



เข็คความเหมือนของประโยชน์หรือข้อความ

Text similarity

จัดทำโดย

นางสาว พิมพ์ลักษณ์ สกุลเจริญกิจ 643020630-5

นาย พัชรพล พันทวี 683380431-8

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา SC351101 วิทยาการคำนวณ

ภาคเรียนที่ 2 ปีการศึกษา 2568

ภาควิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์

1. ความเป็นมาของโครงการ

เทคโนโลยีสารสนเทศและอินเทอร์เน็ตมีบทบาทสำคัญต่อการสื่อสารและการจัดการข้อมูล โดยเฉพาะข้อมูลในรูปแบบข้อความ (Text Data) ซึ่งมีปริมาณเพิ่มขึ้นอย่างรวดเร็ว เช่น บทความ ข่าว งานวิชาการ รีวิวสินค้า และโพสต์บนสื่อสังคมออนไลน์ การจัดการและวิเคราะห์ข้อมูลข้อความจำนวนมากตัวยิ่งการแบบดั้งเดิมจึงทำได้ยากและใช้เวลามาก

หนึ่งในปัญหาที่พบได้บ่อยคือการตรวจสอบความคล้ายคลึงกันของข้อความ (Text Similarity) เช่น การตรวจสอบการคัดลอกผลงาน (Plagiarism Detection) การจัดกลุ่มเอกสารที่มีเนื้อหาใกล้เคียงกัน การค้นหาข้อมูลที่เกี่ยวข้อง หรือการเปรียบเทียบคำตอบที่มีความหมายคล้ายกัน แม้จะใช้ถ้อยคำแตกต่างกันก็ตาม หากไม่มีระบบอัตโนมัติ การวิเคราะห์ลักษณะนี้จะต้องอาศัยมนุษย์ซึ่งอาจเกิดความคลาดเคลื่อนและไม่สามารถรองรับข้อมูลจำนวนมากได้

ดังนั้น โครงการนี้จึงมีวัตถุประสงค์เพื่อศึกษาและพัฒนาระบบวิเคราะห์ความคล้ายคลึงกันของข้อความ (Text Similarity) โดยอาศัยเทคนิคทางด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) และการเรียนรู้ของเครื่อง (Machine Learning) เพื่อให้ระบบสามารถเปรียบเทียบข้อความและคำนวณระดับความคล้ายคลึงได้อย่างมีประสิทธิภาพ โครงการนี้จะช่วยลดเวลาและเพิ่มความแม่นยำในการวิเคราะห์ข้อความ และสามารถนำไปประยุกต์ใช้ในงานด้านการศึกษาธุรกิจ และเทคโนโลยีสารสนเทศในอนาคต

2. วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาโปรแกรมตรวจสอบความคล้ายคลึงของเอกสารข้อความ (Text Similarity) โดยใช้หลักการ Cosine Similarity
- เพื่อศึกษาและประยุกต์ใช้เทคนิคการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) ขั้นพื้นฐาน เช่น
 - การตัดคำ
 - การลบคำหยุด (Stop Words)
 - การทำ Stemming อย่างง่าย
- เพื่อสร้างระบบที่สามารถรับข้อมูลได้ทั้งจากไฟล์ PDF และไฟล์ TXT รวมถึงการป้อนข้อความด้วยตนเอง

4. เพื่อออกแบบส่วนติดต่อผู้ใช้ (GUI) ที่ใช้งานง่าย ด้วยภาษา Python และไลบรารี CustomTkinter
 5. เพื่อแสดงผลลัพธ์ความคล้ายคลึงในรูปแบบเปอร์เซ็นต์ เพื่อช่วยประเมินการคัดลอกหรือความซ้ำของเอกสาร
3. ขอบเขตของโครงการ
 1. โครงการรองรับการเปรียบเทียบเอกสาร ครั้งละ 2 เอกสาร เท่านั้น
 2. ประเภทไฟล์ที่รองรับ ได้แก่
 - ไฟล์ข้อความ (.txt)
 - ไฟล์เอกสาร PDF (.pdf)
 3. ใช้วิธีการวิเคราะห์ความคล้ายคลึงแบบ Bag of Words และ Cosine Similarity
 4. มีการปรับปรุงข้อความก่อนประมวลผล โดย
 - แปลงตัวอักษรเป็นตัวพิมพ์เล็กทั้งหมด
 - ตัดเครื่องหมายวรรคตอน
 - ลบคำที่ไม่จำเป็น (Stop Words)
 - ทำ Stemming แบบง่าย
 5. ระบบไม่รองรับ
 - การเปรียบเทียบเอกสารหลายไฟล์พร้อมกัน
 - การตรวจสอบฐานข้อมูลออนไลน์หรืออินเทอร์เน็ต
 - การประมวลผลภาษาไทย (ออกแบบสำหรับภาษาอังกฤษเป็นหลัก)
 6. ผลลัพธ์จะแสดงเป็นเปอร์เซ็นต์ความคล้ายคลึง และใช้สีแสดงระดับความเสี่ยงของการคัดลอก

4. แผนการดำเนินงาน

ลำดับ	กิจกรรม	ระยะเวลา
1	กำหนดหัวข้อที่จะศึกษา	6 มกราคม 2569
2	วิเคราะห์และออกแบบระบบ	6 มกราคม 2569
3	พัฒนาโปรแกรม	7 มกราคม 2569
4	รวบรวมข้อมูลและทดสอบข้อมูล	8 มกราคม 2569
5	ประเมินผลและปรับปรุงระบบ	13 มกราคม 2569
6	จัดทำรายงานและนำเสนอผลงาน	23 มกราคม 2569

5. สรุปผลการดำเนินงาน

จากการพัฒนาโครงการ Plagiarism Checker System พบร่วมกับสถาบันฯ สามารถตรวจสอบความคล้ายคลึงของข้อความได้อย่างมีประสิทธิภาพในระดับพื้นฐาน โดยระบบสามารถอ่านข้อมูลจากไฟล์ PDF และ TXT ได้อย่างถูกต้อง และนำข้อความมาประมวลผลเพื่อคำนวณค่าความคล้ายคลึงด้วยวิธี Cosine Similarity

ผลลัพธ์ที่ได้สามารถแสดงเป็นเปอร์เซ็นต์ความคล้ายคลึงของเอกสารทั้งสองฉบับ พร้อมการแสดงสีเพื่อช่วยให้ผู้ใช้งานเข้าใจระดับความซ้ำของเอกสารได้ง่ายขึ้น ระบบมีส่วนติดต่อผู้ใช้ที่ใช้งานสะดวก ไม่ซับซ้อน และเหมาะสมสำหรับการใช้งานในเชิงการศึกษา เช่น การตรวจสอบงานรายงานหรืองานเขียนเบื้องต้น

อย่างไรก็ตาม ระบบยังมีข้อจำกัดในด้านความแม่นยำเมื่อเอกสารถูกเขียนใหม่ด้วยคำที่มีความหมายใกล้เคียงกัน (Paraphrase) และยังไม่รองรับภาษาไทยหรือการเชื่อมต่อกับแหล่งข้อมูลภายนอก ซึ่งสามารถนำไปพัฒนาต่อยอดในอนาคตได้

6. ผลที่คาดว่าจะได้รับ

- ได้ระบบตรวจสอบความคล้ายคลึงของเอกสารที่สามารถใช้งานได้จริง
- ระบบสามารถรับข้อมูลทั้งในรูปแบบข้อความและไฟล์ PDF ประมวลผล และแสดงผลความคล้ายคลึงของเอกสารในรูปแบบร้อยละ โดยอาศัยเทคนิค Cosine Similarity
- ผู้พัฒนาเข้าใจหลักการประมวลผลข้อความ (Text Processing) มากขึ้น
ผู้จัดทำโครงงานจะมีความเข้าใจในขั้นตอนการทำความสะอาดข้อความ เช่น การตัดคำ การลบ Stop Words และการปรับรูปคำ (Stemming) รวมถึงการแปลงข้อความเป็นเวกเตอร์
- ได้เรียนรู้และประยุกต์ใช้เทคนิค Cosine Similarity อย่างเป็นรูปธรรม
- สามารถนำแนวคิดทางคณิตศาสตร์และ Data Science มาประยุกต์ใช้กับปัญหาจริงในงานด้านการวิเคราะห์ข้อความ

อ้างอิง

GeeksforGeeks. (2025, July 15). *Cosine Similarity*. Retrieved January 8, 2026, from <https://www.geeksforgeeks.org/dbms/cosine-similarity/>

GeeksforGeeks. (2025, June 12). *How to Calculate Cosine Similarity in Python?* Retrieved January 8, 2026, from

<https://www.geeksforgeeks.org/python/how-to-calculate-cosine-similarity-in-python/>

MarcusCode. (2018, April 7). *Inheritance, การสืบทอดคลาส ในภาษา Python*. Retrieved January 8, 2026, from

<https://marcuscode.com/lang/python/inheritance>

Tom Schimansky. (n.d.). *Window — CustomTkinter documentation*. Retrieved January 8, 2026, from

<https://customtkinter.tomschimansky.com/documentation/windows/window/>