative Clustering ทำงานได้ดีและประสิทธิภาพสูงเมื่อใช้กับข้อมูลขนาดเล็กถึงขนาดกลาง เนื่องจากการรวมกลุ่มทำได้อย่างรวดเร็วและตรงจุด ขณะที่ Divisive Clustering มีข้อจำกัดในเรื่องการประมวลผลเมื่อข้อมูลมีขนาดใหญ่
- เข้าใจง่ายและสามารถแสดงผลผ่าน Dendrogram ได้ชัดเจน: กระบวนการรวมกลุ่มจากจุดข้อมูลเดียวไปสู่กลุ่มใหญ่ช่วยให้เข้าใจโครงสร้างการรวมกลุ่มได้ง่าย และแสดงลำดับขั้น

การรวมกลุ่มด้วย Dendrogram ที่ช่วยวิเคราะห์ความสัมพันธ์ระหว่างกลุ่มได้ชัดเจน ขณะที่ Divisive Clustering มีขั้นตอนการแยกกลุ่มซับซ้อนกว่า ทำให้การแสดงผลและการตีความยุ่งยากขึ้น

- การใช้งานในงานวิจัยและโปรแกรมส่วนใหญ่: Agglomerative Clustering เป็นที่นิยมในงานวิจัยและโปรแกรมต่าง ๆ มากกว่า Divisive เนื่องจากมีการใช้งานอย่างกว้างขวางในซอฟต์แวร์ด้านการทำเหมืองข้อมูลและการเรียนรู้ของเครื่อง เช่น ใน Python libraries อย่าง Scikit-learn

ขั้นตอนของ Agglomerative Hierarchical Clustering ประกอบด้วยขั้นตอนหลักดังนี้:

ขั้นตอนที่ 1: คำนวณระยะห่างระหว่างจุดข้อมูล - คำนวณ ระยะห่าง ระหว่างจุดข้อมูลแต่ละจุด เช่น Euclidean Distance

ขั้นตอนที่ 2: รวมกลุ่มที่ใกล้เคียงกันที่สุด – มีขั้นตอนดังนี้

(1) เริ่มต้นโดยถือว่าจุดข้อมูลแต่ละจุดเป็นกลุ่มของตัวเอง

(2) ค้นหาคู่ของกลุ่มที่มีระยะห่างน้อยที่สุด (ใกล้เคียงกันที่สุด) และรวมกลุ่มนั้นเข้าด้วยกัน

(3) คำนวณระยะห่างระหว่างกลุ่มใหม่กับกลุ่มอื่นๆ ตามวิธีการเชื่อมโยงที่เลือก (Linkage Criteria)

ขั้นตอนที่ 3: การเลือกวิธีการเชื่อมโยง (Linkage Criteria) - ในการรวมกลุ่ม เราจะเลือกวิธีการเชื่อมโยง (Linkage Criteria) ที่เหมาะสมเพื่อคำนวณระยะห่างระหว่างกลุ่มใหม่กับกลุ่มอื่นๆ ตัวเลือกที่นิยมใช้มีดังนี้:

(1) Single Linkage (ระยะใกล้ที่สุด): ใช้ระยะห่างที่ใกล้ที่สุดระหว่างสองกลุ่ม

$$d(A,B) = \min\{d(x_i, x_j) : x_i \in A, x_j \in B\} \quad (2.4)$$

(2) Complete Linkage (ระยะไกลที่สุด): ใช้ระยะห่างที่ไกลที่สุดระหว่างสองกลุ่ม

$$d(A,B) = \max\{d(x_i, x_j) : x_i \in A, x_j \in B\} \quad (2.5)$$

(3) Average Linkage (ค่าเฉลี่ยระยะห่าง): ใช้ค่าเฉลี่ยของระยะห่างทั้งหมดระหว่างกลุ่ม

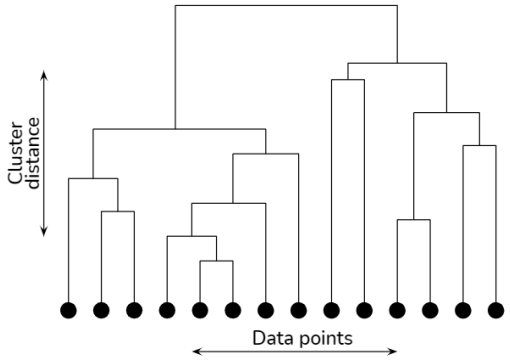
$$d(A,B) = \frac{1}{|A||B|}\sum_{x_i \in A}\sum_{x_j \in B} d(x_i, x_j) \quad (2.6)$$

(4) Centroid Linkage: ใช้ระยะห่างระหว่างจุดศูนย์กลางของแต่ละกลุ่ม

$$d(A,B) = d(centroid_A, centroid_B) \quad (2.7)$$

ขั้นตอนที่ 4: ทำซ้ำการรวมกลุ่ม - ทำการรวมกลุ่มโดยอ้างอิงจาก Linkage Criteria ที่เลือกและคำนวณระยะห่างของกลุ่มที่เกิดขึ้นใหม่กับกลุ่มอื่นๆ ทำซ้ำขั้นตอนนี้จนกระทั่งเหลือกลุ่มเดียวที่มีข้อมูลทั้งหมดอยู่ภายในกลุ่มเดียวกัน

ขั้นตอนที่ 5: สร้าง Dendrogram - แสดงผลลัพธ์ในรูปของ Dendrogram ซึ่งเป็นกราฟที่แสดงลำดับขั้นของการรวมกลุ่ม โดยแกนนอนแสดงระยะห่างระหว่างกลุ่มที่รวมกันและแกนตั้งแสดงลำดับของกลุ่มที่รวมกัน การเลือกจำนวนกลุ่มสามารถทำได้โดยการกำหนดระดับความสูงของ Dendrogram ที่ต้องการ



ภาพที่ 2.4 ตัวอย่าง Dendrogram ของ Agglomerative Hierarchical Clustering[52]

2.1.5 การลดมิติด้วย Uniform Manifold Approximation and Projection (UMAP)

UMAP (Uniform Manifold Approximation and Projection)[50] เป็นเทคนิคการลดมิติที่ช่วยเปลี่ยนข้อมูลจากมิติสูงให้กลายเป็นมิติที่ต่ำกว่า (เช่น 2 หรือ 3 มิติ) เพื่อให้สามารถวิเคราะห์และแสดงผลได้ง่ายขึ้น เทคนิคนี้ถูกออกแบบมาเพื่อรักษาโครงสร้างของข้อมูลในเชิงท้องถิ่นและเชิงโลกในเวกเตอร์ที่ได้จากข้อมูลดิบ

หลักการของ UMAP มีดังนี้

- *การสร้างกราฟความสัมพันธ์ในมิติสูง* - UMAP จะประเมินระยะห่างและความสัมพันธ์ระหว่างตัวอย่างในข้อมูลมิติสูง จากนั้นสร้างกราฟที่แสดงถึงความใกล้ชิดของข้อมูลแต่ละจุด
- *การลดมิติข้อมูล* - โดยการแก้ปัญหาการปรับแต่งแบบ non-linear UMAP จะหาการแสดงผลในมิติที่ต่ำกว่า (เช่น 2D หรือ 3D) ที่ยังคงรักษาความสัมพันธ์ที่สำคัญของข้อมูลเดิมไว้

ขั้นตอนการทำ Dimensionality Reduction ด้วย UMAP

1) เตรียมข้อมูล (Data Preprocessing) - ความสะอาดและปรับปรุงข้อมูล เช่น การลบ noise, การทำ normalization และการแปลงข้อมูลให้อยู่ในรูปแบบเวกเตอร์Frequency หรือ TF-IDF เพื่อเก็บข้อมูลความถี่ของคำในเอกสารเพื่อให้ข้อมูลพร้อมสำหรับการลดมิติ

2) กำหนดพารามิเตอร์ของ UMAP - ระบุจำนวนมิติเป้าหมายที่ต้องการ (เช่น 2 หรือ 3 มิติ) พร้อมทั้งกำหนดพารามิเตอร์อื่น ๆ เช่น จำนวนเพื่อนบ้าน (n_neighbors) และระยะเวลาของการลดมิติ ซึ่งจะมีผลต่อความแม่นยำและการรักษาโครงสร้างของข้อมูล

3) คำนวณระยะห่างและสร้างกราฟ - คำนวณระยะห่างระหว่างข้อมูลในมิติสูง จากนั้นสร้างกราฟความสัมพันธ์ที่แสดงถึงความใกล้ชิดของตัวอย่างแต่ละจุด โดยใช้ metric ที่เหมาะสมกับลักษณะข้อมูล

4) ทำการ Embed ข้อมูลไปสู่มิติที่ต่ำกว่า - ใช้อัลกอริทึม non-linear optimization ในการปรับตำแหน่งของข้อมูลในมิติที่ต่ำกว่า โดยพยายามรักษาความสัมพันธ์ที่ได้จากกราฟความสัมพันธ์ในมิติสูงไว้ให้มากที่สุด

5) ทำซ้ำจนกว่าการแก้ปัญหาจะเสถียร (Iterate Until Convergence) - ทำการปรับปรุงตำแหน่งของข้อมูลซ้ำ ๆ จนกระทั่งผลลัพธ์มีความเสถียรและสามารถแสดงความสัมพันธ์ระหว่างข้อมูลได้อย่างชัดเจน

6) สรุปผลและแสดง Visualization - เมื่อขั้นตอนการ Embed เสร็จสิ้น ผลลัพธ์จะได้เป็นข้อมูลในรูปแบบที่ลดมิติลงแล้ว ซึ่งสามารถแสดงในรูปแบบ scatter plot หรือ visualization อื่น ๆ เพื่อให้เห็นการจัดกลุ่มและความสัมพันธ์ระหว่างข้อมูล

สมมติมีตัวอย่างเอกสารดังนี้

เอกสารที่ 1: This study investigates the efficacy of several novel fungicides in controlling rice blast disease, a major threat to rice production.

เอกสารที่ 2: Results indicate significant reduction in rice blast symptoms with specific fungicides, highlighting potential for integrated disease management approaches.

เอกสารที่ 3: Climate change significantly influences the incidence and severity of rice diseases.

หากกำหนดหัวข้อ K = 3 นั่นคือ ชื่อโรค สาเหตุการเกิดโรค และการรักษาโรค สำหรับ การลดมิติ ด้วย UMAP สามารถแสดงผลลัพธ์ที่คาดว่าจะได้ดังภาพที่ 2.5 ซึ่งเป็นผลจากการรันโปรแกรมในภาพที่ 2.6

dimension_0	dimension_1	dimension_2	dimension_3	dimension_4	dimension_5	dimension_6	dimension_7	dimension_8	dimension_9	dimension_10	dimension_11	dimension_12	dimension_13	dimension_14	dimension_15	dimension_16	dimension_17	dimension_18
9.405964	7.3637185	3.2264907	5.643235	4.3507147	7.3168287	4.0665946	4.273409	3.000444	5.1854396	6.826557	5.145085	7.8055716	6.226971	2.5165417	4.5289273	4.6733184	5.166415	5.004296
9.080899	7.1438656	3.9826615	5.023241	4.4505944	7.451331	3.9683511	4.29877	3.1768656	5.354739	6.5948	4.9623713	7.262846	6.1355395	2.4328463	4.7307987	4.4738345	4.8071065	5.047484
8.336597	7.11361	4.2393003	5.2566495	4.075198	7.6366043	4.2116218	4.2521534	2.6665084	5.5103846	6.986348	5.1608815	7.5234814	6.327841	2.3507907	4.690779	4.9608116	4.939143	5.2228165
10.043323	7.4987717	3.1617303	6.017354	4.1836214	7.356149	4.332286	4.616478	3.0917084	4.976917	7.0176806	5.2026267	8.195692	6.188904	2.3656561	4.276123	4.7806535	5.33242	4.9591208
9.482809	7.698451	3.401543	5.840382	3.8956933	7.3581486	4.222017	4.5274515	2.89791	5.0603027	7.2658505	5.3340535	7.7192035	6.2556233	2.5026727	4.3900113	4.911432	5.008219	5.284412
9.187725	6.9952354	3.9400558	5.2109523	4.5028267	7.261978	4.0863557	4.49491	3.2712333	5.1917515	6.5289054	5.2074	7.4141192	6.2619805	2.461421	4.731381	4.6681156	4.8971305	5.014025
8.286167	7.206642	3.8953052	5.0599933	4.15115	7.402894	3.949344	4.1296053	2.6784093	5.437983	6.9756265	5.23666	7.213394	6.345042	2.601706	4.8249784	4.853925	4.817982	5.293578
9.0461	7.8024726	3.081741	5.5378385	3.9829829	7.5247316	3.900748	3.9649453	2.5460699	5.362276	7.3187203	4.970482	7.6517625	6.130308	2.592366	4.4958887	4.619343	5.09572	5.2291775
8.275705	7.203695	3.9280336	5.0463815	4.14242	7.446102	3.9690068	4.133101	2.6602952	5.457457	6.98736	5.216882	7.2358	6.3429904	2.5840254	4.814036	4.8658104	4.829011	5.282495
9.194921	7.8353953	3.3144639	5.7055964	3.8197193	7.37003	4.054015	4.3135023	2.7231183	5.1535134	7.388913	5.287071	7.521327	6.242285	2.6173365	4.4614744	4.8562336	4.9214244	5.3903775
8.697885	7.1289763	4.422393	4.9930673	4.356829	7.768627	4.086963	4.207114	3.003327	5.567876	6.6341386	4.86489	7.343337	6.1649537	2.3368304	4.701227	4.6175075	4.8213468	5.0491834
8.232848	7.050864	4.2527947	5.057609	4.2064285	7.574342	4.0725517	4.158321	2.729485	5.5420794	6.8435745	5.130488	7.32493	6.3217607	2.4391634	4.80066	4.857923	4.851944	5.2089443
8.350451	7.2747374	4.1148705	5.195137	4.038636	7.6589212	4.116187	4.1648226	2.614541	5.526456	7.064309	5.11773	7.430939	6.3025956	2.437343	4.6883407	4.9090185	4.8937125	5.2583165
8.211141	7.042515	4.214402	5.2251463	4.12341	7.5589366	4.149792	4.184275	2.658479	5.525029	6.9287558	5.180779	7.4596205	6.3389244	2.3802464	4.7590256	4.929543	4.922148	5.239004
8.554836	7.785372	2.916539	5.833232	3.8419452	7.297727	3.8935103	3.8496766	2.3187404	5.3336053	7.4612937	5.2912493	7.6809797	6.3179865	2.7160296	4.572738	4.888329	5.127083	5.411183
9.962099	7.493857	3.551193	5.8052087	4.12701	7.4052873	4.339049	4.7392764	3.233189	4.994064	6.967746	5.2086673	7.855333	6.1695094	2.3415992	4.3263106	4.7687206	5.0674114	5.0683217
8.582029	7.7951837	2.9817653	5.780415	3.8289778	7.32688	3.900411	3.8819344	2.3382363	5.3356466	7.468605	5.2724266	7.64093	6.304458	2.7052844	4.568723	4.878425	5.0932565	5.4163656
9.999736	7.347495	3.4486744	5.765692	4.343663	7.476545	4.3063593	4.5915256	3.2388482	5.0905924	6.784726	5.0341277	8.050783	6.123026	2.285292	4.3353643	4.631342	5.235938	4.8721075
9.989087	7.6088386	3.4565306	5.961259	3.998601	7.365443	4.384344	4.7933273	3.1690714	4.926874	7.1121197	5.3301034	7.902941	6.220679	2.3850029	4.27528	4.8959846	5.076701	5.146735
10.137804	7.3070545	3.3059444	5.977101	4.3145156	7.135836	4.3596387	4.8359985	3.3764079	4.8317676	6.7867846	5.4084105	8.010017	6.2740927	2.4096718	4.371126	4.838585	5.188422	4.9758463
9.333945	7.6561155	2.9437888	5.542079	4.18315	7.393891	3.8874543	4.042635	2.7495573	5.2656355	7.1087036	4.9673657	7.713557	6.107103	2.5843396	4.5091624	4.512615	5.156044	5.1026816
10.2077875	7.2320676	3.7316904	5.668414	4.471402	7.501358	4.381926	4.781248	3.538779	5.051864	6.550308	5.0317135	7.9316773	6.1154227	2.220867	4.3348036	4.5960007	5.10482	4.808741
8.650787	7.6916385	3.1615605	5.7572455	3.9106297	7.448252	4.002458	3.9483793	2.4098055	5.3773685	7.3544965	5.21607	7.75595	6.3073864	2.6163635	4.535157	4.904107	5.1430154	5.3078604
9.426904	7.284551	3.4217105	5.572074	4.3257318	7.306943	4.112887	4.3929706	3.053517	5.152939	6.8215027	5.157273	7.754601	6.2093134	2.4602294	4.5447326	4.6772814	5.122412	5.023466
8.404235	6.433951	4.161878	4.161596	5.0741715	6.6908407	3.4579902	4.2830033	3.6110597	5.2562423	5.9084697	5.4464507	6.2831063	6.4357777	2.8459182	5.4529085	4.376864	4.33513	5.063238
9.323929	7.743467	2.8647954	5.486627	4.145783	7.4132953	3.8269799	3.9825578	2.7009668	5.2876143	7.1715035	4.9164104	7.6508007	6.073446	2.614656	4.5113754	4.460655	5.127801	5.132525
9.025129	7.806875	3.3148918	5.6544423	3.831019	7.4404683	4.007174	4.1590714	2.6056159	5.2638617	7.398786	5.162355	7.5606475	6.195805	2.57325	4.4865456	4.7769513	4.975908	5.3694134
9.429836	7.262906	3.8439453	5.204209	4.3872323	7.557993	4.0645256	4.358689	3.1860836	5.3220744	6.6842885	4.851632	7.494907	6.047947	2.3268342	4.570449	4.422442	4.926996	4.987079
9.437794	7.327288	3.8290284	5.257535	4.3312383	7.595325	4.0793447	4.344807	3.1503859	5.333844	6.7432566	4.837027	7.5213223	6.037254	2.3121917	4.53632	4.430018	4.9305487	5.010127
8.4514475	6.44304	4.1794314	4.141182	5.0484056	6.700209	3.4610617	4.285964	3.6382267	5.2645416	5.9384575	5.4597826	6.280349	6.4943233	2.8268924	5.444396	4.4071364	4.3110533	5.0429277
8.457445	6.4800544	4.1321545	4.2053494	5.042188	6.7164288	3.4946072	4.252503	3.6086025	5.2435894	5.923685	5.430281	6.3217716	6.447522	2.819896	5.398005	4.3811107	4.3351784	5.0960956
8.447013	6.4505715	4.0364723	4.1756597	5.104557	6.632134	3.4449968	4.208573	3.6456141	5.221621	5.8314347	5.5054636	6.2589736	6.4806023	2.8656108	5.4336777	4.355099	4.329711	5.1158624
8.805049	7.411023	4.016066	5.4598036	3.9797304	7.65258	4.2163253	4.3128295	2.74264	5.402077	7.1098084	5.111327	7.6002765	6.242236	2.3631346	4.538344	4.8791814	4.950375	5.236565
9.960215	7.314934	3.6020305	5.6823716	4.2873335	7.3240924	4.306945	4.7611976	3.3429132	4.9797177	6.7886987	5.2133894	7.804049	6.1808796	2.3496656	4.406789	4.7176485	5.062913	4.995525
10.052406	7.2471685	3.3145895	5.9309697	4.3624835	7.136862	4.3311934	4.7780046	3.3535292	4.8701215	6.7426147	5.378291	8.010312	6.2723236	2.408147	4.4064684	4.8112807	5.207998	4.9513884
8.476519	6.461968	4.0672035	4.201256	5.041352	6.6538873	3.4503121	4.2435465	3.6519372	5.2580047	5.9126515	5.486174	6.2906303	6.5316043	2.8107376	5.430439	4.4180403	4.3494616	5.0690613

ภาพที่ 2.5 ตัวอย่างผลลัพธ์การลดมิติด้วย UMAP

```
def reduce_dimensions(X, method='umap', n_components=50):
    """
    Reduces dimensionality using specified method (UMAP or LDA).
    """
    if method == 'umap':
        reducer = umap.UMAP(
            n_components=n_components,
            n_neighbors=30,
            min_dist=0.1,
            random_state=CONFIG['random_state']
        )
        return reducer.fit_transform(X)
    elif method == 'lda':
        lda = LatentDirichletAllocation(
            n_components=n_components,
            random_state=CONFIG['random_state']
        )
        return lda.fit_transform(X)
```

ภาพที่ 2.6 ตัวอย่างโปรแกรมการลดมิติด้วย UMAP

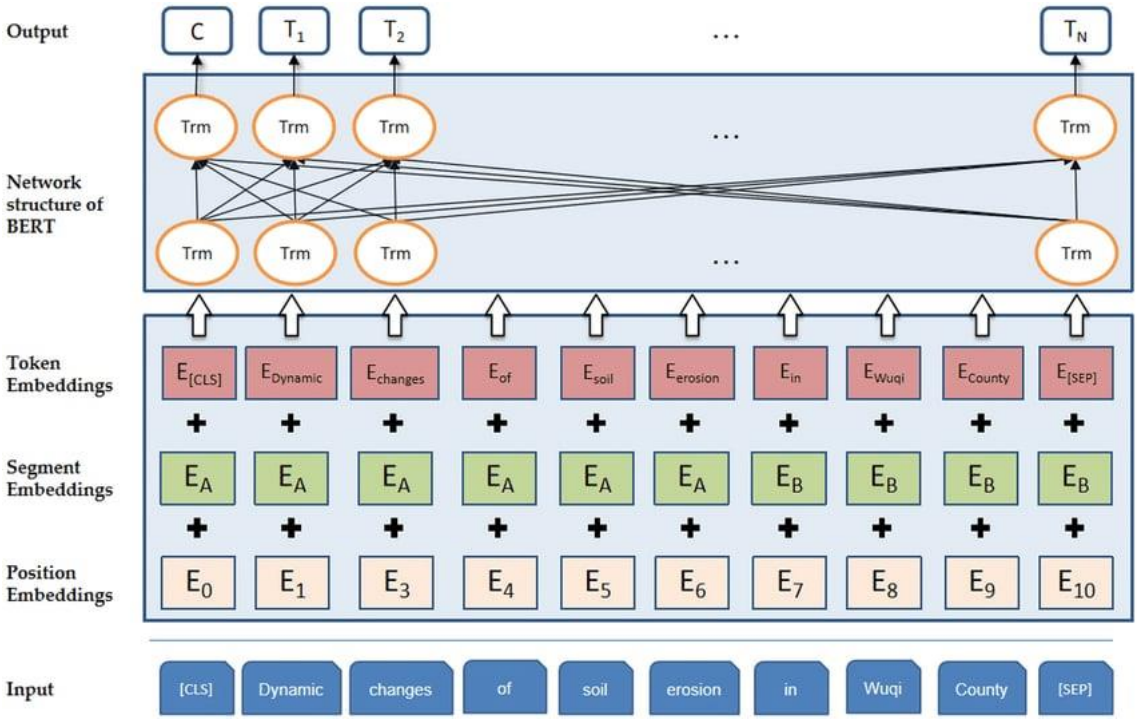
2.1.6 SciBERT โมเดลภาษาสำหรับงานวิจัยทางวิทยาศาสตร์

SciBERT (Scientific BERT)[49] เป็นโมเดลภาษาปัญญาประดิษฐ์ที่พัฒนาโดย Allen Institute for AI เพื่อใช้ในการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) สำหรับข้อมูลที่เกี่ยวข้องกับงานวิจัยทางวิทยาศาสตร์โดยเฉพาะ โดยอ้างอิงจากสถาปัตยกรรม BERT (Bidirectional Encoder Representations from Transformers) ซึ่งเป็นโมเดลพื้นฐานสำหรับการทำความเข้าใจบริบทของคำในข้อความ

1. คุณสมบัติของ SciBERT

SciBERT ได้รับการฝึกด้วยชุดข้อมูลทางวิทยาศาสตร์ที่ประกอบไปด้วยเอกสารจาก Semantic Scholar ซึ่งเป็นฐานข้อมูลขนาดใหญ่ของบทความวิชาการ ชุดข้อมูลนี้มีความสมดุลระหว่างสาขาวิชาวิทยาศาสตร์ชีวภาพและวิศวกรรมศาสตร์ ทำให้ SciBERT มีประสิทธิภาพดีในงานที่เกี่ยวข้องกับข้อความทางวิทยาศาสตร์

- Corpus: SciBERT ถูกฝึกด้วยชุดข้อมูลขนาด 1.14 ล้านบทความ (ประมาณ 3.17 พันล้านคำ)
- Vocabulary: ใช้ WordPiece tokenizer ที่ถูกสร้างขึ้นใหม่จากข้อมูลทางวิทยาศาสตร์ ไม่ใช้คำศัพท์ของ BERT ดั้งเดิม (BERT-Base)



ภาพที่ 2.7 BERT Model[53]

- Model Architecture: ใช้โครงสร้างเดียวกับ BERT-Base คือ 12 layers, 768 hidden units, 12 attention heads

2. การประยุกต์ใช้ SciBERT - SciBERT ถูกนำไปใช้ในหลายงานด้าน NLP ที่เกี่ยวข้องกับเอกสารทางวิทยาศาสตร์ เช่น

- Named Entity Recognition (NER): การระบุชื่อเฉพาะ เช่น ชื่อของสารเคมี ชื่อโรค หรือชื่อโมเลกุล
- Relation Extraction: การดึงความสัมพันธ์ระหว่างหน่วยข้อมูล เช่น การเชื่อมโยงระหว่างยากับโรค
- Text Classification: การจำแนกประเภทของบทความทางวิทยาศาสตร์ เช่น การจัดหมวดหมู่งานวิจัย
- Question Answering (QA): การตอบคำถามที่เกี่ยวข้องกับเนื้อหาทางวิทยาศาสตร์

3. ข้อดีและข้อเสียของ SciBERT

ข้อดี SciBERT

- ได้รับการฝึกจากข้อมูลทางวิทยาศาสตร์ ทำให้เข้าใจศัพท์เฉพาะและโครงสร้างของบทความทางวิชาการได้ดีกว่าโมเดลทั่วไป
- มีประสิทธิภาพสูงกว่ารุ่น BERT-Base ในงานที่เกี่ยวข้องกับข้อมูลทางวิทยาศาสตร์
- สามารถนำไปใช้ได้กับงานประมวลผลภาษาหลายประเภท เช่น NER, QA และ Text Classification
- ใช้งานง่ายผ่าน Hugging Face Transformers ทำให้สามารถนำไปประยุกต์ใช้ได้อย่างสะดวก

ข้อเสีย SciBERT

- โมเดลมีขนาดใหญ่และต้องการทรัพยากรคอมพิวเตอร์สูงในการประมวลผล
- มีข้อจำกัดในความสามารถในการทำงานกับข้อมูลที่ไม่ใช่เชิงวิทยาศาสตร์ เช่น ภาษาในบทสนทนาทั่วไป
- ยังต้องการการปรับแต่งเพิ่มเติม (fine-tuning) เพื่อให้เหมาะสมกับงานเฉพาะด้าน

4. ข้อจำกัดและความท้าทายของ SciBERT

1) ความต้องการทรัพยากรสูง - การฝึกและใช้งาน SciBERT ต้องใช้ GPU หรือ TPU ที่มีประสิทธิภาพสูง ทำให้การประมวลผลมีค่าใช้จ่ายสูง
2) ความสามารถในการประยุกต์ใช้กับข้อมูลที่ไม่ใช่วิทยาศาสตร์ - SciBERT อาจไม่สามารถทำงานได้ดีในบริบทที่ไม่ใช่วิทยาศาสตร์ เนื่องจากชุดข้อมูลที่ใช้ฝึกมาจากบทความวิชาการ
3) ความซับซ้อนของโมเดล - SciBERT มีโครงสร้างที่ซับซ้อน ทำให้ต้องใช้ความเชี่ยวชาญในการปรับแต่งและนำไปใช้งานอย่างเหมาะสม

5. สมการที่ใช้ใน SciBERT - SciBERT ใช้ฟังก์ชันการฝึกแบบเดียวกับ BERT โดยใช้ Masked Language Model (MLM) และ Next Sentence Prediction (NSP) ซึ่งสามารถแสดงได้ด้วยสมการต่อไปนี้:

1. Loss function สำหรับ Masked Language Model (MLM)

$$L_{MLM} = -\sum_{i \in M} logP(x_i \mid x_{-i}) \tag{2.8}$$

โดยที่ เป็นเซ็ตของตำแหน่งที่ถูก mask และ คือ token ที่แท้จริง

ตัวอย่าง

ข้อความต้นฉบับ "The mitochondrion is the powerhouse of the cell."
ข้อความที่ถูก mask "The mitochondrion is the [MASK] of the cell."

2. Loss function สำหรับ Next Sentence Prediction (NSP)

$$L_{NSP} = -[ylogP(y \mid A, B) + (1 - y)log(1 - P(y \mid A, B))] \tag{2.9}$$

โดยที่ เป็นค่าที่บ่งบอกว่าประโยค เป็นประโยคถัดไปของ หรือไม่
ตัวอย่าง

- ประโยค A "DNA contains genetic information."
- ประโยค B (True) "It is found in the nucleus of most cells."
- ประโยค B (False) "Photosynthesis occurs in chloroplasts."

SciBERT จะเรียนรู้ว่า B (True) เป็นประโยคที่เกี่ยวข้องกับ A มากกว่า B (False)

6. เครื่องมือและไลบรารีที่ใช้ร่วมกับ SciBERT

SciBERT สามารถใช้งานผ่าน Hugging Face Transformers ซึ่งเป็นไลบรารียอดนิยมสำหรับการประมวลผลภาษาธรรมชาติ โดยสามารถโหลดโมเดลและใช้งานได้ง่ายผ่าน Python

```
from transformers import AutoTokenizer, AutoModel

tokenizer =
AutoTokenizer.from_pretrained("allenai/scibert_scivocab_uncased")
model = AutoModel.from_pretrained("allenai/scibert_scivocab_uncased")

text = "COVID-19 is caused by the SARS-CoV-2 virus."
inputs = tokenizer(text, return_tensors="pt")
outputs = model(**inputs)
```

ภาพที่ 2.8 ตัวอย่างโปรแกรมการใช้ไลบรารีที่ใช้ร่วมกับ SciBERT

2.1.7 เทคนิคที่ใช้ในการประเมิน

การประเมินผลสำหรับ Clustering ที่เป็นการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) สามารถประเมินด้วยเทคนิคต่อไปนี้:

- Silhouette Score [38]: ใช้วัดว่าข้อมูลแต่ละจุดใกล้เคียงกับกลุ่มของตัวเองมากแค่ไหนเมื่อเทียบกับกลุ่มอื่น ค่าอยู่ในช่วง -1 ถึง 1 ค่าใกล้ 1 หมายถึงการจัดกลุ่มที่ชัดเจน
  - ค่าใกล้ 1: แสดงว่าข้อมูลอยู่ใกล้กับจุดในกลุ่มเดียวกันและอยู่ห่างจากกลุ่มอื่น ซึ่งหมายถึงการจัดกลุ่มที่ดี
  - ค่าใกล้ 0: แสดงว่าข้อมูลอยู่ใกล้กับเขตแดนระหว่างกลุ่ม
  - ค่าใกล้ -1: แสดงว่าข้อมูลอาจถูกจัดในกลุ่มที่ผิดพลาด เนื่องจากข้อมูลใกล้กับกลุ่มอื่นมากกว่า

การคำนวณ Silhouette Score สำหรับแต่ละจุดข้อมูล *i*

(1) ระยะเฉลี่ยภายในกลุ่ม (*a*): คือคำนวณระยะทางเฉลี่ยระหว่างจุดข้อมูล *i* กับจุดข้อมูลอื่นๆ ภายในกลุ่มของมันเอง เรียกว่า *a*(*i*)

(2) ระยะเฉลี่ยภายนอกกลุ่ม (*b*): คือคำนวณระยะทางเฉลี่ยระหว่างจุดข้อมูล *i* กับจุดข้อมูลในกลุ่มที่ใกล้เคียงที่สุดที่มันไม่ได้เป็นสมาชิก เรียกว่า *b*(*i*)

(3) Silhouette Score ของจุดข้อมูล *i*:

$$s(i) = \frac{b(i) - a(i)}{max\big((a(i), b(i))\big)} \tag{2.10}$$

โดย $s(i)$ คือ Silhouette Score ของจุด i ในขณะที่ $a(i)$ คือระยะเฉลี่ยภายในกลุ่ม และ $b(i)$ คือระยะเฉลี่ยภายนอกกลุ่ม (กลุ่มที่ใกล้ที่สุดที่ข้อมูล i ไม่ได้เป็นสมาชิก)

- Calinski-Harabasz Index[47]: ใช้วัดคุณภาพของการจัดกลุ่มข้อมูลโดยพิจารณาจากความแตกต่างระหว่างกลุ่มและความสอดคล้องภายในกลุ่ม
  - ค่า Index ที่สูง: แสดงว่าความแตกต่างระหว่างกลุ่มมากและข้อมูลภายในแต่ละกลุ่มมีความสอดคล้องดี ซึ่งหมายถึงการจัดกลุ่มที่ชัดเจน
  - ค่า Index ที่ต่ำ: บ่งบอกว่าอาจมีการจัดกลุ่มที่ไม่ชัดเจน เนื่องจากความแตกต่างระหว่างกลุ่มน้อยหรือข้อมูลภายในกลุ่มมีความแปรปรวนสูง

การคำนวณ Calinski-Harabasz Index

(1) การกระจายภายในกลุ่ม (SSW: Within-Cluster Dispersion) คำนวณผลรวมของระยะทางกำลังสองระหว่างจุดข้อมูลแต่ละจุดกับจุดศูนย์กลาง (centroid) ของกลุ่มที่มันอยู่

$$SS_W = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (2.11)$$

โดยที่ k คือจำนวนกลุ่ม C_i คือกลุ่มที่ i และ μ_i คือจุดศูนย์กลางของกลุ่ม i

(2) การกระจายระหว่างกลุ่ม (SSB: Between-Cluster Dispersion) คำนวณผลรวมของระยะทางกำลังสองระหว่างจุดศูนย์กลางของแต่ละกลุ่ม (μ_i) กับจุดศูนย์กลางรวมของข้อมูลทั้งหมด (μ) คูณด้วยจำนวนจุดข้อมูลในกลุ่มนั้น (n_i)

$$SS_B = \sum_{i=1}^{k} n_i \|\mu_i - \mu\|^2 \quad (2.12)$$

(3) คำนวณความแปรปรวนระหว่างกลุ่ม (Between-cluster dispersion, B)

B คือผลรวมของความแปรปรวนระหว่างค่าเฉลี่ยของแต่ละกลุ่มกับค่าเฉลี่ยรวมของข้อมูล โดยคูณด้วยจำนวนข้อมูลในแต่ละกลุ่ม

$$CH = \frac{SS_B}{SS_W} \times \frac{N - K}{K - 1} \quad (2.13)$$

โดยที่ k คือจำนวนกลุ่ม n คือจำนวนข้อมูลทั้งหมด และ ค่า CH ที่สูงขึ้นหมายถึงการจัดกลุ่มที่มีความชัดเจนและมีประสิทธิภาพ เนื่องจากความแตกต่างระหว่างกลุ่มสูงและความแปรปรวนภายในกลุ่มต่ำ

- Davies-Bouldin Index[48]: ใช้วัดคุณภาพของการจัดกลุ่มข้อมูลโดยพิจารณาความกระจุกของข้อมูลภายในแต่ละกลุ่มและความแตกต่างระหว่างกลุ่ม
  - ค่า DB ที่ต่ำ: แสดงว่ากลุ่มข้อมูลมีความกระจุกแน่นและแยกจากกันอย่างชัดเจน ซึ่งหมายถึงการจัดกลุ่มที่มีประสิทธิภาพ
  - ค่า DB ที่สูง: บ่งบอกว่ามีการกระจุกตัวที่กว้างหรือกลุ่มข้อมูลซ้อนทับกันมาก จึงแสดงถึงการจัดกลุ่มที่ไม่เหมาะสม

การคำนวณ Davies-Bouldin Index

(1) สำหรับแต่ละกลุ่ม i คำนวณค่า S_i ซึ่งเป็นค่าเฉลี่ยของระยะห่างระหว่างจุดข้อมูลในกลุ่ม i กับจุดศูนย์กลางของกลุ่มนั้น

(2) สำหรับคู่ของกลุ่ม i และ $j(i \neq j)$ คำนวณระยะห่างระหว่างจุดศูนย์กลางของทั้งสองกลุ่ม โดยเรียกว่า M_{ij}

(3) คำนวณอัตราส่วนระหว่างความกระจุกภายในและระยะห่างระหว่างกลุ่มสำหรับแต่ละคู่ โดยใช้สูตร

$$Rij = \frac{S_i + S_j}{M_{ij}} \tag{2.14}$$

(4) สำหรับแต่ละกลุ่ม i หาค่าสูงสุดของ R_{ij} เมื่อเปรียบเทียบกับกลุ่ม j ทั้งหมด (ที่ $j \neq i$) เรียกว่า R_i

(5) Davies-Bouldin Index คือค่าเฉลี่ยของ R_i ทั้งหมด เมื่อมี k กลุ่ม

$$DB = \frac{1}{k}\sum_{i=1}^{k} R_i \tag{2.15}$$

โดยค่า DB ที่ต่ำหมายความว่ากลุ่มข้อมูลมีความกระจุกแน่นและแยกออกจากกันได้ดี ซึ่งเป็นสัญญาณของการจัดกลุ่มที่มีประสิทธิภาพ

2.1.8 การทำ Entity Extraction ด้วย spaCy PhaseMatcher

PhraseMatcher ใน SpaCy เป็นเครื่องมือที่ใช้สำหรับการจับคู่ข้อความ (Text Matching) โดยใช้ลำดับของคำ (phrases) ที่เรากำหนดไว้ล่วงหน้า ซึ่งเหมาะสำหรับการค้นหาประโยคหรือคำสำคัญที่มีโครงสร้างเฉพาะในเอกสาร เช่น ชื่อเฉพาะทาง การแพทย์ ชื่อโรค หรือคำศัพท์เฉพาะทางด้านอื่น ๆ

PhraseMatcher ทำงานได้อย่างมีประสิทธิภาพสูง เพราะมันทำการแปลงข้อความที่ต้องการจับคู่ (patterns) ให้อยู่ในรูป Doc objects จากนั้นใช้ข้อมูลเชิงโครงสร้าง (tokens และโครงสร้างข้อความ) ที่ SpaCy สร้างไว้ มันจึงสามารถค้นหารูปแบบข้อความที่ซับซ้อนในเอกสารขนาดใหญ่ได้อย่างรวดเร็ว

1. หลักการทำงานของ PhraseMatcher

การเตรียม Pattern PhraseMatcher จะเริ่มต้นด้วยการแปลง “วลีเป้าหมาย (Patterns)” เป็นลำดับของคุณสมบัติ (Attributes) ของ Token เช่น อาจใช้ LOWER (ตัวพิมพ์เล็ก) ORTH (รูปแบบดั้งเดิมของคำ) หรือ LEMMA (รากศัพท์) เป็นต้น

ตัวอย่าง: ถ้าต้องการตรวจจับคำว่า *"โรคไหม้"* และ *"เพลี้ยกระโดดสีน้ำตาล"* จะถูกแปลงให้เป็น Pattern ที่สามารถเปรียบเทียบกับ Token ในเอกสารได้

- การสร้างแฮช (Hash Table) spaCy จะสร้างตารางแฮชสำหรับเก็บค่า Attributes ของ Token ตัวแรกในแต่ละ Pattern เพื่อให้สามารถตรวจจับคำที่ตรงกันในเอกสารได้เร็วขึ้น

- การสแกนเอกสาร เมื่อเอกสารที่มีลำดับของ Token เช่น D = (t_1, t_2, t_3, ..., t_n) ถูกประมวลผล อัลกอริทึมจะทำการสแกนเอกสารทีละตำแหน่ง *i* โดยเริ่มตรวจสอบว่า Token t_i ตรงกับ Token แรกของ Pattern *p* หรือไม่
- การตรวจจับการจับคู่ (Matching) หาก Token t_i ตรงกับ Token แรกของ Pattern *p* (ที่มีลำดับ Token คือ $p = (p_1, p_2, ..., p_k)$), spaCy จะตรวจสอบการจับคู่ของ Token ถัดไปในเอกสารกับ Token ถัดไปใน Pattern โดยใช้ฟังก์ชัน $\delta(t, p)$ ดังนี้
  - $\delta(t, p) = 1$ หากคุณสมบัติของ Token *t* ตรงกับ Pattern *p*
  - $\delta(t, p) = 0$ หากไม่ตรงกัน

การจับคู่จะถือว่าสมบูรณ์เมื่อผลคูณของฟังก์ชัน δ ของทุก Token ใน Pattern มีค่าเป็น 1 ดังสมการ

$$\prod_{j=0}^{k-1} \delta\left(t_{i+j}, p_{j+1}\right) = 1 \qquad (2.16)$$

หากผลลัพธ์จากสมการนี้เป็น 1 หมายความว่าพบการจับคู่ Pattern ในเอกสาร ณ ตำแหน่ง i

2. หลักการทำงานของ PhraseMatcher

ตัวอย่างนี้ เราจะจับคู่คำศัพท์เกี่ยวกับโรคที่เกี่ยวข้องกับข้าว เช่น “Rice blast”, “Bacterial leaf blight” หรือ “Sheath blight” จากข้อความ

```
import spacy
from spacy.matcher import PhraseMatcher

nlp = spacy.load("en_core_web_sm")
text = """
Rice blast is a major disease affecting rice crops worldwide.
Another common disease is Bacterial leaf blight, which causes significant
yield losses.
Farmers should also be cautious about Sheath blight, which can spread
rapidly under warm conditions.
"""
matcher = PhraseMatcher(nlp.vocab)
diseases = ["Rice blast", "Bacterial leaf blight", "Sheath blight"]
patterns = [nlp.make_doc(disease) for disease in diseases]
matcher.add("RiceDiseases", patterns)
doc = nlp(text)
matches = matcher(doc)
for match_id, start, end in matches:
string_id = nlp.vocab.strings[match_id]
span = doc[start:end]  # ข้อความที่จับคู่
print(f"Matched: {span.text} (Label: {string_id})")
```

ภาพที่ 2.9 ตัวอย่างโปรแกรมการใช้งาน PhraseMatcher

นั่นคือผลลัพธ์ที่คาดว่าจะได้มีดังนี้

Matched: Rice blast (Label: RiceDiseases)

Matched: Bacterial leaf blight (Label: RiceDiseases)

Matched: Sheath blight (Label: RiceDiseases)

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยล่าสุดมุ่งเน้นไปที่การพัฒนาวิธีการขั้นสูงสำหรับการตรวจจับและจัดการโรคในข้าว โดยอาศัยเทคโนโลยีปัญญาประดิษฐ์ (AI) และการประมวลผลภาพเพื่อให้กระบวนการวินิจฉัยและควบคุมโรคข้าวมีความแม่นยำ รวดเร็ว และมีประสิทธิภาพมากขึ้น การใช้เทคโนโลยีเหล่านี้ได้รับการยอมรับในวงกว้างว่ามีศักยภาพในการแก้ไขปัญหาที่สำคัญในภาคเกษตรกรรม โดยเฉพาะอย่างยิ่งในประเทศที่การปลูกข้าวเป็นแหล่งรายได้หลักและเป็นแหล่งอาหารสำคัญของประชากร

เทคนิคที่ใช้ในการพัฒนานี้รวมถึง เครือข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNN) ซึ่งมีประสิทธิภาพในการวิเคราะห์และจำแนกภาพของโรคข้าวได้อย่างแม่นยำ CNN แสดงศักยภาพในการระบุโรคข้าวที่แตกต่างกันตามภูมิภาคและลักษณะของโรคในรูปแบบต่าง ๆ ซึ่งเป็นสิ่งสำคัญในการนำ AI ไปใช้ในงานเกษตรแบบเจาะจงที่ซับซ้อน เช่น การจัดการโรคข้าวในแต่ละพื้นที่หรือภูมิอากาศ [41] ด้วยโมเดล CNN นี้ การฝึกฝนโมเดลด้วยข้อมูลภาพของโรคข้าวจากหลากหลายแหล่งข้อมูลทำให้สามารถเรียนรู้และวิเคราะห์โรคได้อย่างมีประสิทธิภาพ

นอกจากนี้ การใช้ โมเดลที่ได้รับการฝึกฝนล่วงหน้า (Pretrained Models) และ วิธีการเรียนรู้เชิงลึก (Deep Learning Approaches) ยังเป็นก้าวสำคัญที่ช่วยลดความยุ่งยากในการฝึกฝนโมเดลและเพิ่มความแม่นยำในการตรวจจับโรคข้าว การใช้โมเดลที่มีการฝึกฝนล่วงหน้านี้ช่วยเสริมความสามารถในการสกัดคุณลักษณะเฉพาะจากภาพโรคข้าว ทำให้สามารถระบุลักษณะโรคได้อย่างแม่นยำ โดยไม่จำเป็นต้องฝึกฝนโมเดลใหม่ตั้งแต่ต้น ซึ่งจะช่วยประหยัดเวลาและทรัพยากร [42] กระบวนการเรียนรู้เชิงลึกที่ทันสมัยยังทำให้สามารถวิเคราะห์ข้อมูลที่มีปริมาณมากได้อย่างมีประสิทธิภาพ ซึ่งช่วยให้เกษตรกรและผู้เชี่ยวชาญสามารถใช้ระบบ AI เหล่านี้ในการวินิจฉัยโรคข้าวได้อย่างรวดเร็วและแม่นยำ

การใช้ระบบที่ใช้ AI ในการวิเคราะห์ภาพข้าวยังแสดงให้เห็นถึงอัตราความสำเร็จที่สูงในการวินิจฉัยโรค โดยระบบเหล่านี้สามารถวิเคราะห์ภาพที่ได้จากส่วนต่าง ๆ ของต้นข้าว เช่น ใบข้าว แปลงข้าว หรือเมล็ดข้าว โดยอาศัยการวิเคราะห์ลักษณะเฉพาะของภาพที่ได้รับ การใช้งานนี้ช่วยเพิ่มความสามารถในการวิเคราะห์ให้ละเอียดและรวดเร็วขึ้น ซึ่งมีข้อได้เปรียบที่ชัดเจนเมื่อเทียบกับวิธีการตรวจจับโรคแบบดั้งเดิมที่อาจใช้เวลานานและต้องพึ่งพาความชำนาญของผู้เชี่ยวชาญเฉพาะทาง [43] นอกจากนี้การประมวลผลภาพที่ใช้ AI ยังช่วยลดข้อผิดพลาดที่อาจเกิดจากการวิเคราะห์ด้วยวิธีการแบบดั้งเดิมและเพิ่มความเชื่อถือได้ในการวินิจฉัยโรค

การใช้เทคโนโลยีเหล่านี้ยังมีข้อได้เปรียบเหนือวิธีการตรวจจับโรคข้าวแบบดั้งเดิม ซึ่งอาจต้องใช้เวลามากและมีข้อจำกัดหลายประการในการทำงาน การตรวจจับโรคโดยทั่วไปอาจต้องอาศัยการตรวจวิเคราะห์ที่ใช้ความชำนาญของผู้เชี่ยวชาญซึ่งอาจมีข้อผิดพลาดและความไม่น่าเชื่อถือในบ างกรณี ในขณะที่การใช้ระบบ AI ที่สามารถทำงานอย่างต่อเนื่องและคงที่ มีความแม่นยำที่สูงกว่า สามารถช่วยให้การตรวจวินิจฉัยทำได้ตลอดเวลาโดยไม่จำเป็นต้องมีผู้เชี่ยวชาญอยู่ในทุกขั้นตอน [44]

นอกจากนี้ นักวิจัยยังได้พัฒนาระบบสารสนเทศที่ช่วยเหลือเกษตรกรในการวินิจฉัยโรคและการเลือกวิธีการรักษาที่เหมาะสม โดยระบบนี้สามารถให้ข้อมูลเกี่ยวกับโรคที่ตรวจพบ รวมถึงแนวทางการจัดการโรคที่เหมาะสมที่สุด การมีระบบนี้ช่วยให้เกษตรกรสามารถเข้าถึงข้อมูลที่มีความแม่นยำและเชื่อถือได้ ซึ่งส่งเสริมให้เกษตรกรสามารถจัดการกับปัญหาโรคในข้าวได้อย่างทันท่วงที ระบบสารสนเทศนี้ทำให้เกษตรกรสามารถเลือกวิธีการรักษาที่ตรงกับลักษณะโรคและความต้องการของแปลงเกษตรของตนเอง ทำให้ลดความจำเป็นในการพึ่งพาผู้เชี่ยวชาญเฉพาะทางและเพิ่มประสิทธิภาพในการจัดการโรค [45] นอกจากนี้ การสนับสนุนทางข้อมูลยังช่วยให้เกษตรกรสามารถรับมือกับปัญหาที่เกิดขึ้นจากโรคข้าวได้อย่างมีประสิทธิภาพและยืดหยุ่น

แม้ว่าจะมีความก้าวหน้าอย่างมากในการพัฒนาวิธีการตรวจจับและจัดการโรคข้าวโดยใช้ AI และเทคนิคการประมวลผลภาพ แต่ยังคงมีความท้าทายที่ต้องเผชิญในการสร้างระบบการเกษตรอัตโนมัติอย่างสมบูรณ์เพื่อการจัดการโรคข้าว ความท้าทายนี้เกิดจากปัจจัยหลายประการ เช่น ความซับซ้อนของสภาพแวดล้อมทางธรรมชาติที่อาจเปลี่ยนแปลงไปตามฤดูกาลและภูมิอากาศ รวมถึงความหลากหลายของเชื้อโรคที่ทำให้การพัฒนาโมเดลสำหรับการตรวจจับโรคอย่างสมบูรณ์ทำได้ยาก นอกจากนี้ ยังมีข้อจำกัดในเรื่องของการเข้าถึงข้อมูลภาพที่มีคุณภาพสูงและการใช้เทคโนโลยีในพื้นที่ชนบท การพัฒนาโมเดลที่สามารถทำงานได้อย่างครอบคลุมและเชื่อถือได้ในทุกสภาพแวดล้อมยังคงเป็นสิ่งที่ต้องใช้การวิจัยและพัฒนาอย่างต่อเนื่อง [46] ทั้งนี้ ความท้าทายเหล่านี้ทำให้การสร้างระบบการเกษตรที่ทำงานได้อัตโนมัติอย่างเต็มรูปแบบเพื่อการจัดการโรคข้าวเป็นสิ่งที่ต้องคำนึงถึงอย่างมาก และยังคงต้องใช้เวลาในการพัฒนาต่อไป

การวิจัยเพิ่มเติมในด้านนี้มีความสำคัญอย่างยิ่ง เนื่องจากการใช้เทคโนโลยี AI ในการจัดการโรคข้าวสามารถช่วยเพิ่มความมั่นคงทางอาหารและเสริมสร้างความยั่งยืนในภาคเกษตรกรรมทั่วโลก ด้วยการลดความสูญเสียจากการเกิดโรคข้าว ไม่เพียงช่วยเพิ่มผลผลิต แต่ยังช่วยรักษาคุณภาพของผลผลิตเกษตรกร ซึ่งเป็นปัจจัยสำคัญในการพัฒนาเศรษฐกิจและความมั่นคงทางอาหารในประเทศที่ปลูกข้าวเป็นจำนวนมาก เช่น ประเทศในเอเชียและแอฟริกา นอกจากนี้ ความสามารถในการตรวจจับและจัดการโรคข้าวอย่างแม่นยำยังช่วยลดการใช้สารเคมีเกินความจำเป็นซึ่งส่งผลให้สภาพแวดล้อมและสุขภาพของผู้บริโภคดีขึ้น การวิจัยที่ต่อเนื่องในด้านนี้จึงไม่เพียงแต่ช่วยให้เกษตรกรสามารถจัดการโรคในแปลงข้าวได้อย่างมีประสิทธิภาพ แต่ยังช่วยให้เกิดความยั่งยืนในภาคการเกษตรของโลก

บทที่ 3
ขั้นตอนการดำเนินงาน

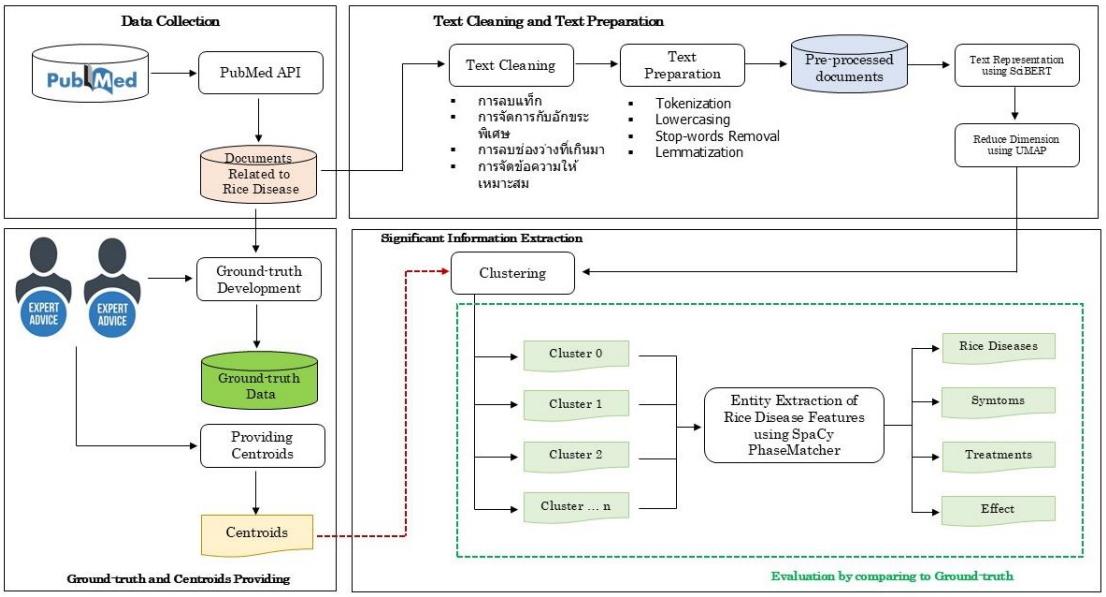
3.1 กรอบการดำเนินงาน

กรอบการดำเนินงานของงานวิจัยนี้ถูกออกแบบเป็นระบบอัตโนมัติแบบ End-to-End โดยมีเป้าหมายหลักในการสกัดข้อมูลเชิงลึกเกี่ยวกับโรคข้าวและแนวทางการรักษาให้มีความแม่นยำและรวดเร็ว ทั้งนี้เพื่อสนับสนุนการพัฒนากลยุทธ์การควบคุมโรคที่มีประสิทธิภาพในภาคการเกษตร โดยระบบได้รับการออกแบบให้รองรับการทำงานในสภาพแวดล้อมที่มีข้อมูลจำนวนมาก ด้วยแนวทางการจัดการแบบ Fault Tolerance, Memory Management และ Adaptive Batch Processing

เริ่มต้นด้วยการรวบรวมวรรณกรรมจากฐานข้อมูล PubMed โดยใช้เทคนิค Advanced Query Optimization ในการสร้างคำค้นที่ครอบคลุมหัวข้อโรคข้าว วิธีการรักษา และอาการที่เกี่ยวข้อง จากนั้นจึงนำ API ของ PubMed ผ่าน BioPython (Entrez) มาดึงข้อมูลแบบแบ่งแบตช์ (Batch Processing) พร้อมทั้งใช้การประมวลผลแบบขนาน (Parallel Data Fetching) ผ่าน ThreadPoolExecutor และระบบ Cache เพื่อให้การดึงข้อมูลเป็นไปอย่างรวดเร็วและมีประสิทธิภาพสูงสุด นอกจากนี้ยังมีการกำหนดกลไก Retry ในกรณีที่เกิดข้อผิดพลาด เพื่อให้ระบบสามารถทำงานได้อย่างต่อเนื่อง

สุดท้าย ระบบจะทำการวิเคราะห์ข้อมูลที่สกัดได้ผ่านแนวทาง Hybrid Analysis ซึ่งผสาน การวิเคราะห์เชิงสถิติ, Entity Extraction เฉพาะทาง และการสร้าง Visualization ผ่าน UAP Plot เพื่อระบุแนวโน้มงานวิจัยที่เกี่ยวข้อง ความสัมพันธ์ระหว่างโรคข้าวและแนวทางการรักษา ตลอดจนการค้นหาช่องว่างทางวิชาการที่ยังไม่ได้รับการศึกษา ระบบนี้ถูกออกแบบให้สามารถรองรับ Fault Tolerance, Memory Management และ Adaptive Batch Processing เพื่อให้สามารถจัดการกับข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพสูงสุด เป้าหมายหลักของงานวิจัยนี้คือการสร้างระบบอัจฉริยะที่สามารถสกัดข้อมูลเชิงลึกเกี่ยวกับโรคข้าวและแนวทางการรักษาได้อย่างแม่นยำและรวดเร็ว ช่วยให้วงการเกษตรสามารถเข้าถึงข้อมูลที่มีประโยชน์ ลดระยะเวลาในการวิเคราะห์ข้อมูลขนาดใหญ่ และสนับสนุนการพัฒนากลยุทธ์การควบคุมโรคข้าวที่มีประสิทธิภาพมากยิ่งขึ้น

กรอบการดำเนินงานสามารถแสดงได้ดังภาพที่ 3.1



ภาพที่ 3.1 กรอบการดำเนินงาน

3.1.1 การรวบรวมข้อมูล (Data Collection)

1) เข้าถึงฐานข้อมูล PubMed: ใช้ PubMed API สำหรับค้นหาและดึงเอกสารวิจัยที่เกี่ยวข้องกับโรคข้าวในประเทศไทย ซึ่งคำค้นหา (keywords) จะครอบคลุมคำที่เกี่ยวข้อง เช่น “*Rice Disease*,” “*Thailand*,” “*Rice Blast*,” “*Bacterial Leaf Blight*,” และ “*Tungro Disease*”

2) รวบรวมบทคัดย่อและเนื้อหาสำคัญ: เก็บข้อมูลสำคัญ เช่น บทคัดย่อ (abstract) ชื่อโรค (disease name) อาการโรค (symptoms) และวิธีการรักษา (treatment methods) โดยใช้ Requests และ BeautifulSoup เพื่อดึงข้อมูลในรูปแบบที่สามารถจัดการได้ง่าย

3.1.2 การตรวจสอบเอกสาร การทำ Ground Truth และกำหนดคำขอ (Query) โดยผู้เชี่ยวชาญ

1) การเลือกผู้เชี่ยวชาญ: คัดเลือกผู้เชี่ยวชาญในสาขาเกษตร โรคพืช หรือผู้เชี่ยวชาญด้านโรคข้าว เพื่อร่วมทำการตรวจสอบข้อมูลและยืนยันความถูกต้องของชุดข้อมูล

2) การทำ Ground Truth: ให้ผู้เชี่ยวชาญตรวจสอบข้อมูลที่ดึงมาว่าเกี่ยวข้องและถูกต้องหรือไม่ เช่น ยืนยันชื่อโรค อาการ และวิธีการรักษา เพื่อให้มั่นใจว่าข้อมูลที่ได้มีความถูกต้องและสอดคล้องกับการใช้งานจริงในงานเกษตร

3) การกำหนดข้อมูลที่จะใช้เป็นคำขอ (Query): ผู้เชี่ยวชาญจะทำการกำหนด “คำขอ” ที่จะใช้เป็น Centroid สำหรับการสกัดข้อมูล

4) ผู้เชี่ยวชาญจะช่วยในการตรวจสอบผลลัพธ์ที่ได้จากการสกัดข้อมูลจากเอกสารงานวิจัยจาก PubMed

3.1.3 การเตรียมเอกสารข้อความ (Text Preparation)

1) การทำความสะอาดข้อความ (Text Cleaning): ใช้ Pandas และ NumPy ในการใช้ Pandas และ NumPy เพื่อจัดการข้อมูลเบื้องต้น เช่น การลบแท็กที่ไม่จำเป็น การจัดการกับอักขระพิเศษ การลบช่องว่างเกินและการปรับรูปแบบข้อความให้เหมาะสม ซึ่งช่วยให้ข้อมูลที่นำไปวิเคราะห์มีความถูกต้องและมีคุณภาพ

2) การประมวลผลภาษาธรรมชาติ (NLP): หลังจากทำความสะอาดแล้ว ข้อความจะถูกเตรียมด้วยการใช้ NLTK หรือ SpaCy เพื่อดำเนินการ Tokenization, Lowercasing, การลบ Stop-words และการทำ Lemmatization ซึ่งช่วยให้ข้อความกระชับและเน้นเฉพาะคำสำคัญที่เกี่ยวข้องกับโรคข้าว

3.1.4 การสกัดสาระสำคัญ (Significant Information Extraction)

1) การลดมิติ(Dimensionality Reduction): ใช้ UMAP ในการลดมิติของข้อมูล embedding ที่ได้จากการประมวลผลภาษาธรรมชาติ ทำให้ข้อมูลมีมิติที่ต่ำลงและสามารถนำไปวิเคราะห์หรือแสดงผลในรูปแบบ 2 มิติได้อย่างมีประสิทธิภาพ

2) การจัดกลุ่มด้วย K-means (Clustering): หลังจากที่ข้อมูลได้รับการลดมิติแล้ว จะนำข้อมูลไปจัดกลุ่มด้วย K-means Clustering โดยการคำนวณหาจุดศูนย์กลางของแต่ละกลุ่มและแบ่งเอกสารออกเป็นกลุ่มตามความคล้ายคลึงกัน ซึ่งจะช่วยให้สามารถแยกหัวข้อที่เกี่ยวข้องกับโรคข้าว เช่น อาการ วิธีการรักษา และชื่อโรค ได้อย่างชัดเจน

3) การสกัดเอนทิตี (Entity Extraction) ด้วย PhraseMatcher): ใช้ SpaCy ร่วมกับ PhraseMatcher ในการสกัดข้อมูลเฉพาะเจาะจงจากข้อความ เช่น ชื่อโรค, อาการ, สาเหตุ และการรักษา โดย PhraseMatcher จะจับคู่คำหลักที่กำหนดไว้ในแต่ละหมวดหมู่ ทำให้สามารถดึงสาระสำคัญออกมาเป็นหมวดหมู่ที่มีความหมายและสามารถนำไปวิเคราะห์ต่อได้อย่างละเอียดและแม่นยำ

3.2 การรวบรวมข้อมูล (Data Collection)

กระบวนการเก็บข้อมูลใน pipeline นี้เริ่มต้นด้วยการสร้าง query ที่ครอบคลุมสำหรับการค้นหาข้อมูลเกี่ยวกับโรคข้าว โดยใช้ทั้งคำหลักทั่วไปและคำเฉพาะที่เกี่ยวข้องกับโรค การรักษา และอาการของโรค ระบบจะใช้ Entrez.esearch เพื่อดึงข้อมูลเบื้องต้น เช่น จำนวนบทความที่เกี่ยวข้อง, WebEnv และ QueryKey ซึ่งจำเป็นสำหรับการเข้าถึงข้อมูลในขั้นตอนถัดไป จากนั้นจะใช้ Entrez.efetch ในการดึงข้อมูลบทความจาก PubMed ในรูปแบบ XML ทีละชุด (batch processing) พร้อมใช้ BeautifulSoup ในการ parse ข้อมูลเพื่อสกัดรายละเอียดที่สำคัญ ได้แก่ ชื่อบทความ, บทคัดย่อ, รหัส PMID และปีที่ตีพิมพ์

เมื่อได้ข้อมูลเบื้องต้นแล้ว ระบบจะทำการกรองและคัดเลือกบทความโดยตัดข้อมูลที่มีบทคัดย่อสั้นเกินไปหรือซ้ำซ้อนออก เพื่อให้มั่นใจว่าข้อมูลที่ได้มีคุณภาพและพร้อมสำหรับการวิเคราะห์ในขั้นตอนต่อไป สุดท้าย ข้อมูลที่ผ่านการคัดกรองจะถูกจัดเก็บใน ไฟล์แคช (pickle file) เพื่อเพิ่มความรวดเร็วในการเข้าถึงและลดภาระในการดึงข้อมูลซ้ำก่อนที่จะถูกนำไปใช้วิเคราะห์

โครงสร้างของกระบวนการ: Input – Process – Output

- Input: ระบบเริ่มต้นด้วยการสร้าง query ที่ครอบคลุมโดยใช้ คำค้นทั้งแบบทั่วไปและแบบเฉพาะเจาะจงเกี่ยวกับโรคข้าว จากนั้นใช้ Entrez API เพื่อดึงข้อมูลบทความจาก PubMed
- Process: ข้อมูลที่ได้รับจะถูก parsing เพื่อแยกข้อมูลสำคัญ ได้แก่ ชื่อบทความ, บทคัดย่อ, รหัส PMID และปีที่ตีพิมพ์ จากนั้นจะผ่านกระบวนการ กรองข้อมูล เพื่อตัดบทความที่ มีบทคัดย่อสั้นเกินไปหรือซ้ำกัน ออก
- Output: ผลลัพธ์ที่ได้จะถูกจัดเก็บในรูปแบบ CSV เพื่อใช้สำหรับการวิเคราะห์ต่อไป

3.3 รายละเอียดขั้นตอนการดำเนินงาน

3.3.1 Pre-processing

ในขั้นตอน Pre-processing นี้จะรับข้อความที่ต้องการประมวลผลโดยเริ่มจากการโหลดโมเดล spaCy (“en_core_web_sm”) แยกในแต่ละ process เพื่อป้องกันการใช้ทรัพยากรร่วมกัน จากนั้นจะสร้าง PhraseMatcher โดยใช้ชุดคำศัพท์ที่เกี่ยวข้องกับโรคข้าว ซึ่งครอบคลุม ชื่อโรค (DISEASE), วิธีการรักษา (TREATMENT), อาการของโรค (SYMPTOM) และผลกระทบ (EFFECT) เมื่อได้รับข้อความ ระบบจะทำการ แปลงเป็นตัวพิมพ์เล็ก (lowercase) เพื่อลดความซ้ำซ้อนของรูปแบบคำ จากนั้นทำ tokenization และ lemmatization โดยคัดกรองเฉพาะ token ที่เป็นตัวอักษรและไม่อยู่ในชุด stopwords (ยกเว้นคำที่ตรงกับคำศัพท์เฉพาะที่กำหนดไว้) หลังจากผ่านกระบวนการคัดกรอง คำที่ได้จะถูกนำมารวมกันเป็น สตริงเดียว เพื่อเตรียมพร้อมสำหรับการวิเคราะห์ในขั้นตอนถัดไป

โครงสร้างกระบวนการ: Input – Process – Output

- Input: ระบบรับข้อมูล Raw Data จากงานวิจัยหรือบทความที่เกี่ยวข้องกับโรคข้าว เพื่อนำมาผ่านกระบวนการแปรรูปก่อนการวิเคราะห์
- Process: ข้อความจะถูก แปลงเป็นตัวพิมพ์เล็ก (lowercase) เพื่อลดความซ้ำซ้อน จากนั้นทำ tokenization และ lemmatization โดยใช้โมเดล spaCy พร้อมทั้งสร้าง PhraseMatcher เพื่อจับคู่คำสำคัญที่เกี่ยวข้องกับโรคข้าว และกรองคำที่ไม่จำเป็นออก

- Output: ผลลัพธ์ที่ได้คือ ข้อความที่ผ่านการประมวลผลแล้ว ซึ่งประกอบด้วยชุด token ที่มีความเหมาะสมสำหรับขั้นตอนการวิเคราะห์ต่อไป เช่น การสร้าง embeddings หรือการจัดกลุ่มข้อมูล

3.3.2 Text Representation using SciBERT Embedding

ในขั้นตอน Text Representation ฟังก์ชันนี้จะรับข้อความดิบและใช้ SciBERT ในการแปลงข้อความเหล่านั้นเป็น เวกเตอร์เชิงตัวเลข (embeddings) ที่สามารถสะท้อนความหมายของเนื้อหาได้อย่างแม่นยำ ระบบจะทำการ แบ่งข้อความออกเป็นชุด (batch) และประมวลผลพร้อมกันผ่าน ThreadPoolExecutor ซึ่งช่วยเร่งการคำนวณโดยใช้ parallelization ภายในแต่ละชุด โดยแต่ละ batch จะถูกประมวลผลผ่านฟังก์ชัน process_batch ที่ทำหน้าที่ส่งข้อความเข้าสู่โมเดล SciBERT เพื่อสร้างเวกเตอร์ตัวแทน เมื่อการประมวลผลเสร็จสิ้น ผลลัพธ์จากทุกชุดจะถูกรวบรวมและรวมเข้าด้วยกันในรูปแบบ numpy array เพื่อให้พร้อมสำหรับการวิเคราะห์หรือการจัดกลุ่มข้อมูลในขั้นตอนต่อไป

โครงสร้างกระบวนการ: Input – Process – Output

- Input: ระบบรับ ข้อความที่ต้องการแปลงเป็น embeddings ซึ่งมาในรูปของรายการข้อความ
- Process: ข้อความจะถูก แบ่งเป็นชุด (batch) ตามขนาดที่กำหนด จากนั้นแต่ละชุดจะถูกประมวลผลพร้อมกันผ่าน ThreadPoolExecutor โดยใช้ฟังก์ชัน process_batch เพื่อแปลงข้อความเป็นเวกเตอร์ representation ด้วย SciBERT และผลลัพธ์จากแต่ละชุดจะถูกรวบรวมเป็นชุดเดียว
- Output: ระบบส่งออกข้อมูลในรูปแบบ numpy array ซึ่งประกอบด้วย embeddings สำหรับข้อความแต่ละรายการ พร้อมสำหรับการนำไปใช้ในกระบวนการวิเคราะห์หรือจัดกลุ่มข้อมูลในขั้นตอนต่อไป

3.3.3 ลดมิติของ Text Representation ด้วยเทคนิค UMAP (Uniform Manifold Approximation and Projection)

ในขั้นตอน การลดมิติด้วย UMAP ระบบจะรับข้อมูลในรูปแบบ matrix ของเวกเตอร์ representation ที่ได้จากโมเดล embedding (เช่น SciBERT) และใช้เทคนิค UMAP (Uniform Manifold Approximation and Projection) ซึ่งเป็นอัลกอริทึมที่ออกแบบมาเพื่อรักษาโครงสร้างเชิงซ้อนของข้อมูลในมิติที่ต่ำลง ระบบจะกำหนด พารามิเตอร์หลัก ได้แก่ จำนวนมิติ (n_components), จำนวนเพื่อนบ้าน (n_neighbors), ระยะห่างขั้นต่ำ (min_dist) และกำหนด random_state เพื่อให้สามารถทำซ้ำได้อย่างสม่ำเสมอ จากนั้นระบบจะดำเนินการ fit_transform(X) เพื่อลดจำนวนมิติของ

เวกเตอร์ลง (เช่น จากเวกเตอร์มิติสูงเป็น 50 มิติ) ซึ่งช่วยให้สามารถนำไปใช้งานในการจัดกลุ่มข้อมูลหรือการวิเคราะห์อื่น ๆ ได้อย่างมีประสิทธิภาพมากขึ้น

โครงสร้างกระบวนการ: Input – Process – Output

- Input: ระบบรับ matrix ของเวกเตอร์ representation ที่มีมิติสูงจากโมเดล embedding
- Process: ใช้ UMAP ในการลดมิติของข้อมูล โดยตั้งค่าพารามิเตอร์หลักและทำ fit_transform กับข้อมูล เพื่อลดมิติให้เหมาะสมกับการวิเคราะห์
- Output: ส่งออก matrix ของเวกเตอร์ representation ที่ถูกลดมิติแล้ว ซึ่งพร้อมสำหรับการนำไปใช้ใน การจัดกลุ่มข้อมูลหรือการวิเคราะห์ขั้นถัดไป

3.3.4 จัดกลุ่มข้อมูลด้วย K-mean Clustering

การจัดกลุ่มข้อมูลด้วย K-means Clustering เป็นเทคนิคที่ใช้ในการแบ่งข้อมูลออกเป็นกลุ่มตามความคล้ายคลึงกัน โดยเริ่มต้นจากการสุ่มเลือก centroid ซึ่งเป็นจุดศูนย์กลางของแต่ละกลุ่ม จากนั้นคำนวณระยะห่างระหว่างแต่ละจุดข้อมูลกับ centroid และจัดสรรข้อมูลไปยังกลุ่มที่ใกล้ที่สุด เมื่อจัดกลุ่มเสร็จแล้ว ระบบจะคำนวณ centroid ใหม่โดยใช้ค่าเฉลี่ยของจุดข้อมูลภายในแต่ละกลุ่ม กระบวนการนี้จะทำซ้ำไปเรื่อย ๆ จนกว่าค่า centroid จะคงที่หรือการเปลี่ยนแปลงอยู่ในระดับที่ยอมรับได้ ส่งผลให้ข้อมูลในแต่ละกลุ่มมีความคล้ายคลึงกันสูงภายในกลุ่ม และแตกต่างจากกลุ่มอื่น

โครงสร้างกระบวนการ: Input – Process – Output

- Input: ระบบรับข้อมูลที่ผ่านการแปลงเป็น เวกเตอร์ embeddings จากกระบวนการประมวลผลและลดมิติ เช่น ข้อมูลที่ได้จาก SciBERT หรือ SentenceTransformer
- Process:
    1) กำหนดค่า centroid เริ่มต้นแบบสุ่ม
    2) คำนวณระยะห่างระหว่างแต่ละจุดข้อมูลกับ centroid
    3) จัดสรรข้อมูลไปยังกลุ่มที่มีระยะห่างน้อยที่สุด
    4) คำนวณ centroid ใหม่ โดยใช้ค่าเฉลี่ยของข้อมูลในแต่ละกลุ่ม
    5) ทำซ้ำกระบวนการจนกว่าค่า centroid จะคงที่หรือลดการเปลี่ยนแปลงให้อยู่ในเกณฑ์ที่ยอมรับได้
- Output: ผลลัพธ์คือ การแบ่งข้อมูลออกเป็นกลุ่มที่ชัดเจน พร้อมการประเมินคุณภาพของการจัดกลุ่มโดยใช้ Silhouette Score และ Inertia เพื่อวัดประสิทธิภาพของโมเดล

3.3.5 การสกัดรคข้าวด้วย PhaseMatcher (Extraction of Rice Disease Features using PhaseMatcher)

ฟังก์ชัน การสกัดข้อมูลจากบทความ ทำงานโดยรับข้อความอินพุตและแปลงให้เป็นเอกสาร (doc) โดยใช้โมเดลภาษา spaCy จากนั้นจะใช้ PhraseMatcher ซึ่งถูกกำหนดให้สามารถตรวจจับคำหลักในแต่ละหมวดหมู่ ได้แก่ โรค (DISEASE), การรักษา (TREATMENT), อาการ (SYMPTOM) และผลกระทบ (EFFECT) ฟังก์ชันจะทำการค้นหาคำที่ตรงกับรายการคำหลักในข้อความ และจัดเก็บผลลัพธ์ในรูปแบบของ เซ็ต (set) เพื่อป้องกันข้อมูลซ้ำซ้อน สุดท้าย ข้อมูลที่สกัดได้จะถูกแปลงเป็น ลิสต์ (list) และจัดเก็บในรูปแบบของ ดิกชันนารี (dictionary) โดยมีคีย์สำหรับแต่ละหมวดหมู่ ก่อนคืนค่าผลลัพธ์ออกมา

โครงสร้างกระบวนการ: Input – Process – Output

- Input: ข้อความที่ต้องการวิเคราะห์ เช่น บทคัดย่อหรือเนื้อหาที่เกี่ยวข้องกับโรคข้าว
- Process:
- แปลงข้อความอินพุตให้เป็นเอกสารด้วย spaCy
- ใช้ PhraseMatcher เพื่อค้นหาคำที่ตรงกับรายการคำหลักของแต่ละหมวดหมู่
- จัดเก็บผลการจับคู่ใน เซ็ต เพื่อป้องกันความซ้ำซ้อน
- Output: ผลลัพธ์เป็น ดิกชันนารี ที่มีคีย์ “DISEASE”, “TREATMENT”, “SYMPTOM” และ “EFFECT” โดยแต่ละคีย์จะมีค่าเป็น ลิสต์ของคำที่พบในข้อความ ซึ่งสามารถนำไปใช้วิเคราะห์เพิ่มเติมในขั้นตอนถัดไป

3.4 ขั้นตอนการจัดกลุ่มและสกัดความรู้จากงานวิจัยโรคข้าว

ในส่วนนี้จแสดงขั้นตอนในการสืบค้นข้อมูลจาก PubMed ตลอดจนการแปลงข้อความเป็นเวกเตอร์ความหมาย, ลดมิติข้อมูล และทำการจัดกลุ่มเอกสาร เพื่อตรวจสอบแนวโน้มของการศึกษาด้านโรคข้าว ระบบยังใช้เทคนิค การสกัดข้อมูลอัตโนมัติ (Automated Information Extraction) เพื่อระบุคำสำคัญใน 4 หมวดหมู่ ได้แก่ โรค (Disease), การรักษา (Treatment), อาการ (Symptom) และผลกระทบ (Effect) โดยมีขั้นตอนดังนี้

ขั้นที่ 1: การตั้งค่าระบบและการจัดการ Log เพื่อเพิ่มประสิทธิภาพการสืบค้นข้อมูลจาก PubMed

การตั้งค่า CONFIG และระบบ Log Management ได้รับการออกแบบเพื่อเพิ่มประสิทธิภาพการทำงานของระบบ โดยกำหนดให้ใช้ ไฟล์แคช "pubmed_cache.pkl" สำหรับจัดเก็บข้อมูลที่ดึงมาจาก PubMed แบบชั่วคราว ซึ่งช่วยลดการเรียกข้อมูลซ้ำซ้อนและเพิ่มความเร็วในการประมวลผล นอกจากนี้ ระบบตั้งค่าพารามิเตอร์ batch_size เป็น 500 เพื่อควบคุมปริมาณข้อมูลที่โหลดในแต่ละรอบ ลดการใช้ทรัพยากรระบบ และป้องกันการโอเวอร์โหลดของข้อมูล

สำหรับการตรวจสอบสถานะการทำงาน ระบบใช้ Realtime Logging ผ่านโมดูล logging โดยบันทึกข้อมูลในระดับ INFO พร้อมระบุเวลาที่เกิดเหตุการณ์อย่างชัดเจน ซึ่งช่วยให้สามารถติดตามความคืบหน้า ตรวจสอบข้อผิดพลาด และวิเคราะห์ประสิทธิภาพของระบบได้อย่างมีประสิทธิภาพยิ่งขึ้น

```
logging.basicConfig(level=logging.INFO,
format="%(asctime)s - %(levelname)s - %(message)s")
```

ภาพที่ 3.2 ตัวอย่างโปรแกรมการ logging

ขั้นที่ 2: การตั้งค่าทางเทคนิคสำหรับการประมวลผลข้อมูล PubMed

ขั้นตอนนี้คือ การตั้งค่าทางเทคนิคประกอบด้วย 3 ส่วนหลัก เพื่อให้ระบบสามารถดึงข้อมูล ประมวลผล และจัดเตรียมข้อความสำหรับการวิเคราะห์ได้อย่างมีประสิทธิภาพ

1) การตั้งค่าการเชื่อมต่อกับ Entrez – กำหนดพารามิเตอร์การเข้าถึงฐานข้อมูล PubMed ผ่าน Web Service โดยระบุ อีเมล (email) และ รหัส API เฉพาะตัว (api_key) เพื่อให้การเรียกใช้งานข้อมูลมีความปลอดภัยและเสถียร
2) การเตรียมเครื่องมือประมวลผลภาษาธรรมชาติ (NLP) – โหลดโมเดล spaCy (en_core_web_sm) พร้อมเสริมความสามารถด้าน ชีวการแพทย์ ผ่าน PhraseMatcher ซึ่งช่วยจับคู่คำศัพท์เฉพาะทางกว่า 100 คำ ที่เกี่ยวข้องกับโรคข้าว การรักษา อาการ และผลกระทบ
3) การดาวน์โหลดทรัพยากรด้านภาษา – ใช้ NLTK (Natural Language Toolkit) สำหรับติดตั้ง tokenizer (punkt) และ stopwords เพื่อปรับปรุงประสิทธิภาพในการทำความสะอาดข้อความ

ระบบทั้งหมดถูกออกแบบให้ทำงาน แบบอัตโนมัติ ตั้งแต่กระบวนการดึงข้อมูลจาก PubMed ไปจนถึง ขั้นตอนการประมวลผลล่วงหน้า (preprocessing) พร้อมบันทึกค่าการตั้งค่าเพื่อให้สามารถนำกลับมาใช้ซ้ำได้ในอนาคต

```
Entrez.email =
"your_email@example.com"
Entrez.api_key = "your_api_key"
nlp = spacy.load("en_core_web_sm")
nltk.download('punkt')
nltk.download('stopwords')
```

ภาพที่ 3.3 ตัวอย่างโปรแกรมการตั้งค่า API และดาวน์โหลดเครื่องด้านภาษา

ขั้นที่ 3: การออกแบบคำค้นหาทางวิทยาศาสตร์เพื่อดึงข้อมูล

กระบวนการออกแบบคำค้นหา (scientific query design) สำหรับการดึงข้อมูลงานวิจัยเกี่ยวกับโรคข้าวใช้แนวทาง การจัดโครงสร้างคำค้นแบบสามชั้น เพื่อให้สามารถค้นหาข้อมูลได้อย่างแม่นยำและครอบคลุม โดยคำนึงถึง ความสมดุลระหว่าง recall และ precision ซึ่งเป็นปัจจัยสำคัญในการสืบค้นข้อมูลเชิงวิชาการ ประกอบด้วยคำในประเภทต่างๆ ต่อไปนี้

(1) คำค้นหาทั่วไป (Generic Terms) - ขั้นตอนแรกกำหนดคำค้นที่มีลักษณะกว้างเพื่อดึงข้อมูลเกี่ยวกับโรคข้าวในบริบทที่ครอบคลุม เช่น “rice disease management”[Title/Abstract] หรือ “rice crop protection”[Title/Abstract] คำค้นเหล่านี้ช่วยให้สามารถดึงบทความที่เกี่ยวข้องกับการจัดการโรคข้าวโดยรวม รวมถึงแนวทางการควบคุมและป้องกันที่มีการศึกษากันอย่างแพร่หลาย

(2) คำค้นหาเฉพาะโรค (Disease-Specific Keywords) - เพื่อเพิ่มความแม่นยำในการสืบค้น ระบบจะเสริมคำค้นที่เจาะจงโรคข้าวแต่ละชนิด โดยใช้ Boolean Operator AND เชื่อมกับคำค้นหาทั่วไป ตัวอย่างเช่น “rice blast”[Title/Abstract] หรือ “bacterial leaf streak”[Title/Abstract] ซึ่งช่วยกรองบทความที่ศึกษาเกี่ยวกับโรคเฉพาะเจาะจง รวมถึงเชื้อสาเหตุที่เกี่ยวข้อง เทคนิคนี้ช่วยให้สามารถคัดแยกงานวิจัยที่เจาะลึกไปยังโรคข้าวแต่ละประเภทได้อย่างมีประสิทธิภาพ

(3) คำค้นหาที่เน้นแนวทางการรักษา (Treatment-Focused Terms) ในส่วนสุดท้ายจะมีการเสริมคำค้นที่เกี่ยวข้องกับวิธีการรักษาและการควบคุมโรค เช่น “fungicide rice disease”[Title/Abstract] หรือ “CRISPR rice disease”[Title/Abstract] เพื่อคัดเลือกเอกสารที่เน้นศึกษาเกี่ยวกับแนวทางการรักษาโรคข้าวโดยเฉพาะ การใช้ Boolean Operator OR ภายในแต่ละชั้นและ AND ระหว่างชั้นช่วยให้สามารถสร้างสมดุลในการสืบค้น ทำให้ผลลัพธ์มีความครอบคลุมทั้งด้านระบาดวิทยา (epidemiology), การวินิจฉัยโรค (diagnosis), และแนวทางการแก้ปัญหาเชิงปฏิบัติ (intervention strategies)

```
query = " OR ".join(base_terms + disease_specific + treatment_specific)
```

ภาพที่ 3.4 ตัวอย่างโปรแกรมการออกแบบคำค้นหาทางวิทยาศาสตร์เพื่อดึงข้อมูล

ขั้นที่ 4: กระบวนการดึงข้อมูลจาก PubMed

กระบวนการดึงข้อมูลจาก PubMed ได้รับการออกแบบให้มี ประสิทธิภาพและความเสถียรสูงสุด โดยใช้แนวทางการ จัดการแคช, การดึงข้อมูลแบบขนาน (parallel fetching), และกลไกการจัดการข้อผิดพลาดอัตโนมัติ เพื่อรองรับการประมวลผลข้อมูลขนาดใหญ่และลดภาระของเซิร์ฟเวอร์

กระบวนการเริ่มต้นด้วย การตรวจสอบไฟล์แคช (“pubmed_cache.pkl”) หากพบว่ามีข้อมูลที่ถูกดึงมาแล้วก่อนหน้านี้ ระบบจะโหลดข้อมูลจากไฟล์แคชแทนการเรียกใช้งาน PubMed API เพื่อลดภาระเครือข่ายและเร่งความเร็วในการประมวลผล อย่างไรก็ตาม หากไม่มีแคช ระบบจะดำเนินการดึงข้อมูลโดยใช้ Entrez API ผ่านกระบวนการสองขั้นตอน

(1) การสืบค้นเบื้องต้น (Initial Querying) - ใช้ Entrez.esearch เพื่อระบุจำนวนบทความที่เกี่ยวข้องกับคำค้น พร้อมสร้าง session key เพื่อจัดการการเข้าถึงข้อมูล

(2) การดึงข้อมูลแบบแบ่งกลุ่ม (Batch Fetching) - เมื่อได้ session key ระบบจะใช้ Entrez.efetch เพื่อดึงข้อมูลจริงในรูปแบบ XML โดยแบ่งเป็น batch ละ 500 บทความ ซึ่งช่วยลดความเสี่ยงของ timeout error และเพิ่มประสิทธิภาพในการจัดการข้อมูลปริมาณมาก

เพื่อเพิ่มความเร็วและลดระยะเวลาประมวลผล ระบบใช้ Parallel Fetching ผ่าน ThreadPoolExecutor ซึ่งช่วยให้สามารถดึงหลายบทความพร้อมกันได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีการตั้งค่า retry mechanism (5 ครั้ง) ควบคู่กับ exponential backoff เพื่อลดปัญหาข้อผิดพลาดชั่วคราว เช่น การเชื่อมต่อขาดหายหรือข้อจำกัดจากเซิร์ฟเวอร์ PubMed

หลังจากได้รับข้อมูลแต่ละ batch ระบบจะใช้ XML parser ในการแปลงข้อมูลให้อยู่ในรูปแบบที่อ่านง่าย และจัดเก็บเป็น DataFrame พร้อมตรวจสอบความถูกต้องของข้อมูล (validation) เช่น กรองบทความที่มี บทคัดย่อสั้นกว่า 100 ตัวอักษร หรือลบข้อมูลซ้ำซ้อนเพื่อรักษาความสมบูรณ์ของชุดข้อมูล

ข้อมูลที่ผ่านการตรวจสอบแล้วจะถูก บันทึกลงไฟล์แคช เพื่อให้สามารถนำกลับมาใช้ใหม่ในอนาคตโดยไม่ต้องดึงข้อมูลจากเซิร์ฟเวอร์ซ้ำ ลดภาระในการประมวลผลและช่วยให้สามารถดำเนินการวิเคราะห์ได้อย่างต่อเนื่องและมีประสิทธิภาพสูงสุด

```
for start in range(0, min(max_articles, total_count), batch_size):
    batch_data = efetch(...)
```

ภาพที่ 3.5 ตัวอย่างโปรแกรมการดึงข้อมูลจาก PubMed

ขั้นที่ 5: การเตรียมข้อมูลด้วยเทคนิค NLP

กระบวนการเตรียมข้อมูล (Data Preprocessing) สำหรับการวิเคราะห์ข้อมูลวิจัยเกี่ยวกับโรคข้าวได้รับการออกแบบให้มี ความถูกต้องแม่นยำและคงไว้ซึ่งความหมายทางวิชาการ โดยใช้เทคนิค Natural Language Processing (NLP) แบบหลายชั้น เพื่อปรับโครงสร้างข้อความให้เหมาะสมสำหรับการประมวลผลและการสร้าง Feature Vector ในขั้นตอนถัดไป

กระบวนการเริ่มต้นด้วย Text Cleaning ซึ่งเป็นการลบ อักขระพิเศษและตัวเลข ที่ไม่จำเป็น พร้อมแปลงข้อความทั้งหมดเป็น ตัวพิมพ์เล็ก (Lowercase) เพื่อลดความซ้ำซ้อนของข้อมูล จากนั้นใช้ spaCy Pipeline สำหรับ Tokenization และ Lemmatization โดยกระบวนการนี้ช่วยให้สามารถแยกคำและแปลงแต่ละคำให้อยู่ในรูปฐานคำ (lemma) เช่น “fungicides” ⟶ “fungicide” เพื่อให้โมเดลสามารถเข้าใจบริบทของคำได้อย่างถูกต้อง นอกจากนี้ ระบบยังใช้ Whitelisting Mechanism เพื่อรักษาคำศัพท์เฉพาะทางที่เกี่ยวข้องกับโรคข้าว เช่น “bakanae disease” หรือ “CRISPR-based resistance” ป้องกันไม่ให้ถูกตัดออกจากการประมวลผล

ในขั้นตอน Stopword Removal ระบบใช้ ชุดคำหยุด (Stopwords) จาก NLTK (Natural Language Toolkit) เพื่อกรองคำที่ไม่มีความหมายสำคัญในการวิเคราะห์ อย่างไรก็ตาม เพื่อรักษาความแม่นยำของข้อมูลทางวิชาการ ระบบได้กำหนด กฎเสริม (custom rules) ที่ ยกเว้นคำใน 4 หมวดหมู่หลัก ได้แก่ โรค (DISEASE), อาการ (SYMPTOM), การรักษา (TREATMENT) และผลกระทบ (EFFECT) โดยใช้ PhraseMatcher ของ spaCy ซึ่งช่วยให้สามารถรักษาคำสำคัญที่เกี่ยวข้องกับโรคข้าวไว้ได้อย่างถูกต้องแม้ว่าจะเป็นคำที่อยู่ในรายการ stopwords ปกติ

ผลลัพธ์ของกระบวนการเตรียมข้อมูลนี้คือ ข้อความมาตรฐานที่คงความหมายหลักทางวิทยาศาสตร์ไว้อย่างครบถ้วน พร้อมสำหรับการนำไปใช้สร้าง Feature Vector และการวิเคราะห์ในขั้นตอนถัดไป ซึ่งช่วยให้โมเดลสามารถเข้าใจและจัดหมวดหมู่ข้อมูลได้อย่างแม่นยำมากยิ่งขึ้น

```
doc = nlp(text.lower())
tokens = [token.lemma_ for token in doc if token.is_alpha and token.text
not in stopwords.words('english')]
```

ภาพที่ 3.6 ตัวอย่างโปรแกรมการเตรียมข้อมูลด้วยเทคนิค NLP

ขั้นที่ 6: การสร้างเวกเตอร์เชิงความหมายด้วย SciBERT

การสร้าง เวกเตอร์เชิงความหมาย (semantic vector representation) ในระบบนี้ใช้ SciBERT ซึ่งเป็นโมเดลภาษาที่ถูกฝึกบนเอกสารวิชาการกว่า 1.14 ล้านชิ้น ทำให้สามารถเข้าใจและประมวลผลข้อมูลที่มี ศัพท์เฉพาะทางด้านวิทยาศาสตร์ การแพทย์ และชีววิทยา ได้อย่างแม่นยำ

กระบวนการเริ่มต้นด้วย การแปลงข้อความเป็น token (Tokenization) โดยใช้ SciBERT Tokenizer ซึ่งออกแบบมาให้สามารถจัดการกับคำศัพท์เชิงเทคนิคและตัวย่อที่มักพบในงานวิจัย จากนั้นข้อมูลที่ได้รับการแปลงเป็น token จะถูกส่งผ่านโครงสร้าง Transformer ที่มี 12 ชั้น เพื่อสร้าง Contextual Embeddings ที่มีมิติขนาด 768 โดยใช้ Token [CLS] ซึ่งเป็น token ตัวแรกของลำดับเป็นตัวแทนความหมายของเอกสารทั้งหมด กลไกนี้ช่วยให้โมเดลสามารถ เก็บบริบทของคำ (contextual meaning) และ เข้าใจความสัมพันธ์เชิงลึกระหว่างคำศัพท์ในงานวิจัย ผ่านกลไก Attention Mechanism

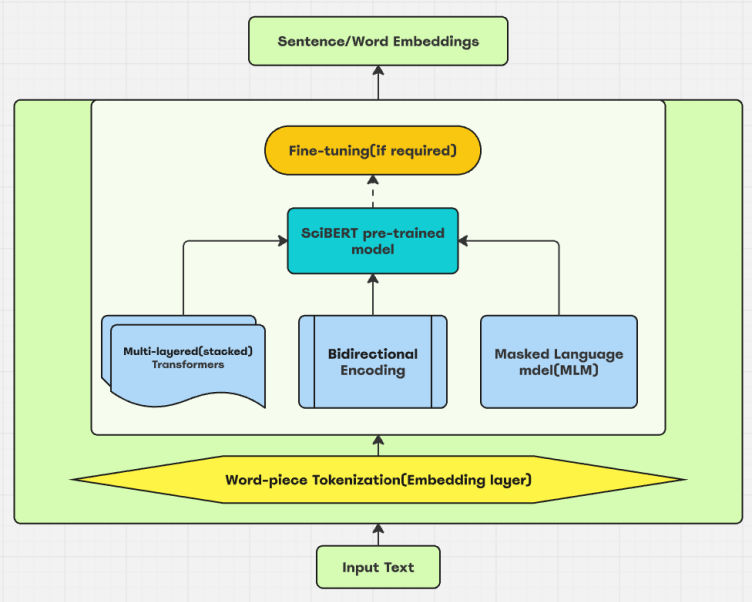
เพื่อให้สามารถประมวลผลข้อความที่ยาวได้อย่างมีประสิทธิภาพ ระบบตั้งค่า max_length = 512 tokens ซึ่งเป็นข้อจำกัดของ SciBERT และใช้กลยุทธ์ truncation (การตัดทอนข้อความอย่างชาญฉลาด) เพื่อคงเนื้อหาที่สำคัญที่สุดของเอกสาร นอกจากนี้ Hidden State ของ Token [CLS] ยังช่วยให้สามารถจับ ความสัมพันธ์เชิงลึกระหว่างแนวคิด เช่น การเชื่อมโยง “rice blast” กับ “fungicide application” ในบริบทของการจัดการโรค รวมถึงการรักษานัยยะของคำศัพท์เชิงสังเคราะห์ เช่น “CRISPR-Cas9 based resistance” ได้อย่างครบถ้วน

โมเดลได้รับการปรับให้สามารถทำงานแบบ Batch Processing เพื่อเพิ่มประสิทธิภาพในการประมวลผล โดยสามารถรันบน GPU หรือ CPU ผ่าน PyTorch ซึ่งช่วยเร่งความเร็วในการสร้างเวกเตอร์ของเอกสารจำนวนมากได้อย่างมีประสิทธิภาพ นอกจากนี้ ระบบยังรองรับ Fallback Mechanism โดยใช้ SentenceTransformer ในกรณีที่มีข้อจำกัดด้านทรัพยากรประมวลผล ซึ่งช่วยให้กระบวนการสร้างเวกเตอร์สามารถดำเนินไปได้อย่างราบรื่นและต่อเนื่องแม้ในสภาพแวดล้อมที่มีข้อจำกัดของฮาร์ดแวร์

กระบวนการนี้ช่วยให้สามารถ แปลงข้อความวิจัยทางวิทยาศาสตร์ให้อยู่ในรูปแบบเวกเตอร์เชิงความหมายที่มีโครงสร้างและบริบทครบถ้วน ซึ่งสามารถนำไปใช้ในการ วิเคราะห์แนวโน้มของงานวิจัย การจัดกลุ่มข้อมูล และการสกัดข้อมูลเชิงลึก ได้อย่างแม่นยำและมีประสิทธิภาพสูงสุด

```
from transformers import AutoTokenizer, AutoModel
tokenizr                                                    =
AutoTokenizer.from_pretrained("allenai/scibert_scivocab_uncased")
model = AutoModel.from_pretrained("allenai/scibert_scivocab_uncased")
outputs  =  model(**tokenizer(texts,  return_tensors="pt",  padding=True,
truncation=True))
embeddings = outputs.last_hidden_state[:, 0, :].numpy()
```

ภาพที่ 3.7 ตัวอย่างโปรแกรมการสร้างเวกเตอร์เชิงความหมายด้วย SciBERT



ภาพที่ 3.8 SciBERT Pre-trained Model for Text-Embedding

ขั้นที่ 7: การลดมิติข้อมูลด้วย UMAP

กระบวนการลดมิติข้อมูล (Dimensionality Reduction) ในระบบนี้ใช้ UMAP (Uniform Manifold Approximation and Projection) ซึ่งเป็นอัลกอริธึมแบบ non-linear manifold learning ที่สามารถบีบอัดคุณลักษณะของข้อมูลในมิติสูงให้อยู่ในมิติที่ต่ำลง โดยยังคงรักษา โครงสร้างเชิงความหมาย และความสัมพันธ์ของข้อมูลเอาไว้

ในขั้นตอนนี้ ระบบใช้ UMAP เพื่อลดขนาดของเวกเตอร์เชิงความหมายจาก SciBERT ซึ่งมีมิติ 768 ให้เหลือ 50 มิติ เพื่อเพิ่มประสิทธิภาพในการ จัดกลุ่มเอกสาร และลดความซับซ้อนของการคำนวณ โดยการกำหนด พารามิเตอร์หลักสองค่า ได้แก่

- n_neighbors = 30 เพื่อรักษา โครงสร้างระดับโลก (global structure) ของข้อมูล ทำให้ข้อมูลที่มีลักษณะคล้ายกันสามารถจัดกลุ่มอยู่ใกล้กันได้
- min_dist = 0.1 เพื่อควบคุมการกระจายตัวของข้อมูลในมิติที่ลดลง ป้องกันการทับซ้อนของข้อมูลมากเกินไป

การใช้ UMAP ในที่นี้ช่วยขจัด สัญญาณรบกวน (noise reduction) ขณะเดียวกันก็ยังคงความสัมพันธ์เชิงความหมายระหว่างเอกสาร ยกตัวอย่างเช่น งานวิจัยที่เกี่ยวข้องกับ “rice blast” และ “brown spot” ซึ่งเป็นโรคที่มีลักษณะใกล้เคียงกันในเชิงพืชวิทยาจะถูกจัดวางให้อยู่ใกล้กันใน latent space

การเลือกให้มิติเป้าหมายเป็น 50 มิติ มาจากผลการทดลองเชิงประจักษ์ เพื่อหาสมดุลระหว่าง การลดความซับซ้อนในการคำนวณ และ การรักษาคุณสมบัติทางโครงสร้างของข้อมูล ซึ่งมี

ความสำคัญต่อกระบวนการจัดกลุ่มข้อมูล การลดมิติโดยใช้ UMAP ยังช่วยแก้ปัญหา “Curse of Dimensionality” ที่มักพบในเวกเตอร์จากโมเดลภาษาขนาดใหญ่ เช่น BERT โดยลดปัญหาการกระจายตัวของข้อมูลในมิติสูงที่ส่งผลให้ระยะห่างแบบ Euclidean Distance ไม่มีความหมาย

นอกจากนี้ การใช้ UMAP ยังช่วยเพิ่มความเร็วของ อัลกอริธึมการจัดกลุ่ม (Clustering Algorithm) ในขั้นตอนต่อไป เช่น K-Means Clustering โดยช่วยลดภาระการคำนวณระยะทางใน high-dimensional space ทำให้การแบ่งกลุ่มมีประสิทธิภาพมากขึ้นและสามารถให้ผลลัพธ์ที่ มีโครงสร้างและความสัมพันธ์ชัดเจนยิ่งขึ้น

```
reducer = umap.UMAP(n_neighbors=30, min_dist=0.1, random_state=42)
reduced_data = reducer.fit_transform(embeddings)
```

ภาพที่ 3.9 ตัวอย่างโปรแกรมการลดมิติข้อมูลด้วย UMAP

ขั้นที่ 8: การจัดกลุ่มข้อมูล (Clustering)

ขั้นตอนการจัดกลุ่มข้อมูล (Clustering) ใช้ อัลกอริธึม K-Means ซึ่งเป็นเทคนิค Unsupervised Learning ที่ช่วยแบ่งหมวดหมู่บทความวิจัยตาม ลักษณะร่วมกัน โดยพิจารณาความคล้ายคลึงของเนื้อหา เช่น ชื่อโรคหลัก, อาการที่เกี่ยวข้อง หรือแนวทางการรักษา

หลักการของ K-Means คือการกำหนดจำนวนกลุ่ม (K clusters) และทำการแบ่งข้อมูลตามตำแหน่งของ centroid หรือจุดศูนย์กลางของแต่ละกลุ่ม โดยกระบวนการนี้ทำงานโดยการลดค่าผลรวมของระยะทางกำลังสอง (Sum of Squared Distances, SSD) ระหว่างจุดข้อมูล x_i และศูนย์กลางกลุ่ม μ_j

$$\min_{C} \sum_{i=1}^{n} \sum_{j=1}^{k} 1(x_i \in C_j) \|x_i - \mu_j\|^2 \tag{3.1}$$

โดยที่

x_i คือเวกเตอร์ของข้อมูลแต่ละจุด

μ_j คือจุดศูนย์กลางของกลุ่ม

C_j คือชุดของจุดข้อมูลที่อยู่ในกลุ่มที่

ฟังก์ชันบ่งชี้ $1(x_i \in C_j)$ ใช้เพื่อตรวจสอบว่าข้อมูล x_i อยู่ในกลุ่ม C_j หรือไม่

กลไกของ K-Means ในบริบทของการวิเคราะห์งานวิจัย

1) กำหนดจำนวนกลุ่ม (K) และสุ่มเลือก centroid เริ่มต้น
2) คำนวณระยะห่างของแต่ละบทความวิจัยกับ centroid โดยใช้ Euclidean Distance

3) กำหนดกลุ่มของข้อมูล โดยจัดหมวดหมู่บทความเข้ากับ centroid ที่ใกล้ที่สุด
4) คำนวณ centroid ใหม่ โดยใช้ค่าเฉลี่ยของจุดข้อมูลภายในแต่ละกลุ่ม
5) ทำซ้ำกระบวนการ (Iterative Refinement) จนกว่าการเปลี่ยนแปลงของ centroid จะน้อยมากหรืออยู่ในเกณฑ์ที่กำหนด

การเลือกจำนวนกลุ่ม (k) ที่เหมาะสมสำหรับ K-Means Clustering เป็นขั้นตอนสำคัญในการปรับแต่งโมเดลให้สามารถจำแนกข้อมูลได้อย่างแม่นยำและมีประสิทธิภาพ โดยในงานวิจัยนี้ ระบบใช้ ดัชนี Silhouette Score ซึ่งเป็นตัวชี้วัดเชิงสถิติที่ประเมินคุณภาพของการจัดกลุ่ม โดยพิจารณาทั้งความหนาแน่นภายในกลุ่ม (Cohesion) และ ความแตกต่างจากกลุ่มอื่น (Separation)

ดัชนี Silhouette Score คำนวณจากค่าเฉลี่ยของ Silhouette Coefficient $S(i)$ สำหรับแต่ละจุดข้อมูล i ตามสมการ:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3.2}$$

โดยที่

- $a(i)$ คือ ค่าเฉลี่ยของระยะทางภายในกลุ่ม ซึ่งสะท้อนถึงความคล้ายคลึงของจุดข้อมูล i กับจุดอื่นในกลุ่มเดียวกัน ยิ่งค่าต่ำ หมายถึงข้อมูลในกลุ่มมีความใกล้เคียงกันมาก
- $b(i)$ คือ ค่าเฉลี่ยของระยะทางระหว่างกลุ่ม โดยพิจารณาจากระยะทางระหว่าง i และกลุ่มที่ใกล้ที่สุดที่มันไม่ได้เป็นสมาชิก ยิ่งค่าสูง หมายถึงจุดข้อมูลถูกแยกจากกลุ่มอื่นอย่างชัดเจน

ค่าของ Silhouette Score อยู่ในช่วง [−1,1]

- ค่าที่ เข้าใกล้ 1 หมายถึง การจัดกลุ่มที่ชัดเจน ข้อมูลในกลุ่มมีความคล้ายคลึงกันสูงและแยกจากกลุ่มอื่นได้ดี
- ค่าที่ เข้าใกล้ 0 หมายถึง ข้อมูลอยู่ใกล้รอยต่อระหว่างกลุ่ม ซึ่งอาจบ่งบอกถึงจำนวนกลุ่มที่มากเกินไปหรือลักษณะข้อมูลที่มีการซ้อนทับกัน
- ค่าที่ ติดลบ หมายถึง การจัดกลุ่มที่ไม่เหมาะสม โดยข้อมูลอาจถูกจัดเข้ากลุ่มผิดพลาด

การเลือกค่าของ k ที่เหมาะสม - กระบวนการเลือกจำนวนกลุ่ม k เริ่มจากการทดลองค่าต่าง ๆ ของ k ในช่วงที่เหมาะสม (เช่น k = 2 ถึง k = 10) แล้วคำนวณค่า Silhouette Score ของแต่ละค่า จากนั้นเลือก k ที่ให้ค่า Silhouette Score สูงสุด เพื่อให้ได้โครงสร้างกลุ่มที่เหมาะสมที่สุด

ข้อดีของการใช้ Silhouette Score ในการกำหนด k

- ไม่ต้องอาศัยข้อมูลฉลากล่วงหน้า (Unsupervised Method) ทำให้สามารถใช้กับข้อมูลที่ไม่มีป้ายกำกับได้
- สะท้อนคุณภาพของการจัดกลุ่มได้อย่างเป็นระบบ โดยพิจารณาทั้งความคล้ายคลึงภายในกลุ่มและความแตกต่างจากกลุ่มอื่น
- ช่วยป้องกันปัญหาการกำหนดจำนวนกลุ่มมากเกินไปหรือน้อยเกินไป ซึ่งอาจส่งผลต่อความแม่นยำในการวิเคราะห์ข้อมูล

การใช้ Silhouette Score ในกระบวนการเลือกจำนวนกลุ่มช่วยให้การจัดหมวดหมู่บทความวิจัยเป็นไปอย่างมีประสิทธิภาพ สร้าง โครงสร้างข้อมูลที่สามารถนำไปวิเคราะห์และสกัดแนวโน้มของงานวิจัยด้านโรคข้าวได้อย่างชัดเจน

ขั้นที่ 9: การระบุคำศัพท์เฉพาะทางด้วย spaCy's PhraseMatcher

การสกัดคำศัพท์เฉพาะทาง (Domain-specific terminology extraction) ในระบบนี้ใช้ PhraseMatcher ของ spaCy ซึ่งเป็นเครื่องมือที่ออกแบบมาเพื่อ จับคู่คำศัพท์ทางวิชาการแบบกำหนดกฎ (Rule-based Matching) โดยโฟกัสไปที่ 4 หมวดหมู่หลักที่เกี่ยวข้องกับโรคข้าว ได้แก่:

1) โรค (DISEASE) เช่น “rice blast”, “bacterial leaf streak”
2) การรักษา (TREATMENT) เช่น “fungicide”, “CRISPR-based resistance”
3) อาการ (SYMPTOM) เช่น “leaf wilting”, “chlorosis”
4) ผลกระทบ (EFFECT) เช่น “yield loss”, “economic impact”

กลไกการทำงานของระบบ Phrase Matching - ก่อนดำเนินการประมวลผล ระบบจะทำการเตรียมคำศัพท์ล่วงหน้า โดยแปลงทุกคำเป็น ตัวพิมพ์เล็ก (lowercase) และใช้ Attribute "LOWER" ของ spaCy เพื่อให้สามารถจับคู่คำศัพท์ได้โดยไม่คำนึงถึงตัวพิมพ์เล็ก-ใหญ่ นอกจากนี้ คำศัพท์ทั้งหมดจะถูกแปลงเป็น Doc Object ซึ่งช่วยให้ระบบสามารถเปรียบเทียบคำศัพท์แบบ Token-to-Token Matching ได้อย่างแม่นยำ ขณะประมวลผลเอกสาร ระบบจะทำการ สแกนทุกประโยคในข้อความวิจัยเพื่อตรวจสอบการปรากฏของคำศัพท์เฉพาะทางที่อยู่ในแต่ละหมวดหมู่ ผลลัพธ์ที่ได้จะถูก จัดเก็บในรูปแบบ Dictionary โดยแยกประเภทตามกลุ่มคำศัพท์ ซึ่งช่วยให้สามารถนำไปใช้วิเคราะห์เชิงปริมาณและสถิติได้อย่างเป็นระบบ

การวิเคราะห์แนวโน้มงานวิจัยผ่านการจับคู่คำศัพท์ - ข้อมูลที่ได้จาก PhraseMatcher สามารถนำไปใช้ในการตรวจสอบ แนวโน้มของหัวข้อการวิจัย ภายในแต่ละกลุ่มข้อมูล ยกตัวอย่างเช่น หากพบว่าในคลัสเตอร์หนึ่งมีการกล่าวถึง “rice blast” ร่วมกับ “silicon amendment” บ่อยครั้ง อาจบ่งชี้ถึงแนวโน้มงานวิจัยที่เน้นการใช้ สารซิลิกอน ในการควบคุมโรคใบไหม้ ซึ่งสามารถใช้เป็น ข้อมูล

สนับสนุนการตีความผลลัพธ์ของการจัดกลุ่ม และช่วยให้นักวิจัยเข้าใจแนวทางการศึกษาที่กำลังได้รับความสนใจ

ข้อดีของการใช้ PhraseMatcher:

- แม่นยำและมีประสิทธิภาพสูง – เนื่องจากเป็น rule-based approach ที่สามารถกำหนดชุดคำศัพท์ล่วงหน้าได้
- รองรับการวิเคราะห์เอกสารจำนวนมาก – ทำงานได้รวดเร็วกว่าเทคนิคที่ใช้ Named Entity Recognition (NER) แบบดั้งเดิม
- ช่วยสกัดองค์ความรู้จากข้อมูลที่ไม่มีโครงสร้าง (unstructured text) – ทำให้สามารถดึงแนวโน้มของหัวข้อวิจัยจากบทความทางวิทยาศาสตร์ได้อย่างมีประสิทธิภาพ

กระบวนการนี้ช่วยให้การ วิเคราะห์แนวโน้มงานวิจัยเกี่ยวกับโรคข้าวมีโครงสร้างที่ชัดเจน และสามารถนำไปใช้เพื่อสนับสนุนการตัดสินใจในการวิจัยและพัฒนาแนวทางการจัดการโรคพืชได้อย่างเป็นระบบ

```
disease_terms = [
    "rice blast", "bacterial blight", "rice stripe virus",
    "brown spot", "sheath blight", "bacterial leaf streak",
    "tungro virus", "bakanae disease", "stem rot",
    "false smut", "rice yellow mottle virus"
]

treatment_terms = [
    "fungicide", "pesticide", "integrated pest management",
    "biocontrol", "chemical control", "cultural practices",
    "resistant varieties", "crop rotation", "seed treatment",
    "biological control", "disease-resistant cultivars"
]

symptom_terms = [
    "leaf spotting", "wilting", "stunted growth",
    "chlorosis", "necrosis", "lesions",
    "yellowing", "discoloration", "leaf blight",
    "leaf streak", "rotting", "grain discoloration"
]

effects_terms = [
    "yield loss", "crop damage", "economic loss",
    "reduced grain quality", "harvest failure", "growth inhibition",
    "production decline", "food security threat"
]
```

```
matcher = PhraseMatcher(nlp.vocab, attr="LOWER")
matcher.add("DISEASE", [nlp.make_doc(term) for term in disease_terms])
```

ภาพที่ 3.10 ตัวอย่างโปรแกรมการระบุคำศัพท์เฉพาะทางด้วย spaCy’s PhraseMatcher

ขั้นที่ 10: การสกัดเอนทิตี (Entity Extraction)

ขั้นตอนการ สรุปผลการจัดกลุ่ม (Cluster Summarization) เป็นกระบวนการสำคัญที่ช่วยให้สามารถ ตีความข้อมูลที่ได้จากการจัดกลุ่มบทความวิจัย ได้อย่างเป็นระบบ โดยอาศัย เทคนิคการวิเคราะห์ความถี่เชิงสถิติ (Statistical Frequency Analysis) และการสกัดเอนทิตี (Entity Extraction) เพื่อระบุ ลักษณะเฉพาะของแต่ละคลัสเตอร์ และทำให้สามารถเชื่อมโยงข้อมูลเชิงบริบทของงานวิจัยได้อย่างมีประสิทธิภาพ

กระบวนการวิเคราะห์และสรุปผล - กระบวนการนี้ดำเนินการผ่าน 4 ขั้นตอนหลัก ได้แก่:

1) รวบรวมข้อมูลภายในคลัสเตอร์ – ดึงข้อมูลจากบทความทั้งหมดที่ถูกจัดอยู่ในแต่ละกลุ่ม เพื่อใช้เป็นฐานข้อมูลสำหรับการวิเคราะห์
2) สกัดคำศัพท์เฉพาะทาง – ใช้ PhraseMatcher ของ spaCy เพื่อดึงคำศัพท์ที่เกี่ยวข้องใน 4 หมวดหมู่หลัก ได้แก่ โรค (DISEASE), อาการ (SYMPTOM), วิธีการรักษา (TREATMENT) และผลกระทบ (EFFECT)
3) คำนวณความถี่สัมพัทธ์ (Relative Frequency Calculation) – ใช้ pandas.Series.value_counts() เพื่อวิเคราะห์ อัตราการเกิดของแต่ละคำศัพท์ภายในคลัสเตอร์ ซึ่งช่วยระบุแนวโน้มสำคัญของงานวิจัยในแต่ละกลุ่ม
4) เลือกคำศัพท์ที่พบบ่อยที่สุด – คัดเลือก 5 อันดับแรก ของคำศัพท์ในแต่ละหมวดหมู่เพื่อสร้าง โปรไฟล์ของคลัสเตอร์ และทำให้สามารถตีความความสัมพันธ์ของข้อมูลได้อย่างเป็นระบบ

โครงสร้างของผลลัพธ์ - ผลลัพธ์ที่ได้จากกระบวนการนี้จะแสดงในรูปแบบ ตารางเชิงสรุป ซึ่งช่วยให้สามารถ วิเคราะห์แนวโน้มของงานวิจัยในแต่ละกลุ่ม ได้อย่างชัดเจน โดยมีองค์ประกอบดังนี้:

1) จำนวนบทความในคลัสเตอร์ (Cluster Size) – แสดงจำนวนบทความทั้งหมดที่ถูกจัดอยู่ในแต่ละกลุ่ม
2) โรคที่พบบ่อยที่สุด (Common Diseases) – ตัวอย่างเช่น { "rice blast": 45, "bacterial blight": 32 } ซึ่งช่วยชี้ให้เห็นว่าโรคใดเป็นหัวข้อหลักของกลุ่ม
3) วิธีการรักษาที่โดดเด่น (Common Treatments) – เช่น { "fungicide": 28, "CRISPR": 15 } ซึ่งสะท้อนให้เห็นแนวทางการรักษาที่ได้รับความสนใจมากที่สุด
4) อาการหลัก (Common Symptoms) และผลกระทบที่สำคัญ (Common Effects) – ทำให้สามารถสรุปได้ว่าแต่ละกลุ่มเน้นศึกษาประเด็นใดในด้านอาการของโรคและผลกระทบทางเศรษฐกิจหรือสิ่งแวดล้อม

ความสำคัญของการสรุปผลการจัดกลุ่ม - กระบวนการนี้ช่วยให้สามารถ ทำความเข้าใจแนวโน้มของงานวิจัยเกี่ยวกับโรคข้าวได้อย่างเป็นระบบ ช่วยให้สามารถ:

- ระบุหัวข้อการศึกษาหลักภายในแต่ละคลัสเตอร์ เช่น กลุ่มที่มุ่งเน้น การใช้ชีววิธีในการควบคุมโรคข้าว อาจแตกต่างจากกลุ่มที่ศึกษา การพัฒนาพันธุ์ต้านทานโรค
- เชื่อมโยงความสัมพันธ์ระหว่างหัวข้อการวิจัย โดยการดูแนวโน้มของ โรค อาการ และวิธีการรักษา ที่มักปรากฏร่วมกัน
- สนับสนุนการตัดสินใจทางวิชาการและเชิงนโยบาย ช่วยให้นักวิจัยสามารถมุ่งเน้นไปที่ประเด็นที่ได้รับความสนใจหรือยังไม่ได้รับการศึกษาอย่างเพียงพอ

กระบวนการนี้ไม่เพียงช่วยให้เข้าใจโครงสร้างของข้อมูลที่จัดกลุ่มได้ดีขึ้น แต่ยังเป็นเครื่องมือสำคัญในการสกัดองค์ความรู้และแนวโน้มการศึกษาวิจัยด้านโรคข้าวในระดับสากล

```
def summarize_clusters(df):
    """
    Provides comprehensive cluster summaries with entity
frequencies.
    """
    results = []
    for cluster_id in sorted(df['cluster'].unique()):
        cluster_df = df[df['cluster'] == cluster_id]
        texts = cluster_df['processed_abstract'].tolist()
        entities_list = [extract_entities(text) for text in texts]

        # Extract all entity types
        entities = {
            entity_type: [e for ent in entities_list for e in
ent.get(entity_type, [])]
            for   entity_type   in   ['DISEASE',   'TREATMENT',
'SYMPTOM', 'EFFECT']
        }

        # Calculate frequencies for each entity type
        frequencies = {
            f"common_{k.lower()}s":
pd.Series(v).value_counts().head(5).to_dict()
            for k, v in entities.items() if v
        }

        results.append({
            "cluster": cluster_id,
            "size": len(cluster_df),
            **frequencies
        })

    return pd.DataFrame(results)
```

ภาพที่ 3.11 ตัวอย่างโปรแกรมการสกัดเอนทิตี (Entity Extraction)

ขั้นที่ 11: การประเมิน

การประเมินประสิทธิภาพการจัดกลุ่มใช้ดัชนี 3 ตัวหลักที่ทำงานเสริมกันเชิงทฤษฎี

1) Silhouette Score - วัดความชัดเจนของกลุ่ม โดยพิจารณาความใกล้ชิดภายในกลุ่ม (a(i)*a(i)*) และระยะห่างจากกลุ่มเพื่อนบ้านใกล้สุด (b(i)*b(i)*) ค่าใกล้ 1 บ่งชี้กลุ่มมีความเป็นเอกภาพสูงและแยกจากกลุ่มอื่นดี

$$s(i) = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)} \tag{3.2}$$

- $a(i)$ ระยะเฉลี่ยภายในกลุ่มเดียวกัน
- $b(i)$ ระยะเฉลี่ยถึงกลุ่มที่ใกล้ที่สุด

ตัวอย่างการคำนวณ Silhouette Score - สมมติว่าเรามีข้อมูล 6 จุด ที่ถูกจัดกลุ่มออกเป็น 2 คลัสเตอร์ ดังตารางข้างล่างนี้

ตารางที่ 3.1 ตารางตัวอย่างการคำนวณ Silhouette Score

จุด	คลัสเตอร์	ค่าเฉลี่ยระยะห่างภายในคลัสเตอร์ (a)	ค่าเฉลี่ยระยะห่างไปยังคลัสเตอร์ที่ใกล้ที่สุด (b)
A	1	2.0	5.0
B	1	2.2	4.8
C	1	2.1	4.9
D	2	1.5	5.5
E	2	1.7	5.3
F	2	1.6	5.4

การคำนวณ Silhouette Score :

- จุด A: $s_A = \frac{5.0-2.0}{\max(2.0,5.0)} = \frac{3.0}{5.0} = 0.6$
- จุด B: $s_B = \frac{4.8-2.2}{\max(2.2,4.8)} = \frac{2.6}{4.8} \approx 0.54$
- จุด C: $s_C = \frac{4.9-2.1}{\max(2.1,4.9)} = \frac{2.8}{4.9} \approx 0.57$
- จุด D: $s_D = \frac{5.5-1.5}{\max(1.5,5.5)} = \frac{4.0}{5.5} \approx 0.73$
- จุด E: $s_E = \frac{5.3-1.7}{\max(1.7,5.3)} = \frac{3.6}{5.3} \approx 0.68$
- จุด F: $s_F = \frac{5.4-1.6}{\max(1.6,5.4)} = \frac{3.8}{5.4} \approx 0.70$

คำนวณ Silhouette Score โดยรวม

$$SilhouetteScore = \frac{s_A + s_B + s_C + s_D + s_E + s_F}{6}$$

$$= \frac{0.6 + 0.54 + 0.57 + 0.73 + 0.68 + 0.70}{6} \approx 0.64$$

1) Calinski-Harabasz Index เน้นอัตราส่วนความแปรปรวนระหว่างกลุ่มต่อภายในกลุ่ม ค่าสูงหมายถึงกลุ่มมีความหนาแน่นและแตกต่างชัดเจน

$$CH = \frac{SS_B}{SS_W} \times \frac{N - K}{K - 1} \quad (3.3)$$

- SS_B ผลรวมระยะกำลังสองระหว่างกลุ่ม (Between-cluster variance)
- SS_w ผลรวมระยะกำลังสองภายในกลุ่ม (Within-cluster variance)
- N จำนวนข้อมูลทั้งหมด
- K จำนวนกลุ่ม

ตัวอย่างการคำนวณ Calinski-Harabasz Index โดยใช้ข้อมูลเดียวกับตัวอย่างการคำนวณ Silhouette Score ซึ่งกำหนดให้มี 6 จุด อยู่ใน 2 คลัสเตอร์ ดังนี้:

เซ็นทรอยด์ของข้อมูลทั้งหมด (Global Centroid) คือ

ตารางที่ 3.2 ตารางตัวอย่างการคำนวณ Calinski-Harabasz Index

จุด	คลัสเตอร์	ค่าเฉลี่ยจุดในคลัสเตอร์ (Centroid)
A	1	(2.0, 2.0)
B	1	(2.0, 2.0)
C	1	(2.0, 2.0)
D	2	(8.0, 8.0)
E	2	(8.0, 8.0)
F	2	(8.0, 8.0)

$$C_g = \left(\frac{(2 + 2 + 2 + 8 + 8 + 8)}{6}, \frac{(2 + 2 + 2 + 8 + 8 + 8)}{6}\right) = (5,5)$$

ระยะห่างระหว่างแต่ละเซ็นทรอยด์กับ Cg :

$$d(C_1, C_g) = \sqrt{(2-5)^2 + (2-5)^2} = \sqrt{9+9} = \sqrt{18}$$

$$d(C_2, C_g) = \sqrt{(8-5)^2 + (8-5)^2} = \sqrt{9+9} = \sqrt{18}$$

คำนวณ $Tr(B_k)$:

$$(Tr(B_k) = 3 \times 18 + 3 \times 18 = 108$$

คำนวณ Tr(Wk) (ระยะห่างภายในคลัสเตอร์):

คำนวณจากระยะห่างระหว่างจุดแต่ละจุดกับเซ็นทรอยด์ของคลัสเตอร์ตัวเอง เนื่องจากแต่ละจุดในคลัสเตอร์มีค่าเฉลี่ยที่ใกล้กับเซ็นทรอยด์มาก ค่า $Tr(W_k)$ จะต่ำ สมมติให้ค่า $Tr(W_k) = 6$ (สมมุติเพื่อความง่าย)

คำนวณค่า CH Index:

$$CH = \frac{108}{6} \times \frac{6-2}{2-1} = 18 \times 4 = 72$$

2) Davies-Bouldin Index ประเมินความคล้ายกลุ่มผ่านอัตราส่วนการกระจายภายในกลุ่มต่อระยะระหว่างเซนทรอยด์ ค่าต่ำกว่า 0.5 ถือว่าดี เนื่องจากกลุ่มไม่เหลื่อมซ้อนกัน

$$DB = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \quad (3.4)$$

- σ_i, σ_j : ค่าเฉลี่ยระยะทางภายในกลุ่ม i และ j
- $d(C_i, C_j)$: ระยะทางระหว่างเซนทรอยด์ของกลุ่ม i และ j

ตัวอย่างการคำนวณ Davies-Bouldin Index โดยใช้ข้อมูลเดียวกับที่ใช้ใน Silhouette Score และ Calinski-Harabasz Index สมมติมี 6 จุด อยู่ใน 2 คลัสเตอร์

ตารางที่ 3.3 ตารางตัวอย่างการคำนวณ Davies-Bouldin Index

จุด	คลัสเตอร์	พิกัด (x, y)
A	1	(1,2)
B	1	(2,1)
C	1	(2,3)
D	2	(8,8)
E	2	(9,9)
F	2	(7,8)

คำนวณค่า Si (ความกระจัดกระจายในคลัสเตอร์)

คลัสเตอร์ 1 (Centroid C1 = (1.67,2)) :

$$S_1 = \frac{d(A,C_1) + d(B,C_1) + d(C,C_1)}{3}$$

$$= \frac{\sqrt{(1-1.67)^2 + (2-2)^2} + \sqrt{(2-1.67)^2 + (1-2)^2} + \sqrt{(2-1.67)^2 + (3-2)^2}}{3}$$

$$= \frac{\sqrt{0.4489} + \sqrt{0.9489} + \sqrt{1.4489}}{3}$$

$$\approx \frac{0.67 + 0.97 + 1.20}{3} = 0.95$$

คลัสเตอร์ 2 (Centroid C1 = (8,8.33)) :

$$S_2 = \frac{d(D,C_2) + d(E,C_2) + d(F,C_2)}{3}$$

$$= \frac{\sqrt{(8-8)^2 + (8-8.33)^2} + \sqrt{(9-8)^2 + (9-8.33)^2} + \sqrt{(7-8)^2 + (8-8.33)^2}}{3}$$

$$= \frac{\sqrt{0.1089} + \sqrt{1.4489} + \sqrt{1.1089}}{3}$$

$$\approx \frac{0.33 + 1.20 + 1.05}{3} = 0.86$$

คำนวณค่า $\mathrm{d}(C_i, C_j)$ (ระยะห่างระหว่างเซ็นทรอยด์ของแต่ละคลัสเตอร์) :

$$d(C_1,C_2) = \sqrt{(1.67-8)^2 + (2-8.33)^2}$$
$$= \sqrt{(6.33)^2 + (6.33)^2} = \sqrt{80.1989} \approx 8.95$$

คำนวณค่า $R_{1,2}$:

$$R_{1,2} = \frac{S_1 + S_2}{d(C_1,C_2)}$$
$$= \frac{0.95 + 0.86}{8.95} = \frac{1.81}{8.95} \approx 0.20$$

คำนวณ Davies-Bouldin Index:

$$DB = \frac{1}{2}\sum_{i=1}^{2} \max_{j \neq i} R_{i,j}$$

เนื่องจากมีแค่ 2 คลัสเตอร์ ค่า $(R_{1,2})$ และ $(R_{2,1})$ มีค่าเท่ากัน $DB = \frac{1}{2}(0.20 + 0.20) = 0.20$

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลอง

ในการออกแบบและพัฒนาระบบ ได้มีการเปรียบเทียบ ผลลัพธ์ของอัลกอริธึมการจัดกลุ่ม K-Means ในรูปแบบที่แตกต่างกัน ได้แก่ K-Means Clustering, Single K-Means Clustering และ Hierarchical K-Means Clustering (HKMeans) เพื่อประเมินประสิทธิภาพของแต่ละอัลกอริธึมภายใต้เงื่อนไขต่าง ๆ

การวิเคราะห์ดำเนินการโดยใช้ ตัวชี้วัดคุณภาพของการจัดกลุ่ม (Clustering Evaluation Metrics) ได้แก่:

- Inertia – วัดผลรวมของระยะทางกำลังสองระหว่างจุดข้อมูลและเซนทรอยด์ของกลุ่ม (ยิ่งต่ำ ยิ่งดี)
- Silhouette Score – วัดระดับการแยกตัวของกลุ่มและความคล้ายคลึงกันภายในคลัสเตอร์ (ยิ่งสูง ยิ่งดี)
- Calinski-Harabasz Index – วัดอัตราส่วนระหว่างความกระจายของข้อมูลภายในคลัสเตอร์กับความกระจายระหว่างคลัสเตอร์ (ค่าสูงบ่งบอกถึงการจัดกลุ่มที่ชัดเจน)
- Davies-Bouldin Index – วัดความคล้ายคลึงระหว่างคลัสเตอร์ โดยค่ายิ่งต่ำหมายถึงการแยกกลุ่มที่ดียิ่งขึ้น

การทดลองดำเนินการใน สองเงื่อนไข คือ

- ไม่มีการลดมิติ (No Red Dim) – ใช้ข้อมูลต้นฉบับที่มีมิติสูง
- มีการลดมิติ (Red Dim) – ใช้เทคนิคลดมิติ เช่น UMAP เพื่อลดความซับซ้อนของข้อมูลก่อนการจัดกลุ่ม

4.1.1 K-Means Clustering

การวิเคราะห์ประสิทธิภาพของอัลกอริธึม K-Means ระหว่างชุดข้อมูลต้นฉบับและชุดข้อมูลที่ผ่านการลดมิติแล้ว แสดงให้เห็นความแตกต่างที่ชัดเจนในหลายมิติ ดังนี้

แนวโน้มของค่า Inertia

ค่า Inertia มีแนวโน้มลดลงเมื่อจำนวนคลัสเตอร์เพิ่มขึ้นในทั้งสองสถานการณ์ ซึ่งสอดคล้องกับหลักการทำงานของ K-Means ที่พยายามลดระยะทางภายในคลัสเตอร์ อย่างไรก็ดี การลดมิติข้อมูล

ส่งผลให้ค่า Inertia ลดลงอย่างมาก (จาก 87,288.48 เหลือ 1,836.53 เมื่อใช้ 2 คลัสเตอร์) แสดงว่าข้อมูลในมิติที่น้อยลงมีการกระจายตัวที่กระชับขึ้น

ดัชนีประเมินประสิทธิภาพ

Silhouette Score: ในข้อมูลต้นฉบับ คะแนนลดลงจาก 0.10676 (2 คลัสเตอร์) เป็น 0.03856 (10 คลัสเตอร์) ชี้ว่าคลัสเตอร์มีความเหลื่อมล้ำมากขึ้นเมื่อเพิ่มจำนวน ในทางตรงข้าม ข้อมูลที่ลดมิติแล้วมีคะแนนสูงกว่าอย่างมีนัยสำคัญ (0.40638 ที่ 2 คลัสเตอร์) และลดลงเพียงเล็กน้อย แสดงว่าการลดมิติช่วยรักษาความชัดเจนของขอบเขตคลัสเตอร์ได้ดีกว่า

Calinski-Harabasz Index: ในข้อมูลต้นฉบับดัชนีลดลงจาก 88.107 เป็น 32.871 เมื่อเพิ่มคลัสเตอร์ สะท้อนว่าความแยกกันของคลัสเตอร์ลดลง ในขณะที่ข้อมูลลดมิติมีดัชนีสูงกว่า (ช่วง 617-857) แม้จะลดลงตามจำนวนคลัสเตอร์

Davies-Bouldin Index: ข้อมูลต้นฉบับมีค่าดัชนีสูง (3.16-3.33) ซึ่งบ่งชี้ประสิทธิภาพการจัดกลุ่มต่ำ แต่หลังลดมิติค่าดัชนีดีขึ้นใกล้เคียง 1 (0.98-1.05) แสดงความสมดุลของคลัสเตอร์ที่ดีขึ้น

ผลกระทบจากการลดมิติ

การลดมิติข้อมูลไม่เพียงแต่ลดค่า Inertia อย่างเห็นได้ชัด แต่ยังปรับปรุงดัชนีทั้งหมด โดยเฉพาะ Silhouette Score และ Davies-Bouldin Index ซึ่งชี้ให้เห็นว่าการกำจัดมิติข้อมูลที่ redundant หรือมีสัญญาณรบกวนช่วยให้ K-Means จัดกลุ่มได้มีประสิทธิภาพมากขึ้น อย่างไรก็ตาม ค่า Silhouette Score ที่สูงสุดในข้อมูลลดมิติเกิดขึ้นที่ 2 คลัสเตอร์ (0.40638) และมีแนวโน้มลดลงเมื่อเพิ่มคลัสเตอร์ แสดงว่าจำนวนคลัสเตอร์ที่เหมาะสมอาจไม่จำเป็นต้องเพิ่มตามขนาดข้อมูล

ตารางที่ 4.1 ผลการจัดกลุ่มด้วย K-Means Clustering

n_clusters	Inertia (No Red Dim)	Silhouette (No Red Dim)	Calinski-Harabasz (No Red Dim)	Davies-Bouldin (No Red Dim)	Inertia (Red Dim)	Silhouette (Red Dim)	Calinski-Harabasz (Red Dim)	Davies-Bouldin (Red Dim)
2	87,288.48	0.10676	88.107	3.1643	1,836.53	0.40638	857.131	0.97791
3	83,190.35	0.05774	70.340	3.3831	1,347.05	0.40502	761.934	1.09801
4	80,385.88	0.05635	59.876	3.2663	1,104.92	0.33037	690.224	1.08357
5	78,529.55	0.04335	51.707	3.29535	941.48	0.30800	649.398	1.11847
6	77,131.06	0.03889	45.169	3.27485	797.59	0.31262	647.910	1.11427
7	75,860.41	0.03712	41.340	3.32462	683.23	0.33791	656.844	1.04883

n_clusters	Inertia (No Red Dim)	Silhouette (No Red Dim)	Calinski-Harabasz (No Red Dim)	Davies-Bouldin (No Red Dim)	Inertia (Red Dim)	Silhouette (Red Dim)	Calinski-Harabasz (Red Dim)	Davies-Bouldin (Red Dim)
8	74,788.85	0.03403	37.903	3.23945	601.31	0.34966	658.117	0.98987
9	73,963.69	0.03495	34.986	3.21305	552.31	0.32656	637.110	1.01036
10	72,959.27	0.03856	32.871	3.33290	513.54	0.32555	616.614	1.05407

4.1.2 K-Means Clustering vs. Single K-Means Clustering

ผลลัพธ์จากการเปรียบเทียบ Single K-Means Clustering กับ K-Means Clustering แสดงให้เห็นว่า พฤติกรรมหลักของทั้งสองอัลกอริธึมมีความคล้ายคลึงกัน โดยเฉพาะในด้าน ค่า Inertia ที่ลดลงเมื่อจำนวนคลัสเตอร์เพิ่มขึ้น และ Silhouette Score ที่มีแนวโน้มลดลงตามจำนวนคลัสเตอร์ที่เพิ่มขึ้น อย่างไรก็ตาม มีความแตกต่างในบางตัวชี้วัด ซึ่งสะท้อนถึง โครงสร้างการกระจายตัวของข้อมูลที่แตกต่างกัน

ตารางที่ 4.2 ผลการจัดกลุ่ม Single K-Means Clustering

n_clusters	Inertia (No Red Dim)	Silhouette (No Red Dim)	Calinski-Harabasz (No Red Dim)	Davies-Bouldin (No Red Dim)	Inertia (Red Dim)	Silhouette (Red Dim)	Calinski-Harabasz (Red Dim)	Davies-Bouldin (Red Dim)
2	87,330.14	0.11664	87.597	3.03676	1,836.61	0.40653	857.050	0.97350
3	83,205.22	0.06076	70.239	3.40675	1,347.05	0.36502	761.934	1.09801
4	80,721.50	0.04820	58.269	3.34822	1,144.24	0.32415	655.273	1.16585
5	78,762.80	0.04469	50.830	3.46452	962.94	0.32231	629.475	1.08056
6	77,835.42	0.04410	43.437	3.44824	797.59	0.31246	647.910	1.11423
7	76,402.94	0.04090	39.891	3.49710	684.02	0.33741	655.965	1.05475
8	75,074.78	0.03751	37.228	3.40880	606.69	0.35113	654.015	0.98479
9	74,302.68	0.03608	34.487	3.45781	552.32	0.32692	637.083	1.02019
10	73,435.62	0.03245	31.957	3.33480	518.54	0.31985	609.633	1.03505

ในกรณีของข้อมูลต้นฉบับ (ไม่มีการลดมิติ) พบว่า ค่า Davies-Bouldin Index ของ Single K-Means Clustering ต่ำกว่าของ K-Means Clustering (เช่น 3.03676 เทียบกับ 3.1643 ที่ 2 คลัสเตอร์) ซึ่งบ่งชี้ว่าการกระจายตัวของคลัสเตอร์มีความคมชัดมากขึ้นเล็กน้อย

เมื่อพิจารณาข้อมูลที่ผ่านการลดมิติ Single K-Means ยังคงให้ผลลัพธ์ที่ดีกว่าในบางตัวชี้วัด โดย

- ค่า Davies-Bouldin Index อยู่ในช่วง 0.97-1.16 ซึ่งสะท้อนถึงความชัดเจนของคลัสเตอร์ที่ดีขึ้น
- Silhouette Score สูงกว่าค่าเฉลี่ยของ K-Means Clustering โดยเฉพาะที่ 2 คลัสเตอร์ (0.40653)
- Calinski-Harabasz Index ลดลงช้ากว่าแบบปกติ (จาก 857.05 เหลือ 609.63 เมื่อเพิ่มจำนวนคลัสเตอร์) ซึ่งหมายถึง การคงโครงสร้างของกลุ่มข้อมูลได้ดีขึ้นแม้เมื่อเพิ่มจำนวนคลัสเตอร์

แม้ว่าทั้ง Single K-Means Clustering และ K-Means Clustering จะแสดงพฤติกรรมที่ใกล้เคียงกันโดยรวม แต่ Single K-Means Clustering มีประสิทธิภาพที่ดีกว่าเล็กน้อยในด้านความสมดุลของคลัสเตอร์ โดยเฉพาะเมื่อใช้ร่วมกับ การลดมิติข้อมูล ซึ่งอาจเกิดจากการ ปรับปรุงกระบวนการคำนวณและการกำหนดพารามิเตอร์เริ่มต้นของอัลกอริธึม ส่งผลให้ Single K-Means Clustering เหมาะสำหรับการประมวลผลข้อมูลที่มีมิติสูงและต้องการความเสถียรของโครงสร้างกลุ่มมากขึ้น

4.1.3 HKMeans Clustering

ผลลัพธ์จาก Hierarchical K-Means (HKMeans) Clustering แสดงให้เห็นถึงลักษณะเฉพาะที่แตกต่างจาก K-Means Clustering และ Single K-Means Clustering โดยเฉพาะในด้านความหนาแน่นภายในกลุ่ม (cluster compactness) และความสามารถในการแยกตัวระหว่างกลุ่ม (cluster separation)

หนึ่งในตัวชี้วัดสำคัญที่สะท้อนความแตกต่างนี้คือ ค่า Calinski-Harabasz Index ซึ่งในกรณีของ HKMeans มีค่าต่ำกว่าวิธีอื่นอย่างต่อเนื่อง เช่น

- ข้อมูลต้นฉบับ: 88.107
- ข้อมูลลดมิติ (2 คลัสเตอร์): 857.131

ค่า Calinski-Harabasz Index ที่ต่ำกว่า บ่งชี้ว่า คลัสเตอร์ที่ได้มีความหนาแน่นภายในกลุ่มต่ำกว่า และมีความแยกตัวระหว่างกลุ่มน้อยกว่าวิธีอื่น ๆ

นอกจากนี้ Davies-Bouldin Index ซึ่งใช้วัดระดับการทับซ้อนกันของคลัสเตอร์ ยังพบว่า HKMeans มีค่าที่สูงกว่า K-Means และ Single K-Means ในทุกกรณี โดยเฉพาะเมื่อจำนวนคลัสเตอร์เพิ่มขึ้น เช่น

- ข้อมูลต้นฉบับ: 3.16 - 3.90
- ข้อมูลลดมิติที่ 2 คลัสเตอร์: 0.97791

แสดงให้เห็นว่า HKMeans มีแนวโน้มเกิดการทับซ้อนของกลุ่มข้อมูลมากกว่า เมื่อเปรียบเทียบกับ K-Means แบบอื่น

ผลกระทบของการลดมิติข้อมูล: ค่าของ Inertia ใน HKMeans อยู่ในระดับกลางระหว่าง K-Means และ Single K-Means ทั้งในข้อมูลต้นฉบับและข้อมูลลดมิติ ตัวอย่างเช่น

- Inertia ในข้อมูลลดมิติที่ 2 คลัสเตอร์
  - HKMeans: 1,836.53
  - Single K-Means: 1,836.61

ค่าดังกล่าว สะท้อนให้เห็นว่า HKMeans พยายามสร้างสมดุลระหว่างความกระชับของคลัสเตอร์และความสามารถในการแยกกลุ่ม แม้ว่าการลดมิติข้อมูลจะช่วยปรับปรุงตัวชี้วัดบางค่า เช่น

- Silhouette Score สูงขึ้นเป็น 0.40638
- Davies-Bouldin Index ปรับลดลงเป็น 0.97791

แต่โดยรวมแล้ว ประสิทธิภาพของ HKMeans ยังคงด้อยกว่า Single K-Means เล็กน้อย โดยเฉพาะในด้าน ความเสถียรของ Calinski-Harabasz Index ซึ่งลดลงอย่างรวดเร็วเมื่อจำนวนคลัสเตอร์เพิ่มขึ้น (เช่น ลดจาก 857.131 เหลือ 568.863 เมื่อเพิ่มจาก 2 เป็น 10 คลัสเตอร์)

ตารางที่ 4.3 ผลการจัดกลุ่ม HKMeans Clustering

n_clusters	Inertia (No Red Dim)	Silhouette (No Red Dim)	Calinski-Harabasz (No Red Dim)	Davies-Bouldin (No Red Dim)	Inertia (Red Dim)	Silhouette (Red Dim)	Calinski-Harabasz (Red Dim)	Davies-Bouldin (Red Dim)
2	87,288.49	0.10676	88.107	3.16429	1,836.53	0.40638	857.131	0.97791
3	83,567.52	0.04944	67.808	3.37610	1,403.40	0.34818	711.641	1.10122
4	81,177.27	0.05297	56.107	3.55605	1,153.91	0.30725	647.050	1.17039
5	79,325.03	0.03506	48.734	3.61638	999.47	0.28215	597.524	1.25550
6	77,991.54	0.03492	42.358	3.65216	857.94	0.28124	588.531	1.23902
7	77,040.71	0.03634	38.213	3.79784	754.04	0.30060	579.923	1.09382
8	76,016.10	0.03105	35.041	3.72918	680.22	0.29764	565.592	1.03822
9	75,218.18	0.03107	32.247	3.77632	593.50	0.31358	548.234	1.06122
10	74,561.02	0.03118	29.841	3.90026	549.76	0.31676	568.863	1.12688

แม้ว่าผลการวิเคราะห์จะชี้ให้เห็นว่า HKMeans อาจไม่ได้เหมาะสำหรับการจัดกลุ่มที่ต้องการโครงสร้างที่ชัดเจนและแยกจากกันอย่างเข้มข้น แต่ HKMeans ยังคงเป็นทางเลือกที่ดีสำหรับข้อมูลที่

ต้องการโครงสร้างลำดับชั้น และสามารถสร้างสมดุลระหว่าง ความเรียบง่ายของกลุ่มและความสัมพันธ์เชิงโครงสร้าง ได้ดีกว่า K-Means ปกติในบางบริบท

ดังนั้น HKMeans Clustering อาจเหมาะสำหรับการจัดกลุ่มข้อมูลที่มีลำดับชั้นมากกว่าการจัดกลุ่มที่ต้องการความกระชับสูง เช่น การสำรวจแนวโน้มของงานวิจัยในระดับกว้าง ก่อนเข้าสู่การวิเคราะห์เชิงลึกในคลัสเตอร์ย่อย

4.2 สรุปผลการทดลองเชิงเปรียบเทียบ

จากผลการทดลองจะแสดงการเปรียบเทียบประสิทธิภาพของ อัลกอริธึมการจัดกลุ่ม 3 รูปแบบ ได้แก่ K-Means Clustering, Single K-Means Clustering และ Hierarchical K-Means Clustering (HKMeans) โดยใช้ ตัวชี้วัดประสิทธิภาพหลัก ได้แก่ Inertia, Silhouette Score, Calinski-Harabasz Index และ Davies-Bouldin Index

4.2.1 ผลกระทบของการลดมิติข้อมูลต่อคุณภาพการจัดกลุ่ม

การลดมิติข้อมูลช่วยให้ ค่า Inertia ลดลงอย่างมีนัยสำคัญ ในทุกอัลกอริธึม ตัวอย่างเช่น ค่า Inertia ลดลงจากมากกว่า 87,000 ในข้อมูลต้นฉบับ เหลือน้อยกว่า 2,000 เมื่อใช้การลดมิติที่ 2 คลัสเตอร์ ซึ่งสะท้อนถึงการที่ข้อมูลใน มิติที่ต่ำกว่ามีการกระจายตัวที่กระชับขึ้น ทำให้สามารถกำหนดขอบเขตของคลัสเตอร์ได้อย่างมีประสิทธิภาพมากขึ้น

อย่างไรก็ตาม ผลกระทบต่อดัชนีอื่น ๆ มีความแตกต่างกันขึ้นอยู่กับอัลกอริธึมที่ใช้

- Silhouette Score และ Calinski-Harabasz Index มีแนวโน้มลดลงในบางกรณีหลังจากลดมิติ ซึ่งอาจเกิดจากการที่โครงสร้างบางอย่างของข้อมูลถูกลดทอนลง
- Davies-Bouldin Index ในบางกรณีเพิ่มขึ้น ซึ่งสะท้อนถึงการทับซ้อนของคลัสเตอร์ที่มากขึ้น เช่น ใน HKMeans Clustering ที่มีค่า Davies-Bouldin Index สูงถึง 1.25 ในข้อมูลลดมิติที่ 5 คลัสเตอร์

4.2.2 เปรียบเทียบประสิทธิภาพระหว่างอัลกอริธึม

จากการวิเคราะห์พบว่า Single K-Means Clustering ให้ประสิทธิภาพดีที่สุดในข้อมูลที่ผ่านการลดมิติ โดยมี

- Silhouette Score สูงสุด (0.40653) ที่ 2 คลัสเตอร์
- Davies-Bouldin Index ต่ำสุด (0.97) ซึ่งสะท้อนถึงการแยกตัวของคลัสเตอร์ที่ชัดเจน
- มีความเสถียรของดัชนีดีกว่าเมื่อเพิ่มจำนวนคลัสเตอร์

K-Means Clustering มีแนวโน้มที่คล้ายคลึงกับ Single K-Means Clustering แต่ด้อยกว่าเล็กน้อยในแง่ของ ความสมดุลของโครงสร้างกลุ่ม

ในทางกลับกัน HKMeans Clustering แสดงข้อจำกัดในด้านความแยกตัวของกลุ่ม โดยมี

- Calinski-Harabasz Index ต่ำที่สุด เช่น ลดจาก 857.13 เหลือ 568.86 เมื่อเพิ่มคลัสเตอร์จาก 2 เป็น 10
- Davies-Bouldin Index สูงกว่าวิธีอื่น ๆ แสดงถึงความทับซ้อนกันของกลุ่มที่เพิ่มขึ้น
- ค่า Inertia อยู่ในระดับกลาง ระหว่าง K-Means และ Single K-Means ซึ่งสะท้อนถึงความสมดุลระหว่าง ความเรียบง่ายและโครงสร้างลำดับชั้น

4.2.3 ข้อสรุปและการเลือกใช้งานอัลกอริธึม

Single K-Means Clustering เป็นอัลกอริธึมที่มีประสิทธิภาพดีที่สุดสำหรับข้อมูลที่ผ่านการลดมิติ เนื่องจากให้ Silhouette Score สูงสุดและ Davies-Bouldin Index ต่ำสุด ทำให้เหมาะสำหรับการจัดกลุ่มข้อมูลที่ต้องการความชัดเจนของโครงสร้างคลัสเตอร์

K-Means Clustering ยังคงเป็นตัวเลือกที่เสถียรสำหรับการจัดกลุ่มข้อมูลทั่วไป ขณะที่ HKMeans Clustering เหมาะสำหรับกรณีที่ต้องการโครงสร้างลำดับชั้นของคลัสเตอร์ แม้ว่าจะมีข้อจำกัดในด้านความสามารถในการแยกกลุ่มอย่างชัดเจน

- หากต้องการ การจัดกลุ่มที่ชัดเจนและมีขอบเขตแน่นอน ⟶ Single K-Means เป็นตัวเลือกที่ดีที่สุด
- หากต้องการ ความสมดุลและความเสถียรในหลายเงื่อนไข ⟶ K-Means มาตรฐานยังคงเป็นตัวเลือกที่แข็งแกร่ง
- หากต้องการ การจัดกลุ่มแบบลำดับชั้นที่สามารถรองรับโครงสร้างข้อมูลที่ซับซ้อน ⟶ HKMeans อาจเป็นตัวเลือกที่เหมาะสมกว่า แม้จะมีข้อจำกัดในด้านความชัดเจนของคลัสเตอร์

บทที่ 5

สรุปและอภิปรายผลการทดลอง / สรุปผลและข้อเสนอแนะ

"[คลิกที่นี่เพื่อเริ่มพิมพ์รายละเอียดกล่าวนำ (ถ้ามี)]"

5.1 สรุปผลและอภิปรายผล

[คลิกที่นี่เพื่อพิมพ์รายละเอียด]

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

[คลิกที่นี่เพื่อพิมพ์รายละเอียด]

5.3 ข้อเสนอแนะ

[คลิกที่นี่เพื่อพิมพ์รายละเอียด]

เอกสารอ้างอิง

[1] O. Butso and S. Isvilanonda, (2010), Two Decades of the Rice Economy of Thailand, Applied Economics, Available at https://www.semanticscholar.org/paper/Two-Decades-of-the-Rice-Economy-of-Thailand-Butso-Isvilanonda/9e5cd00bd9708bf4f287f8bc6fd7e15451df418b

[2] ศุภวรรณ วิเศษน้อย และ เวย์น เนลลิส, (2559), ภาพสะท้อนบทบาทของมหาวิทยาลัยในเอเชียตะวันออกเฉียงใต้กับการให้บริการส่งเสริมการเกษตรและความมั่นคงทางอาหาร, UNISEARCH (Unisearch Journal): Vol. 4, Iss. 3, Article 4, หน้า 15-20.

[3] S. Kamonlimsakun, T. Watcharakiettisak, D. Kitatron, S. Techateerapreda, (2017), Rice Logistics and Supply Chain Management in Nakhon Ratchasima Province: Current Situation, Relations, Problems and Development Guidelines, The Suranaree Journal of Social Science (SJSS), Vol. 11, No. 2.

[4] พยอม โคเบลลี่ และ ธีรดา หวังสมบูรณ์ดี, (2559), โรคขอบใบแห้งของข้าวในประเทศไทย : สถานการณ์การระบาดของโรคปัจจุบัน, Unisearch Journal, Vol. 4 (2017), Iss. 1, หน้า 23-27.

[5] ฐิติ เติมเศรษฐเจริญ, (2561), การใช้สารกระตุ้นชีวภาพไคโตซานในการเพิ่มคุณภาพของเมล็ดข้าว, วิทยานิพนธ์, วิทยาศาสตรมหาบัณฑิต, จุฬาลงกรณ์มหาวิทยาลัย.

[6] S. Seneviratne, P. Jeyanandarajah, (2010), Rice diseases - problems and progress, Tropical Agricultural Research and Extension, vol. 7, pp. 29-48.

[7] T. Mew, N. Castilla, C. V. Cruz, (2001), The Etiology of Red Stripe of Rice: Current Status and Future Directions, International Rice Research Notes, Available at https://www.semanticscholar.org/paper/The-Etiology-of-Red-Stripe-of-Rice%3A-Current-Status-Mew-Castilla/ebe9e9dc047163cf4c22359cd1c2551bfc5b9199

[8] B. Dhan, (2015), Incidence and Severity of Brown Spot (BS) and Bacterial Leaf Blight (BLB) in Hybrid and Inbreed Rice Varieties in Bangladesh, Agricultural and Food Sciences, Available at https://www.semanticscholar.org/paper/Incidence-and-Severity-of-Brown-Spot-(-BS-)-and-(-)-Dhan/9f761b8e972e5e8d34ea0c01507610cabff0587b

[9] M. Kumar, A. Kumar, P. Shukla, A. K. Mishra, A. Kumar, (2023), Biology of Rice Bacterial Brown Stripe Pathogen and Integrated Strategies for Its Management, Journal of Experimental Agriculture International, vol. 45, Iss. 1, pp. 1-8.

[10] A. E. Asibi, Q. Chai, J. A. Coulter, (2019), Rice Blast: A Disease with Implications for Global Food Security, Agronomy, vol. 9, Iss. 8.

[11] K. Simkhada, R. Thapa, (2022), Rice Blast, A Major Threat to the Rice Production and its Various Management Techniques, Turkish Journal of Agriculture - Food Science and Technology, vol. 10, no. 2.

[12] K. Nagendran, G. Karthikeyan, M. F. Peeran, M. Raveendran, K. Prabakar, T. Raguchander, (2013), Management of Bacterial Leaf Blight Disease in Rice with Endophytic Bacteria, Agricultural and Food Sciences, Available at https://www.semanticscholar.org/paper/Management-of-Bacterial-Leaf-Blight-Disease-in-Rice-Nagendran-Karthikeyan/deb7b0b0d5ffaf310bf61ba9551378a7e30bf712

[13] G. Foreman, B. Hudelson, (2020), Bacterial blight, PlantwisePlus Knowledge Bank. Available at https://plantwiseplusknowledgebank.org/doi/10.1079/pwkb.20167800210

[14] K. Heong, N. Ho, (2019), Farmers' Perceptions Of The Rice Tungro Virus Problem In The Muda Irrigation Scheme, Malaysia, Management of Pests and Pesticides, 1st Edition, CRC Press.

[15] G. Kumar, F. Zarreen, I. Dasgupta, (2020), Rice Tungro Disease (Secoviridae, Caulimoviridae), Encyclopedia of Virology (Fourth Edition), Vol. 3, pp. 667-674.

[16] T. Mew, N. Castilla, C. V. Cruz, (2001), The Etiology of Red Stripe of Rice: Current Status and Future Directions, International Rice Research Notes, Available at https://www.semanticscholar.org/paper/The-Etiology-of-Red-Stripe-of-Rice%3A-Current-Status-Mew-Castilla/ebe9e9dc047163cf4c22359cd1c2551bfc5b9199

[17] K. Riangwong, W. Aesomnuk, Y. Sonsom, M. Siangliw, J. Unartngam, T. Toojinda, S. Wanchana, S. Arikit, (2023), QTL-seq Identifies Genomic Regions Associated with Resistance to Dirty Panicle Disease in Rice, Agronomy, vol. 13, no. 7.

[18] P.O. Williamson, C. Minter, (2019), Exploring PubMed as a reliable resource for scholarly communications services, Journal of the Medical Library Association, vol. 17, no. 1.

[19] B. Gülmez, (2024), Advancements in rice disease detection through convolutional neural networks: A comprehensive review, Heliyon, vol. 10, Iss. 12.

[20] T. Bera, Ankur Das, J. Sil, A. Das, (2018), A Survey on Rice Plant Disease Identification Using Image Processing and Data Mining Techniques, Advances in Intelligent Systems and Computing, Emerging Technologies in Data Mining and Information Security.

[21] M.M.F. Azizi, H. Y. Lau, (2022), Advanced diagnostic approaches developed for the global menace of rice diseases: a review, Canadian journal of plant pathology, pp. 627-651.

[22] M. Agrawal, S. Agrawal, (2020), Rice Plant Diseases Detection & Classification using Deep Learning Models: A Systematic Review, Available at https://www.semanticscholar.org/paper/RICE-PLANT-DISEASES-DETECTION-%26-CLASSIFICATION-DEEP-Agrawal-Agrawal/bd18a1e6b8078299693046e6ea94246b9fe6d896

[23] M. Younas, G. Wang, H. Du, Y. Zhang, I. Ahmad, N. Rajput, M. Li, Z. Feng, K. Hu, N.U. Khan, W. Xie, M. Qasim, Z. Chen, S. Zuo, (2023), Approaches to Reduce Rice Blast Disease Using Knowledge from Host Resistance and Pathogen Pathogenicity, nt. J. Mol. Sci., vol. 24, no. 5.

[24] R. Manavalan, (2022), Towards a Highly Intelligent Image Processing Techniques for Rice Diseases Identification: A Review, Current Chinese Computer Science, Vol. 2, Iss. 1.

[25] S. Aggarwal, M. Suchithra, N. Chandramouli, M. Sarada, Amit Verma, D. Vetrithangam, B. Pant, B.A. Adugna, (2023), Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improvise Agro-Business, Scientific Programming.

[26] E. E. Helliwell, Y. Yang, (2013), Molecular strategies to improve rice disease resistance, Methods in molecular biology, Available at https://doi.org/10.1007/978-1-62703-194-3_21

[27] M. M. F. Azizi, H. Y. Lau, (2022), Advanced diagnostic approaches developed for the global menace of rice diseases: a review, Canadian journal of plant pathology, Available at https://doi.org/10.1080/07060661.2022.2053588

[28] A. Tabassum, R. R. Patil, (2020), A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing, Available at

https://www.semanticscholar.org/paper/A-Survey-on-Text-Pre-Processing-%26-Feature-in-Tabassum-Patil/f308488e996599115fe478c03b74a0b19b9a8f06

[29] D. A. Naik, S. Mythreyan, S. Seema, (2023), Relevance Feature Discovery in Text Mining Using NLP, 2022 3rd International Conference for Emerging Technology (INCET).

[30] M. Kunilovskaya, A. Plum, (2021), Text Preprocessing and its Implications in a Digital Humanities Project, Recent Advances in Natural Language Processing, Available at https://doi.org/10.26615/issn.2603-2821.2021_013

[31] T. Madhulatha, (2012), An Overview on Clustering Methods, arXiv.org, Available at https://doi.org/10.9790/3021-0204719725

[32] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, (2002), An Efficient k-Means Clustering Algorithm: Analysis and Implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Available at https://doi.org/10.1109/TPAMI.2002.1017616

[33] S. Yue-heng, (2005), Research on text hierarchical clustering algorithm based on K-Means, Avaialble at https://www.semanticscholar.org/paper/Research-on-text-hierarchical-clustering-algorithm-Yue-heng/f78e816c87674dd4a4014e98acee5e81f7331b96

[34] J. Yu, B. Bohnet, M. Poesio, (2020), Named Entity Recognition as Dependency Parsing, Annual Meeting of the Association for Computational Linguistics, Available at https://doi.org/10.18653/v1/2020.acl-main.577

[35] H. L. Chieu, H. Ng, (2003), Named Entity Recognition with a Maximum Entropy Approach, Conference on Computational Natural Language Learning, Available at https://doi.org/10.3115/1119176.1119199

[36] H. Jelodar, Y. Wang, C. Yuan, X. Feng, (2017), Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey, Multimedia tools and applications, Available at https://doi.org/10.1007/s11042-018-6894-4

[37] E. S. Negara, D. Triadi, R. Andryani, (2019), Topic Modelling Twitter Data with Latent Dirichlet Allocation Method, 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), Available at https://doi.org/10.1109/ICECOS47637.2019.8984523

[38] K. Shahapure, C. K. Nicholas, (2020), Cluster Quality Analysis Using Silhouette Score, International Conference on Data Science and Advanced Analytics, Available at https://doi.org/10.1109/DSAA49011.2020.00096

[39] R. Yacouby, D. Axman, (2020), Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models, EVAL4NLP, Available at https://doi.org/10.18653/v1/2020.eval4nlp-1.9

[40] K. Adebayo, L.D. Caro, G. Boella, (2016), Text Segmentation with Topic Modeling and Entity Coherence, International Conference on Health Information Science, Available at https://doi.org/10.1007/978-3-319-52941-7_18

[41] B. Gülmez, (2024), Advancements in rice disease detection through convolutional neural networks: A comprehensive review, Available at https://doi.org/10.1016/j.heliyon.2024.
e33328

[42] A. Arya, P.K. Mishra, (2023), A Comprehensive Review: Advancements in Pretrained and Deep Learning Methods in the Disease Detection of Rice Plants, Available at https://doi.org/10.36548/jaicn.2023.3.003

[43] S. Parveen, Savita, S. Ganguly, AI for Agro-Business in the Identification of Rice Diseases, 2024 IEEE International Conference on Computing, Power and Communication Technologies

[44] M.M.F. Azizi, H.Y. Lau, (2022), Advanced diagnostic approaches developed for the global menace of rice diseases: a review, Canadian journal of plant pathology, Available at https://doi.org/10.1080/07060661.2022.2053588

[45] R. Chbeir, A. Kawtrakul, D. Laurent, N. Spyratos, (2012), DiseaseMedia: An Information System for Helping Diagnosing and Treating Rice Diseases, International Symposium on Information Processing.

[46] R. Manavalan, (2022), Towards a Highly Intelligent Image Processing Techniques for Rice Diseases Identification: A Review, Current Chinese Computer Science, Available at https://doi.org/10.2174/2665997202666220608125036

[47] T. Calinski, J. Harabasz, (1974), A dendrite method for cluster analysis, Communications in Statistics, Theory and Methods, 3(1), 1–27

[48] D.L. Davies, D.W. Bouldin, (1979), A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2), 224–227

[49] I. Beltagy, K. Lo, & A. Cohan, (2019), SciBERT: A Pretrained Language Model for Scientific Text, arXiv preprint arXiv:1903.10676

[50] L. McInnes, J. Healy, & J. Melville, (2018), UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, arXiv preprint arXiv:1802.03426

[51] P. Venugopal, "Understanding K-Means Clustering," Medium, June. 25, 2020. [Online]. Available: https://medium.com/@pranav3nov/understanding-k-means-clustering-f5e2e84d2129.

[52] F. Omarzai, "Hierarchical clustering in-depth," Medium, July. 20, 2024. [Online]. Available: https://medium.com/@fraidoonomarzai99/hierarchical-clustering-in-depth-d5f71c8522d4.

[53] A. Devlin, J. Chang, K. Lee, and M. Toutanova, "The overall structure of the BERT model," ResearchGate, Mar. 2022. [Online]. Available: https://www.researchgate.net/figure/The-overall-structure-of-the-BERT-model_fig1_359301499.

ภาคผนวก

ภาคผนวก ก

[คลิกที่นี่เพื่อเริ่มพิมพ์รายละเอียด]

ภาคผนวก ข

[คลิกที่นี่เพื่อเริ่มพิมพ์รายละเอียด]

บทความวิจัย

โปสเตอร์โครงงาน

ประวัติย่อผู้จัดทำโครงงาน

ประวัติย่อผู้จัดทำโครงงาน

ประวัติย่อผู้จัดทำโครงงานคนที่ 1

ชื่อ ชื่อสกุล [ชื่อผู้จัดทำ]

วัน เดือน ปีเกิด วันที่ "[วันที่ เดือน ปี พ.ศ.ผู้จัดทำ]"

สถานที่เกิด อำเภอ[ชื่ออำเภอ] จังหวัด[ชื่อจังหวัด]

ที่อยู่ที่สามารถติดต่อได้ "[ระบุที่อยู่ปัจจุบันของผู้จัดทำ ที่สามารถติดต่อได้]"

โทรศัพท์มือถือ [ระบุเบอร์โทรศัพท์มือถือ]

อีเมล์ [ระบุอีเมล์]

ประวัติการศึกษา พ.ศ. [ปีพ.ศ.] [ระดับการศึกษาปัจจุบันที่สำเร็จ]
[อักษรย่อวุฒิการศึกษา] [สาขาวิชา] [สถาบัน]
(ประวัติการศึกษา ควรระบุตั้งแต่ มัธยมศึกษาตอนต้น เป็นต้นไป)

ประวัติย่อผู้จัดทำโครงงานคนที่ 2

ชื่อ ชื่อสกุล [ชื่อผู้จัดทำ]

วัน เดือน ปีเกิด วันที่ "[วันที่ เดือน ปี พ.ศ.ผู้จัดทำ]"

สถานที่เกิด อำเภอ[ชื่ออำเภอ] จังหวัด[ชื่อจังหวัด]

ที่อยู่ที่สามารถติดต่อได้ "[ระบุที่อยู่ปัจจุบันของผู้จัดทำ ที่สามารถติดต่อได้]"

โทรศัพท์มือถือ [ระบุเบอร์โทรศัพท์มือถือ]

อีเมล์ [ระบุอีเมล์]

ประวัติการศึกษา พ.ศ. [ปีพ.ศ.] [ระดับการศึกษาปัจจุบันที่สำเร็จ]
[อักษรย่อวุฒิการศึกษา] [สาขาวิชา] [สถาบัน]
(ประวัติการศึกษา ควรระบุตั้งแต่ มัธยมศึกษาตอนต้น เป็นต้นไป)