แบบเสนอโครงงานพิเศษ (ปริญญานิพนธ์)

สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย
ภาควิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีและการจัดการอุตสาหกรรม

1. ข้อมูลขั้นต้นของโครงงาน

1.1 ชื่อโครงงาน

(ภาษาไทย) **แพลตฟอร์มวิศวกรรมความรู้เชิงสัมพันธ์การปรากฏร่วม**
(ภาษาอังกฤษ) **Co-Occurrence Knowledge Engineering Platform**

1.2 ชื่อนักศึกษาผู้ทำโครงงาน

ชื่อ-นามสกุล **นายยงยุทธ ชวนขุนทด**
สาขาวิชา **วิศวกรรมสารสนเทศและเครือข่าย**
ภาควิชา **เทคโนโลยีสารสนเทศ**
ภาคเรียนที่ 1
ปีการศึกษา 2568

1.3 ชื่ออาจารย์ที่ปรึกษา

ชื่อ-นามสกุล **รองศาสตราจารย์ ดร. อนิราช มิ่งขวัญ**

2. รายละเอียดของโครงงาน

2.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคดิจิทัลปัจจุบัน ข้อมูลและความรู้ที่เกิดขึ้นในแต่ละวันมีปริมาณที่เพิ่มขึ้นอย่างรวดเร็ว โดยเฉพาะอย่างยิ่งเอกสารทางวิชาการ หนังสือ และงานวิจัยต่าง ๆ ที่มีการเผยแพร่ในรูปแบบดิจิทัล เช่น ไฟล์ PDF ซึ่งเป็นแหล่งความรู้ที่มีคุณค่าสูง อย่างไรก็ตาม การจัดการ การวิเคราะห์ และการค้นหาความเชื่อมโยงระหว่างความรู้จากเอกสารเหล่านี้ยังคงเป็นปัญหาที่ท้าทาย

ปัญหาหลักที่พบในปัจจุบันคือ การที่ผู้ใช้งานไม่สามารถมองเห็นภาพรวมของความสัมพันธ์และความเชื่อมโยงระหว่างแนวคิดต่าง ๆ ที่ปรากฏในเอกสารหลายฉบับได้อย่างชัดเจน การอ่านและทำความเข้าใจเอกสารแต่ละฉบับแยกกันทำให้เกิดการสูญเสียโอกาสในการค้นพบความรู้ใหม่ที่อาจเกิดขึ้นจากการเชื่อมโยงข้อมูลจากแหล่งที่แตกต่างกัน

นอกจากนี้ การวิเคราะห์ความถี่ของการใช้คำและการปรากฏร่วมของความรู้ **(Co-occurrence)** ในเอกสารยังเป็นกระบวนการที่ซับซ้อนและใช้เวลามาก หากต้องทำด้วยมือหรือเครื่องมือพื้นฐาน ทำให้การสกัดความรู้และการสร้างความเข้าใจเชิงลึกจากเอกสารเป็นไปได้ยาก

ด้วยเหตุนี้ จึงจำเป็นต้องมีระบบที่สามารถแปลงเอกสารจากแหล่งต่าง ๆ ให้กลายเป็นกราฟเครือข่าย **(Network Graph)** ที่แสดงความสัมพันธ์และความเชื่อมโยงระหว่างแนวคิดได้อย่างชัดเจน รวมถึงสามารถสกัดส่วนของกราฟเพื่อนำไปผสมผสานกับข้อมูลจากแหล่งอื่น ๆ เพื่อสร้างความรู้ใหม่และค้นพบความเป็นไปได้ที่ไม่เคยมีมาก่อน ซึ่งจะช่วยเพิ่มประสิทธิภาพในการจัดการความรู้และส่งเสริมการเกิดนวัตกรรมใหม่ ๆ ในอนาคต

2.2 วัตถุประสงค์ของโครงงาน

2.2.1 เพื่อพัฒนาแพลตฟอร์มวิศวกรรมความรู้เชิงสัมพันธ์การปรากฏร่วมที่สามารถแปลงเอกสารให้กลายเป็นกราฟเครือข่ายความรู้ได้อย่างมีประสิทธิภาพ

2.2.2 เพื่อพัฒนาระบบวิเคราะห์การปรากฏร่วมของคำและแนวคิด (Co-occurrence Analysis) ที่สามารถระบุความถี่และความสัมพันธ์ระหว่างคำศัพท์ในเอกสารได้อย่างแม่นยำ

2.2.3 เพื่อสร้างส่วนติดต่อผู้ใช้ (User Interface) ที่ช่วยให้ผู้ใช้สามารถแสดงผลและโต้ตอบกับกราฟเครือข่ายความรู้ได้อย่างง่ายดายและเข้าใจง่าย

2.2.4 เพื่อพัฒนาฟีเจอร์การจัดการส่วนของกราฟ (Graph Management) เพื่อนำไปใช้ในการสร้างกราฟเครือข่ายใหม่หรือผสมผสานกับข้อมูลจากแหล่งอื่น

2.2.5 เพื่อพัฒนาระบบการผสมผสานความรู้จากหลายแหล่งข้อมูลเพื่อค้นหาความเชื่อมโยงและสร้างความรู้ใหม่ที่มีความเชื่อมโยงกัน

2.2.6 เพื่อพัฒนาระบบบูรณาการกับ Large Language Models (LLM) ที่สามารถใช้ข้อมูลจากกราฟเครือข่ายในการปรับปรุงความแม่นยำและประสิทธิภาพของการค้นหาและสกัดความรู้

2.3 ขอบเขตของการทำโครงงานพิเศษ (Scope of Special Project)

2.3.1 การพัฒนาระบบประมวลผลและวิเคราะห์เอกสาร เช่น PDF ที่สามารถดึงข้อความ วิเคราะห์โครงสร้างและแยกแยะเนื้อหาสำคัญจากเอกสารได้อย่างมีประสิทธิภาพ รวมถึงการจัดการกับรูปแบบการจัดวางข้อความและภาษาที่หลากหลาย

2.3.2 การพัฒนาระบบวิเคราะห์การปรากฏร่วม (Co-occurrence Analysis) ที่สามารถ

- 2.3.2.1 วิเคราะห์ความถี่ของคำและวลีในเอกสาร
- 2.3.2.2 ระบุความสัมพันธ์และการปรากฏร่วมของแนวคิดต่าง ๆ
- 2.3.2.3 คำนวณค่าความแข็งแกร่งของความเชื่อมโยงระหว่างคำหรือแนวคิด
- 2.3.2.4 สร้างเมทริกซ์ความสัมพันธ์สำหรับการสร้างกราฟเครือข่าย

2.3.3 การพัฒนาระบบสร้างและจัดการกราฟเครือข่ายความรู้ (Knowledge Network Graph) ที่สามารถ

- 2.3.3.1 แปลงข้อมูลการวิเคราะห์ให้เป็นโครงสร้างกราฟ
- 2.3.3.2 จัดกลุ่มและจัดระเบียบโหนดและขอบเชื่อมตามความสัมพันธ์
- 2.3.3.3 คำนวณคุณสมบัติของกราฟ เช่น ความหนาแน่น ความเป็นศูนย์กลาง
- 2.3.3.4 สนับสนุนการแสดงผลแบบ Interactive และ Dynamic

2.3.4 การพัฒนาส่วนติดต่อผู้ใช้ (User Interface) ที่มีคุณสมบัติ

- 2.3.4.1 อัปโหลดและจัดการไฟล์ PDF
- 2.3.4.2 แสดงผลกราฟเครือข่ายแบบโต้ตอบได้
- 2.3.4.3 เครื่องมือการค้นหาและกรองข้อมูล

2.3.4.4 ส่งออกผลลัพธ์ในรูปแบบต่าง ๆ (รูปภาพ, JSON, CSV)

2.3.5 การพัฒนาฟีเจอร์การสกัดและจัดการส่วนของกราฟ (Graph Management) ที่สามารถ

2.3.5.1 เลือกและสกัดส่วนที่สนใจจากกราฟใหญ่

2.3.5.2 บันทึกและจัดเก็บส่วนกราฟที่สกัดไว้

2.3.5.3 ผสมผสานกราฟจากหลายแหล่งข้อมูล

2.3.5.4 สร้างกราฟใหม่จากการรวมข้อมูลหลายเอกสาร

2.3.6 การพัฒนาระบบฐานข้อมูลสำหรับจัดเก็บข้อมูล

2.3.6.1 ข้อมูลเอกสารและเมตาดาต้า

2.3.6.2 ผลการวิเคราะห์และกราฟเครือข่าย

2.3.6.3 ประวัติการทำงานและการแก้ไข

2.3.6.4 การตั้งค่าและ Preferences ของผู้ใช้

2.3.7 การพัฒนาระบบบูรณาการกับ Large Language Models (LLM) ที่สามารถ

2.3.7.1 แปลงข้อมูล Network Graph ให้เป็นรูปแบบที่ LLM สามารถเข้าใจและประมวลผลได้

2.3.7.2 สร้างระบบ Query Interface ที่ช่วยให้ผู้ใช้สามารถสอบถามข้อมูลจากกราฟผ่าน LLM ได้

2.3.7.3 พัฒนา Context-aware Search ที่ใช้ความรู้จากกราฟเครือข่ายในการปรับปรุงความแม่นยำของการค้นหา

2.3.7.4 สร้างระบบ Knowledge Discovery ที่ใช้ LLM ในการวิเคราะห์และสกัดความรู้ใหม่จากความสัมพันธ์ในกราฟ

2.4 รายละเอียดของทฤษฎีที่ใช้ในการจัดทำปริญญานิพนธ์

2.4.1 สมมติฐาน หรือ ข้อตกลงเบื้องต้นในการจัดทำโครงงานพิเศษ (Assumption of the Study)

2.4.1.1 เอกสารที่นำเข้าสู่ระบบจะอยู่ในรูปแบบ PDF เป็นต้น ฯลฯ ที่มีข้อความที่สามารถสกัดได้ (Text-extractable) และ มี คุณภาพ เพียง พอ สำหรับ การ ประมวล ผล ด้วย เทคนิค Optical Character Recognition (OCR) ในกรณีที่จำเป็น

2.4.1.2 เอกสารที่ใช้ในการวิเคราะห์จะเป็นเอกสารทางวิชาการ งานวิจัย หรือเอกสารที่มีโครงสร้างและเนื้อหาที่ชัดเจน โดยมีการใช้คำศัพท์และแนวคิดที่สามารถระบุและวิเคราะห์ความสัมพันธ์ได้

2.4.1.3 ระบบจะทำงานภายใต้สมมติฐานที่ว่าการปรากฏร่วมของคำหรือแนวคิดในระยะทางที่ใกล้กันภายในเอกสารแสดงถึงความสัมพันธ์หรือความเชื่อมโยงระหว่างแนวคิดเหล่านั้น

2.4.1.4 ผู้ใช้งานระบบจะมีความรู้พื้นฐานในการตีความและวิเคราะห์กราฟเครือข่าย รวมถึงสามารถระบุความสำคัญและความหมายของความสัมพันธ์ที่แสดงในกราฟได้

2.4.1.5 ระบบจะมีประสิทธิภาพในการประมวลผลเอกสารที่มีขนาดปานกลางถึงใหญ่ โดยสมมติว่าทรัพยากรคอมพิวเตอร์ที่ใช้งานมีความสามารถเพียงพอสำหรับการประมวลผลและการแสดงผลกราฟเครือข่ายแบบโต้ตอบได้

2.4.1.6 การบูรณาการกับ Large Language Models (LLM) จะช่วยปรับปรุงความแม่นยำในการระบุและจัดกลุ่มแนวคิด โดยสมมติว่า LLM จะสามารถเข้าใจบริบทและความหมายของข้อความในเอกสารได้อย่างถูกต้อง

2.4.1.7 ผลลัพธ์ที่ได้จากระบบจะมีความเชื่อถือได้และสามารถนำไปใช้ในการตัดสินใจหรือการวิจัยเพิ่มเติมได้ โดยระบบจะมีกลไกในการตรวจสอบและปรับปรุงความแม่นยำของผลการวิเคราะห์

2.4.1.8 การผสมผสานข้อมูลจากหลายแหล่งจะช่วยสร้างความรู้ใหม่ที่มีคุณค่า โดยสมมติว่าข้อมูลจากแหล่งต่าง ๆ จะมีความเข้ากันได้และสามารถรวมเข้าด้วยกันได้อย่างมีความหมาย

2.4.2 คำจำกัดความ (Key Word)

2.4.2.1 **การปรากฏร่วม (Co-occurrence)** หมายถึง การที่คำหรือแนวคิดสองคำหรือมากกว่าปรากฏในตำแหน่งที่ใกล้เคียงกันภายในเอกสารหรือข้อความ ซึ่งแสดงถึงความสัมพันธ์หรือความเชื่อมโยงระหว่างแนวคิดเหล่านั้น

2.4.2.2 **วิศวกรรมความรู้ (Knowledge Engineering)** หมายถึง กระบวนการในการสกัด จัดระเบียบ จัดเก็บ และจัดการความรู้จากแหล่งข้อมูลต่าง ๆ เพื่อให้สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพ

2.4.2.3 **แพลตฟอร์ม (Platform)** หมายถึง ระบบซอฟต์แวร์ที่ให้บริการและเครื่องมือครบครันสำหรับการดำเนินงานเฉพาะด้าน ในที่นี้คือการวิเคราะห์และจัดการความรู้เชิงสัมพันธ์

2.4.2.4 **กราฟเครือข่าย (Network Graph)** หมายถึง โครงสร้างข้อมูลที่ประกอบด้วยโหนด (Nodes) และขอบเชื่อม (Edges) ที่แสดงความสัมพันธ์ระหว่างแนวคิดหรือเอนทิตีต่าง ๆ ในรูปแบบที่เข้าใจได้ง่าย

2.4.2.5 **การวิเคราะห์การปรากฏร่วม (Co-occurrence Analysis)** หมายถึง เทคนิคการวิเคราะห์ข้อมูลที่ใช้ในการระบุและวัดความถี่ของการปรากฏร่วมของคำหรือแนวคิดในข้อความ เพื่อค้นหาความสัมพันธ์และรูปแบบต่าง ๆ

2.4.2.6 **โหนด (Node)** หมายถึง จุดหรือองค์ประกอบพื้นฐานในกราฟเครือข่ายที่แทนคำ แนวคิด หรือเอนทิตีต่าง ๆ ที่ได้จากการวิเคราะห์เอกสาร

2.4.2.7 **ขอบเชื่อม (Edge)** หมายถึง เส้นเชื่อมระหว่างโหนดในกราฟเครือข่ายที่แสดงความสัมพันธ์หรือความเชื่อมโยงระหว่างแนวคิดหรือเอนทิตีต่าง ๆ รวมถึงค่าน้ำหนักที่บ่งบอกถึงความแข็งแกร่งของความสัมพันธ์

2.4.2.8 **การสกัดข้อความ (Text Extraction)** หมายถึง กระบวนการในการดึงข้อความจากเอกสารดิจิทัล เช่น ไฟล์ PDF โดยใช้เทคนิคการประมวลผลเอกสารหรือ OCR (Optical Character Recognition)

2.4.2.9 **เมทริกซ์ความสัมพันธ์ (Relationship Matrix)** หมายถึง ตารางสองมิติที่แสดงค่าความแข็งแกร่งของความสัมพันธ์ระหว่างคำหรือแนวคิดต่าง ๆ ที่ได้จากการวิเคราะห์การปรากฏร่วม

2.4.2.10 **Large Language Models (LLM)** หมายถึง โมเดลปัญญาประดิษฐ์ที่ได้รับการฝึกฝนด้วยข้อมูลข้อความขนาดใหญ่ เพื่อให้สามารถเข้าใจและประมวลผลภาษาธรรมชาติได้อย่างมีประสิทธิภาพ

2.4.2.11 **การแสดงผลแบบโต้ตอบ (Interactive Visualization)** หมายถึง การแสดงผลข้อมูลในรูปแบบที่ผู้ใช้สามารถโต้ตอบและปรับเปลี่ยนมุมมองหรือรายละเอียดได้ตามต้องการ

2.4.2.12 **การจัดการส่วนของกราฟ (Graph Management)** หมายถึง ระบบที่ช่วยในการเลือก สกัด บันทึก และจัดการส่วนต่าง ๆ ของกราฟเครือข่ายเพื่อนำไปใช้งานหรือวิเคราะห์เพิ่มเติม

2.4.2.13 **การค้นพบความรู้ (Knowledge Discovery)** หมายถึง กระบวนการในการค้นหาและระบุรูปแบบความสัมพันธ์ หรือความรู้ใหม่ที่ซ่อนอยู่ในข้อมูลขนาดใหญ่ผ่านเทคนิคการวิเคราะห์ข้อมูลต่าง ๆ

2.4.3 รายงานการค้นคว้า การศึกษา หรือการวิจัยที่เกี่ยวข้อง

2.4.3.1 **Centroid Terms as Text Representatives**

งาน วิจัย ของ Mario M. Kubek, Herwig Unger (2016) เรื่อง "Centroid Terms as Text Representatives" ได้ศึกษาเทคนิคการสร้างกราฟความรู้จากข้อความ โดยเฉพาะการใช้เทคนิค Co-occurrence Analysis ในการระบุความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ การศึกษานี้ได้ชี้ให้เห็นว่าอัลกอริทึมสำหรับการจัดกลุ่มและการจำแนกข้อความตามหัวข้อนั้นอาศัยข้อมูลเกี่ยวกับระยะทางและความคล้ายคลึงเชิงความหมาย วิธีการมานตรฐานที่อิงตามแบบจำลอง bag-of-words ในการกำหนดข้อมูลนี้จะให้เพียงการประมาณแบบคร่าว ๆ เกี่ยวกับความเกี่ยวข้องของข้อความ นอกจากนี้วิธีการเหล่านี้ยังไม่สามารถค้นหาคำศัพท์ที่เป็นนามธรรมหรือคำที่สามารถอธิบายเนื้อหาของข้อความได้อย่างครอบคลุม งานวิจัยนี้จึงได้นำเสนอวิธีการใหม่ในการกำหนดคำศูนย์กลาง (Centroid Terms) ในข้อความและการประเมินความคล้ายคลึงโดยใช้คำที่เป็นตัวแทนเหล่านั้น ซึ่งแสดงให้เห็นว่าการวิเคราะห์การปรากฏร่วมสามารถช่วยในการค้นพบรูปแบบความสัมพันธ์ที่ซ่อนอยู่ในข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ และสามารถนำไปประยุกต์ใช้ในการพัฒนาระบบวิศวกรรมความรู้เชิงสัมพันธ์ได้

2.4.3.2 **Spreading activation: a fast calculation method for text centroids**

งาน วิจัย ของ Mario M. Kubek, Thomas Böhme, Herwig Unger (2017) เรื่อง "Spreading activation: a fast calculation method for text centroids" ได้นำเสนอเทคนิคการคำนวณ Centroid Terms แบบใหม่ที่มีประสิทธิภาพสูง โดยใช้ Spreading Activation Algorithm ซึ่งเป็นเทคนิคที่ทำงานบนพื้นฐานของกราฟและหลักการทำงานแบบเฉพาะที่ (Local Working Principle) การศึกษานี้ชี้ให้เห็นว่า Centroids เป็นเครื่องมือที่สะดวกในการแสดงคำค้นหาและข้อความทั้งหมดด้วยคำศัพท์เชิงบรรยายเพียงคำเดียว ซึ่งสามารถนำไปใช้ในการกำหนดความคล้ายคลึงของเนื้อหาข้อความและการจัดกลุ่มเอกสารแบบลำดับชั้น อย่างไรก็ตาม การคำนวณตามคำจำกัดความแบบดั้งเดิมอาจใช้เวลามากและเป็นอุปสรรคต่อการนำไปใช้งานจริง ดังนั้น การพัฒนาอัลกอริทึมแบบใหม่ที่อิงตามกราฟ Co-occurrence จึงมีความสำคัญต่อการเพิ่มประสิทธิภาพการประมวลผล ซึ่งสอดคล้องกับแนวทางที่จะใช้ในโครงงานนี้สำหรับการวิเคราะห์การปรากฏร่วมและการสร้างกราฟเครือข่ายความรู้ที่มีประสิทธิภาพในการประมวลผลเอกสารขนาดใหญ่

2.4.3.3 **Enhancing Retrieval-Augmented Generation Systems by Text-Representing Centroid**

การ ศึกษา ของ Yanakorn Ruamsuk, Anirach Mingkhawn, Herwig Unger (2025) เรื่อง "Enhancing Retrieval-Augmented Generation Systems by Text-Representing Centroid" ได้นำเสนอแนวทางใหม่ในการปรับปรุงระบบ Retrieval-Augmented Generation (RAG) โดยการบูรณาการเทคนิค Text-Representing Centroid (TRC) เพื่อแก้ไขข้อจำกัดของฐานข้อมูลเวกเตอร์แบบดั้งเดิม วิธีการนี้สามารถรักษาความสัมพันธ์เชิงโครงสร้างและปรับตัวตามความซับซ้อนของเนื้อหา ส่งผลให้การค้นคืนข้อมูลมีประสิทธิภาพและความแม่นยำที่สูงขึ้น การมีส่วนสนับสนุนที่สำคัญได้แก่ การสร้างกราฟขั้นสูง อัลกอริทึมการให้คะแนนความเกี่ยวข้อง และการตรวจสอบความถูกต้องอย่างครอบคลุม พร้อมการอภิปรายเกี่ยวกับการประยุกต์ใช้ที่เป็นไปได้และการวิจัยในอนาคต หลักฐานเชิงประจักษ์แสดงให้เห็นว่าเทคนิค TRC สามารถบรรลุอัตราความสำเร็จ 75 เปอร์เซ็นต์จากคำถาม 100 ข้อ ซึ่งมีประสิทธิภาพเหนือกว่าวิธีการเวกเตอร์แบบดั้งเดิม การศึกษานี้มีความเกี่ยวข้องโดยตรงกับโครงงานที่เสนอ เนื่องจากแสดงให้เห็นถึงความเป็นไปได้ในการใช้เทคนิค Co-occurrence Analysis และ Centroid-based Methods ในการพัฒนาระบบวิศวกรรมความรู้ที่มีประสิทธิภาพสูง

2.4.4 เนื้อหา เหตุผล และทฤษฎีที่สำคัญ

โครงงาน Co-Occurrence Knowledge Engineering Platform นี้มีพื้นฐานทางทฤษฎีที่แข็งแกร่งและเหตุผลที่ชัดเจนในการพัฒนา โดยอาศัยหลักการทางวิศวกรรมความรู้และเทคนิคการวิเคราะห์ข้อมูลขั้นสูงหลายแนวทาง

เหตุผลในการพัฒนาโครงงาน

ในยุคสารสนเทศปัจจุบัน ปริมาณข้อมูลและเอกสารดิจิทัลเพิ่มขึ้นอย่างรวดเร็ว แต่การจัดการและการค้นหาความเชื่อมโยงระหว่างความรู้จากแหล่งต่าง ๆ ยังคงเป็นความท้าทายสำคัญ ผู้ใช้งานส่วนใหญ่ไม่สามารถมองเห็นภาพรวมของความสัมพันธ์ระหว่างแนวคิดที่ปรากฏในเอกสารหลายฉบับได้อย่างชัดเจน การวิเคราะห์การปรากฏร่วม (Co-occurrence Analysis) แบบดั้งเดิมใช้เวลามากและซับซ้อน ดังนั้น จึงจำเป็นต้องมีระบบที่สามารถแปลงเอกสารให้เป็นกราฟเครือข่ายความรู้ที่เข้าใจได้ง่าย และสามารถผสมผสานข้อมูลจากหลายแหล่งเพื่อสร้างความรู้ใหม่

ทฤษฎีพื้นฐานที่ใช้ในการพัฒนา

1. ทฤษฎีการปรากฏร่วม (Co-occurrence Theory)

หลักการพื้นฐานของการวิเคราะห์การปรากฏร่วมอิงตามสมมติฐานที่ว่า คำหรือแนวคิดที่ปรากฏใกล้กันในข้อความมักจะมีความสัมพันธ์หรือความเชื่อมโยงกัน ทฤษฎีนี้ได้รับการสนับสนุนจากงานวิจัยของ Mario M. Kubek et al. (2016, 2017) ที่แสดงให้เห็นว่าการวิเคราะห์ Centroid Terms และ Spreading Activation Algorithm สามารถช่วยในการระบุความสัมพันธ์เชิงความหมายได้อย่างมีประสิทธิภาพ

2. ทฤษฎีกราฟและเครือข่าย (Graph Theory and Network Theory)

การแสดงความรู้ในรูปแบบกราฟเครือข่ายอิงตามทฤษฎีกราฟ ซึ่งประกอบด้วยโหนด (Nodes) แทนแนวคิดหรือเอนทิตี และขอบเชื่อม (Edges) แทนความสัมพันธ์ ทฤษฎีนี้ช่วยให้สามารถคำนวณคุณสมบัติต่าง ๆ ของเครือข่าย เช่น ความหนาแน่น (Density) ความเป็นศูนย์กลาง (Centrality) และการจัดกลุ่ม (Clustering) ซึ่งมีความสำคัญต่อการวิเคราะห์และการค้นพบความรู้

3. ทฤษฎีการประมวลผลภาษาธรรมชาติ (Natural Language Processing Theory)

การสกัดข้อความและการวิเคราะห์เนื้อหาจากเอกสาร PDF อาศัยหลักการของ NLP รวมถึงเทคนิค Tokenization, Part-of-Speech Tagging, Named Entity Recognition และ Semantic Analysis เพื่อให้สามารถระบุและแยกแยะแนวคิดสำคัญได้อย่างแม่นยำ

4. ทฤษฎีการเรียนรู้ของเครื่อง (Machine Learning และ Deep Learning Theory)

การบูรณาการกับ Large Language Models (LLM) อาศัยหลักการของ Deep Learning และ Transformer Architecture เพื่อปรับปรุงความแม่นยำในการระบุความสัมพันธ์และการทำ Semantic Reasoning งานวิจัยของ Yanakorn Ruamsuk et al. (2025) แสดงให้เห็นว่าการใช้ Text-Representing Centroid ร่วมกับ LLM สามารถบรรลุอัตราความสำเร็จ 75 เปอร์เซ็นต์ในการตอบคำถาม

5. ทฤษฎีการจัดการฐานข้อมูล (Database Management Theory)

การจัดเก็บและจัดการข้อมูลกราฟอาศัยหลักการของ Graph Database และ NoSQL Database เพื่อรองรับการประมวลผลข้อมูลเชิงสัมพันธ์ที่ซับซ้อนและการ Query แบบ Real-time

2.5 วิธีดำเนินการจัดทำโครงงานพิเศษ

การพัฒนาแพลตฟอร์ม Co-Occurrence Knowledge Engineering Platform จะดำเนินการโดยใช้แนวทางการพัฒนาระบบแบบครบวงจร (Full-Stack Development) ร่วมกับเทคโนโลยีคลาวด์และเครื่องมือออกแบบสมัยใหม่ เพื่อให้ได้ระบบที่มีประสิทธิภาพ ปลอดภัย และใช้งานง่าย

วิธีการดำเนินการหลัก

2.5.1 **การพัฒนาระบบบนคลาวด์เซิร์ฟเวอร์**

- 2.5.1.1 ใช้คลาวด์เซิร์ฟเวอร์เป็นสภาพแวดล้อมหลักในการ Host โครงงาน
- 2.5.1.2 ติดตั้งและกำหนดค่าระบบปฏิบัติการ Linux (Server) สำหรับการประมวลผล
- 2.5.1.3 ปรับแต่งสภาพแวดล้อมสำหรับการพัฒนา TypeScript, Go และฐานข้อมูล

2.5.2 **การออกแบบ User Interface และ User Experience**

- 2.5.2.1 ใช้เครื่องมือ Figma ในการออกแบบ Wireframes และ Mockups ทั้งหมด
- 2.5.2.2 สร้าง Design System ที่สอดคล้องกับการใช้งานของ Knowledge Engineering Platform
- 2.5.2.3 ออกแบบ Interactive Prototypes สำหรับการแสดงผลกราฟเครือข่าย
- 2.5.2.4 ทดสอบ Usability และปรับปรุงการออกแบบตามผลการทดสอบ
- 2.5.2.5 สร้าง Responsive Design เพื่อรองรับการใช้งานบนอุปกรณ์ต่าง ๆ

2.5.3 **การพัฒนาระบบจำลองและการทดสอบ**

- 2.5.3.1 สร้างระบบจำลองการวิเคราะห์ Co-occurrence ด้วยข้อมูลตัวอย่าง
- 2.5.3.2 พัฒนา Proof of Concept สำหรับการสร้างกราฟเครือข่ายจากเอกสาร PDF
- 2.5.3.3 สร้างและทดสอบการบูรณาการกับ Large Language Models (LLM)

2.5.4 **การสร้างเอกสารและข้อกำหนดระบบ**

- 2.5.4.1 จัดทำเอกสาร System Requirements และ Functional Specifications
- 2.5.4.2 สร้าง API Documentation สำหรับการใช้งานระบบ
- 2.5.4.3 เขียน User Manual และ Administrator Guide
- 2.5.4.4 จัดทำเอกสาร Technical Architecture และ Database Schema
- 2.5.4.5 สร้าง Test Cases และ Test Plans สำหรับการทดสอบระบบ

2.5.5 **การสร้างสภาพแวดล้อมความปลอดภัย**

- 2.5.5.1 ติดตั้งและกำหนดค่า VPN Server สำหรับการเชื่อมต่อที่ปลอดภัย
- 2.5.5.2 สร้าง Private Network สำหรับการเข้าถึงฐานข้อมูลและทรัพยากรภายใน
- 2.5.5.3 ปรับแต่งระบบ Firewall และ Security Groups สำหรับการควบคุมการเข้าถึง
- 2.5.5.4 ใช้ SSL/TLS Certificates สำหรับการเข้ารหัสข้อมูล

2.5.6 **การพัฒนาและการทดสอบระบบ**

- 2.5.6.1 สร้าง CI/CD Pipeline สำหรับการ Deploy และ Testing อัตโนมัติ
- 2.5.6.2 ทดสอบระบบด้วย Unit Testing, Integration Testing และ End-to-End Testing
- 2.5.6.3 ประเมินประสิทธิภาพระบบด้วย Performance Testing และ Load Testing
- 2.5.6.4 ทดสอบความปลอดภัยด้วย Security Testing และ Penetration Testing

สถานที่ดำเนินการ

2.5.7 **สถานที่หลักในการพัฒนา**

2.5.7.1 ห้องปฏิบัติการคอมพิวเตอร์ ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม สำหรับการพัฒนาและทดสอบระบบ

2.5.7.2 ห้องส่วนตัวที่มีการเชื่อมต่ออินเทอร์เน็ตความเร็วสูง สำหรับการพัฒนาและการออกแบบ

2.5.7.3 คลาวด์เซิร์ฟเวอร์ สำหรับการ Host และ Deploy ระบบ

2.5.8 **สภาพแวดล้อมการทำงาน**

2.5.8.1 ระบบพัฒนาบนเครื่อง Local Development Environment (macOS/Linux)

2.5.8.2 ระบบทดสอบบน Staging Environment ในคลาวด์เซิร์ฟเวอร์

2.5.8.3 ระบบจริงบน Production Environment ที่มีการรักษาความปลอดภัยสูง

2.5.8.4 ระบบการออกแบบบน Figma Cloud Platform

2.6 แผนกิจกรรมและตารางเวลาในการจัดทำ

2.6.1 แผนกิจกรรมหลักและระยะเวลา

ตารางที่ 1: แผนการดำเนินงานภาคการศึกษาที่ 1

ขั้นตอนการดำเนินการ	ก.ค.				ส.ค.				ก.ย.				ต.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1) วางแผนการพัฒนา																
2) เก็บ Requirement																
3) ออกแบบ UI และ UX																
4) ออกแบบสถาปัตยกรรมซอฟต์แวร์																
5) พัฒนาระบบส่วนแกนหลัก																

ตารางที่ 2: แผนการดำเนินงานภาคการศึกษาที่ 2

ขั้นตอนการดำเนินการ	ธ.ค.				ม.ค.				ก.พ.				มี.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
6) ทดสอบระบบส่วนแกนหลัก																
7) ทดสอบความปลอดภัยของระบบ																
8) ทดสอบสภาพแวดล้อมจริง																
9) จัดทำเอกสารแพลตฟอร์ม																
10) ส่งมอบแพลตฟอร์ม																

2.6.2 แผนภูมิขั้นตอนการจัดทำโครงงานพิเศษ โดยละเอียด

การพัฒนาแพลตฟอร์ม Co-Occurrence Knowledge Engineering Platform จะดำเนินการตามผังงานที่มีการแบ่งขั้นตอนและกิจกรรมอย่างชัดเจน พร้อมทั้งกำหนดอัตราส่วนของแต่ละกิจกรรมเป็นร้อยละเพื่อการติดตามความก้าวหน้าอย่างมีประสิทธิภาพ

ผังงานการดำเนินงาน (Development Flow Chart)

Phase 1: การวางแผนและวิเคราะห์ (Planning & Analysis Phase) - 15% ของโครงงานทั้งหมด

1.1 **การวางแผนการพัฒนา (5%)**

- 1.1.1 กำหนดขอบเขตโครงงานและวัตถุประสงค์
- 1.1.2 วิเคราะห์ความเป็นไปได้ทางเทคนิค (Technical Feasibility)
- 1.1.3 จัดทำ Project Charter และ Timeline
- 1.1.4 กำหนดทรัพยากรและงบประมาณ

1.2 **การเก็บความต้องการ (Requirements Gathering) (10%)**

- 1.2.1 วิเคราะห์ Functional Requirements
- 1.2.2 วิเคราะห์ Non-Functional Requirements
- 1.2.3 สร้าง Use Case Diagrams และ User Stories
- 1.2.4 กำหนด Acceptance Criteria สำหรับแต่ละฟีเจอร์
- 1.2.5 วิเคราะห์ข้อจำกัดและความเสี่ยง

Phase 2: การออกแบบระบบ (System Design Phase) - 25% ของโครงงานทั้งหมด

2.1 **การออกแบบ UI และ UX (12%)**

- 2.1.1 สร้าง User Persona และ User Journey Mapping
- 2.1.2 ออกแบบ Wireframes และ Mockups ด้วย Figma
- 2.1.3 สร้าง Interactive Prototypes
- 2.1.4 ทดสอบ Usability และปรับปรุงการออกแบบ
- 2.1.5 สร้าง Design System และ Component Library

2.2 **การออกแบบสถาปัตยกรรมซอฟต์แวร์ (13%)**

- 2.2.1 ออกแบบ System Architecture และ Component Diagram
- 2.2.2 ออกแบบ Database Schema และ Data Model
- 2.2.3 ออกแบบ API Architecture และ Endpoints
- 2.2.4 ออกแบบ Security Architecture และ Authentication
- 2.2.5 ออกแบบ Deployment Architecture และ Infrastructure
- 2.2.6 สร้าง Technical Specifications Document

Phase 3: การพัฒนาระบบ (Development Phase) - 40% ของโครงงานทั้งหมด

3.1 **การพัฒนาระบบส่วนแกนหลัก (Core System Development) (40%)**

3.1.1 **Backend Development (20%)**

3.1.1.1 สร้าง API สำหรับการอัปโหลดและประมวลผล PDF (4%)

3.1.1.2 พัฒนาระบบ Co-occurrence Analysis Engine (6%)

3.1.1.3 สร้างระบบ Graph Generation และ Management (5%)

3.1.1.4 พัฒนาระบบ Database และ Data Storage (3%)

3.1.1.5 บูรณาการกับ Large Language Models (LLM) (2%)

3.1.2 **Frontend Development (15%)**

3.1.2.1 สร้าง Dashboard และ Main Interface (4%)

3.1.2.2 พัฒนา File Upload และ Management System (3%)

3.1.2.3 สร้าง Interactive Graph Visualization (5%)

3.1.2.4 พัฒนา Search และ Filter Components (2%)

3.1.2.5 สร้าง Export และ Share Features (1%)

3.1.3 **Integration และ API Development (5%)**

3.1.3.1 Integration Testing ระหว่าง Frontend และ Backend (2%)

3.1.3.2 สร้าง RESTful API Documentation (1%)

3.1.3.3 พัฒนา Error Handling และ Logging System (1%)

3.1.3.4 Optimization และ Performance Tuning (1%)

Phase 4: การทดสอบระบบ (Testing Phase) - 15% ของโครงงานทั้งหมด

4.1 **การทดสอบระบบส่วนแกนหลัก (8%)**

4.1.1 Unit Testing สำหรับแต่ละ Component (2%)

4.1.2 Integration Testing สำหรับระบบทั้งหมด (3%)

4.1.3 Performance Testing และ Load Testing (2%)

4.1.4 Functional Testing และ User Acceptance Testing (1%)

4.2 **การทดสอบความปลอดภัยของระบบ (4%)**

4.2.1 Security Testing และ Vulnerability Assessment (2%)

4.2.2 Authentication และ Authorization Testing (1%)

4.2.3 Data Protection และ Privacy Testing (1%)

4.3 **การทดสอบสภาพแวดล้อมจริง (3%)**

4.3.1 Deployment Testing บน Production Environment (1%)

4.3.2 End-to-End Testing กับข้อมูลจริง (1%)

4.3.3 User Experience Testing และ Feedback Collection (1%)

Phase 5: การจัดทำเอกสารและส่งมอบ (Documentation & Delivery Phase) - 5% ของโครงงานทั้งหมด

5.1 **การจัดทำเอกสารแพลตฟอร์ม (3%)**

5.1.1 เขียน User Manual และ Administrator Guide (1%)

5.1.2 สร้าง API Documentation และ Technical Guide (1%)

5.1.3 จัดทำ Installation Guide และ Troubleshooting (0.5%)

5.1.4 สร้าง Video Tutorial และ Demo Materials (0.5%)

5.2 **การส่งมอบแพลตฟอร์ม (2%)**

5.2.1 Final System Testing และ Quality Assurance (0.5%)

5.2.2 Knowledge Transfer และ Training Session (0.5%)

5.2.3 Project Handover และ Final Presentation (0.5%)

5.2.4 Post-Implementation Support Planning (0.5%)

การติดตามความก้าวหน้า (Progress Monitoring)

การประเมินความก้าวหน้าจะดำเนินการโดยใช้ระบบ Milestone-based Tracking ซึ่งแบ่งการประเมินออกเป็น:

5.3 **Weekly Progress Review (รายสัปดาห์)**

5.3.1 ตรวจสอบความก้าวหน้าตาม Timeline ที่กำหนด

5.3.2 ประเมินคุณภาพของงานที่ส่งมอบในแต่ละ Phase

5.3.3 ระบุและแก้ไข Issues หรือ Blockers ที่เกิดขึ้น

5.3.4 ปรับปรุง Plan หากจำเป็น

5.4 **Phase Gate Reviews (รายเฟส)**

5.4.1 การประเมินผลลัพธ์ที่ได้จากแต่ละ Phase

5.4.2 การตรวจสอบคุณภาพตาม Acceptance Criteria

5.4.3 การอนุมัติให้ดำเนินการใน Phase ถัดไป

5.4.4 การจัดทำ Lessons Learned สำหรับปรับปรุงในอนาคต

สรุปการแบ่งสัดส่วนงาน

- การวางแผนและวิเคราะห์: 15%
- การออกแบบระบบ: 25%
- การพัฒนาระบบ: 40%
- การทดสอบระบบ: 15%
- การจัดทำเอกสารและส่งมอบ: 5%

2.7 ทรัพยากรที่ต้องใช้ในการจัดทำโครงงานพิเศษ

การพัฒนาแพลตฟอร์ม Co-Occurrence Knowledge Engineering Platform ต้องใช้ทรัพยากรหลากหลายประเภทเพื่อให้การดำเนินงานมีประสิทธิภาพและบรรลุวัตถุประสงค์ที่กำหนดไว้ โดยทรัพยากรเหล่านี้จะแบ่งออกเป็นหมวดหมู่ต่าง ๆ ตามลักษณะการใช้งานและความสำคัญต่อโครงงาน

ทรัพยากรหลัก (Primary Resources)

2.7.1 ทรัพยากรด้านฮาร์ดแวร์และโครงสร้างพื้นฐาน

2.7.1.1 เครื่องคอมพิวเตอร์สำหรับการพัฒนา

- 2.7.1.1.1 MacBook (CPU: Apple M2, RAM: 16GB ขึ้นไป, SSD: 256GB ขึ้นไป)
- 2.7.1.1.2 จอภาพเพิ่มเติมขนาด 24-27 นิ้ว สำหรับการแสดงผลและออกแบบ
- 2.7.1.1.3 อุปกรณ์รับเสียงและไมโครโฟนสำหรับการจัดทำวิดีโอการนำเสนอ

2.7.1.2 คลาวด์เซิร์ฟเวอร์และบริการ

- 2.7.1.2.1 Virtual Private Server (VPS) หรือ Cloud Instance สำหรับ Development และ Testing Environment
- 2.7.1.2.2 Production Server พร้อม Load Balancer สำหรับการใช้งานจริง
- 2.7.1.2.3 Content Delivery Network (CDN) สำหรับการกระจายข้อมูลและเพิ่มประสิทธิภาพ
- 2.7.1.2.4 SSL/TLS Certificates สำหรับการรักษาความปลอดภัย

2.7.1.3 ระบบฐานข้อมูลและการจัดเก็บข้อมูล

- 2.7.1.3.1 Graph Database (เช่น Neo4j, ArangoDB) สำหรับการจัดเก็บกราฟเครือข่าย
- 2.7.1.3.2 Document Database (MongoDB) สำหรับการจัดเก็บเอกสารและเมตาดาต้า
- 2.7.1.3.3 File Storage System (MinIO) สำหรับการจัดเก็บไฟล์ PDF และผลลัพธ์ที่ส่งออก
- 2.7.1.3.4 Cache Database (Redis) สำหรับการเพิ่มประสิทธิภาพการประมวลผล

2.7.2 ทรัพยากรด้านซอฟต์แวร์และเครื่องมือพัฒนา

2.7.2.1 เครื่องมือการพัฒนาและ IDE

- 2.7.2.1.1 Visual Studio Code สำหรับการเขียนโปรแกรม
- 2.7.2.1.2 Git และ GitHub สำหรับการจัดการเวอร์ชันโค้ด
- 2.7.2.1.3 Docker และ Docker Compose สำหรับการจัดการ Container
- 2.7.2.1.4 Kubernetes หรือ Docker Swarm สำหรับการ Orchestration

2.7.2.2 เครื่องมือการออกแบบและ UI/UX

- 2.7.2.2.1 Figma สำหรับการออกแบบ Interface, Flowchart และ Prototyping
- 2.7.2.2.2 Excalidraw สำหรับการสร้าง Diagram ทั่วไป

2.7.2.3 เครื่องมือการทดสอบและการจัดการคุณภาพ

- 2.7.2.3.1 Trivy สำหรับการวิเคราะห์ความปลอดภัยของ Container
- 2.7.2.3.2 Postman สำหรับการทดสอบ API
- 2.7.2.3.3 Performance Testing Tools (k6)

2.7.3 **ทรัพยากรด้านเทคโนโลยีและภาษาโปรแกรม**

2.7.3.1 **เทคโนโลยี Backend Development**

2.7.3.1.1 Go (Golang) สำหรับการพัฒนา High-Performance Backend Services

2.7.3.1.2 GraphQL สำหรับการสร้าง Flexible API

2.7.3.1.3 RESTful API Frameworks (Fiber for Go)

2.7.3.2 **เทคโนโลยี Frontend Development**

2.7.3.2.1 React.js หรือ Next.js สำหรับการสร้าง Interactive Web Interface

2.7.3.2.2 TypeScript สำหรับการเพิ่มความปลอดภัยของโค้ด

2.7.3.2.3 D3.js หรือ Vis.js สำหรับการสร้าง Interactive Graph Visualization

2.7.3.2.4 CSS Frameworks (Tailwind CSS, Material-UI) สำหรับการออกแบบ

2.7.3.3 **เทคโนโลยี Machine Learning และ NLP**

2.7.3.3.1 Python Libraries (NumPy, Pandas, Scikit-learn) สำหรับการวิเคราะห์ข้อมูล

2.7.3.3.2 Natural Language Processing Libraries (spaCy, NLTK, Transformers)

2.7.3.3.3 PDF Processing Libraries (PyPDF2, pdfplumber, Tesseract OCR)

2.7.3.3.4 Large Language Model APIs (OpenAI GPT, Google PaLM, Claude)

2.7.3.4 **เทคโนโลยี DevOps และ Infrastructure**

2.7.3.4.1 CI/CD Tools (GitHub Actions, GitLab CI, Jenkins)

2.7.3.4.2 Infrastructure as Code (Terraform, CloudFormation)

2.7.3.4.3 Monitoring และ Logging (Prometheus, Grafana, ELK Stack)

2.7.3.4.4 Container Orchestration (Kubernetes, Docker Swarm)

2.7.4 **ทรัพยากรด้านข้อมูลและการวิจัย**

2.7.4.1 **ข้อมูลสำหรับการทดสอบและพัฒนา**

2.7.4.1.1 เอกสาร PDF ตัวอย่างจากแหล่งวิชาการต่าง ๆ สำหรับการทดสอบระบบ

2.7.4.1.2 Dataset ข้อมูลงานวิจัยและเอกสารทางวิชาการจากแหล่งเปิด

2.7.4.1.3 Sample Co-occurrence Networks สำหรับการเปรียบเทียบผลลัพธ์

2.7.4.2 **แหล่งข้อมูลและการอ้างอิง**

2.7.4.2.1 การเข้าถึงฐานข้อมูลวิชาการ (IEEE Xplore, ACM Digital Library, SpringerLink)

2.7.4.2.2 เครื่องมือการจัดการ References (Zotero, Mendeley, EndNote)

2.7.4.2.3 Online Learning Resources และ Technical Documentation

2.7.4.2.4 การสมัครสมาชิก Professional Communities และ Forums

2.7.5 **ทรัพยากรด้านบุคลากรและการสนับสนุน**

2.7.5.1 **ทีมพัฒนาและที่ปรึกษา**

2.7.5.1.1 อาจารย์ที่ปรึกษาโครงงาน

2.7.6 **ทรัพยากรด้านการเงินและงบประมาณ**

2.7.6.1 **ค่าใช้จ่ายด้านโครงสร้างพื้นฐาน**

2.7.6.1.1 ค่า Cloud Server อยู่ที่ 0 บาท (ใช้ Cloud Server ของตนเอง จึงไม่นับค่าใช้จ่าย)

2.7.6.1.2 ค่าจัดทำปริญญานิพนธ์ ประมาณ 1,000 บาท